In this issue:
Learn how the zEnterprise System is transforming the world.
Also, performance benefits from z/OS V1R12

Greg Daynes
Systems Software Development
S/390 Installation Deployment Architect
Why is Greg Daynes smiling? Because he’s got the future of distributed processing in his hands, an IBM zEnterprise 196 (z196) processor. Along with the IBM zEnterprise BladeCenter Extension (zBX), it’s just one of the two new IBM hardware components that make up the zEnterprise System. Think of them as “engines of a smarter planet.”

As Greg, our cover subject, explains in the lead feature of this issue, the zEnterprise System combines “the built-in support” of these hardware elements with the IBM zEnterprise Unified Resource Manager (zManager) software to offer “a heterogeneous computing system . . . designed with improved scalability, performance, security, resiliency, availability, and virtualization.”

This issue is primarily devoted to all things zEnterprise. From Greg’s lead article that describes some of the migration actions to take, to descriptions of the hardware and software functions of the zEnterprise System, including zManager, the articles in this issue will help clarify why it’s truly the “IBM gold standard of enterprise computing.”

For example, “Understanding the IBM Smart Analytics Optimizer” describes how that product is designed for the zEnterprise, capable of hosting a large array of workloads including analytical business intelligence (BI) data.

In “Gain insight using HIS data” the authors explain how Hardware Instrumentation Services (HIS) lets you capture data about hardware events such as cycles, instructions, and cache misses for z10 and z196 processors.

And you will find plenty of information about the performance benefits of z/OS V1R12 as well. In fact, it’s sort of our second theme.

You can see how performance enhancements help ICSF improve throughput to cryptographic coprocessors and accelerators and how improvements to DFSMS provide fast replication processing to recover data sets.

There are updates to RACF generic profile loading and VSAM CA reclaim, and an article whose title says it all about changes to XML System Services support: “A whole lot of performance gained from a little bit of fragment parsing.”

As always, Hot Topics is devoted to you, with tips and “how tos” about the latest z/OS functions for products and components like z/OS Timed Event Data, reallocation, global resource serialization, RACF, DB2 security, IPSec, and disaster recovery for the sysplex, as well as our regular DFSMS feature “Ask Mr. Catalog” (along with a special guest author).

So come on in and explore the brave new world of zEnterprise and performance. You won’t be disappointed.

Drop us a line at newsletr@us.ibm.com.

The Editors
## Contents: Featured articles

<table>
<thead>
<tr>
<th>Page</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>zEnterprise gets you where you’re going</td>
</tr>
<tr>
<td></td>
<td>GREG DAYNES</td>
</tr>
<tr>
<td>12</td>
<td>z/OS goes hybrid with zEnterprise</td>
</tr>
<tr>
<td></td>
<td>GISELA CHENG AND DERRICK WASHINGTON</td>
</tr>
<tr>
<td>15</td>
<td>zEnterprise: An interactive guide to working with an ensemble</td>
</tr>
<tr>
<td></td>
<td>KIRSTEN BRUNNER MCDONALD, LAURA M. BOSTIAN, AND GAYLE C. STEINBUGLER</td>
</tr>
<tr>
<td>17</td>
<td>Understanding the IBM Smart Analytics Optimizer</td>
</tr>
<tr>
<td></td>
<td>KERSHAW S. MEHTA AND THERESA TAI</td>
</tr>
<tr>
<td>19</td>
<td>zEnterprise Unified Resource Manager</td>
</tr>
<tr>
<td></td>
<td>KIRSTEN BRUNNER MCDONALD, HIREN SHAH, AND PETER YOCOM</td>
</tr>
<tr>
<td>24</td>
<td>z/OS V1R12 capacity provisioning provides</td>
</tr>
<tr>
<td></td>
<td>MICHAEL GROETZNER, ADRIAN ALVAREZ DIEZ, AND RALF THELEN</td>
</tr>
<tr>
<td>29</td>
<td>How HCD finds paths (caution: some math involved)</td>
</tr>
<tr>
<td></td>
<td>MATTHEW SABINS</td>
</tr>
<tr>
<td>30</td>
<td>ICSF and Crypto Express</td>
</tr>
<tr>
<td></td>
<td>STEVEN HART, DON AULT, AND DAVID MANNERS</td>
</tr>
<tr>
<td>34</td>
<td>Classified: Highly effective system programmers</td>
</tr>
<tr>
<td></td>
<td>JODI EVERDON AND KRISTINE LOGAN</td>
</tr>
<tr>
<td>35</td>
<td>‘OLA from WebSphere Application Server on z/OS!</td>
</tr>
<tr>
<td></td>
<td>JIM MULVEY AND WENDY BRINKMAN</td>
</tr>
<tr>
<td>39</td>
<td>ICSF supports X9.24-compliant CBC wrapping for DES keys</td>
</tr>
<tr>
<td></td>
<td>ELEANOR CHAN, KATHLEEN FADDEN, AND KEN KERR</td>
</tr>
<tr>
<td>40</td>
<td>DFSMSHsm Fast Replication recovery</td>
</tr>
<tr>
<td></td>
<td>YUFEN DAVIS AND HENRY VALENZUELA</td>
</tr>
<tr>
<td>42</td>
<td>GRS RNL gotchas</td>
</tr>
<tr>
<td></td>
<td>JOSEPH GENTILE, TAMMY GARREN, AND STEVEN PARTLOW</td>
</tr>
<tr>
<td>44</td>
<td>What can VSAM CA reclaim do for you?</td>
</tr>
<tr>
<td></td>
<td>ARTHUR BARISKA, LARRY LAW, AND RENEE BROOKS</td>
</tr>
<tr>
<td>47</td>
<td>A whole lot of performance gained from a bit of fragment parsing</td>
</tr>
<tr>
<td></td>
<td>BILL CAREY AND MATTHEW COUSENS</td>
</tr>
<tr>
<td>48</td>
<td>Ask Mr. Catalog</td>
</tr>
<tr>
<td></td>
<td>STEPHEN M. BRANCH (MR. CATALOG) AND TAN Q. NGUYEN (SPECIAL GUEST)</td>
</tr>
<tr>
<td>49</td>
<td>zFavorites</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>Does your network fabric health check out?</td>
</tr>
<tr>
<td></td>
<td>MARISA FREIDHOF AND HARRY YUDENFRIEND</td>
</tr>
<tr>
<td>52</td>
<td>Longing for larger catalogs?</td>
</tr>
<tr>
<td></td>
<td>RICH LUNGERHAUSEN</td>
</tr>
<tr>
<td>54</td>
<td>OAM’s the place with the space</td>
</tr>
<tr>
<td></td>
<td>KEVIN GOLDSMITH</td>
</tr>
<tr>
<td>56</td>
<td>Your mileage may vary, but on z/OS V1R12 it is definitely going to be better!</td>
</tr>
<tr>
<td></td>
<td>STEVEN JONES</td>
</tr>
<tr>
<td>59</td>
<td>RACF “ghosts” busted</td>
</tr>
<tr>
<td></td>
<td>RUSSELL HARDGROVE, JUN OGATA, AND SCOTT WOOLLEY</td>
</tr>
<tr>
<td>60</td>
<td>Greatly improved RACF generic profile loading</td>
</tr>
<tr>
<td></td>
<td>RUSSELL HARDGROVE AND GEORGE MARKOUIZOS</td>
</tr>
<tr>
<td>61</td>
<td>Improve performance by understanding program flow with timed event data</td>
</tr>
<tr>
<td></td>
<td>SCOTT TUTTLE AND RITA BEISEL</td>
</tr>
<tr>
<td>63</td>
<td>SMF flood insurance</td>
</tr>
<tr>
<td></td>
<td>ANTHONY SOFIA</td>
</tr>
<tr>
<td>64</td>
<td>Take a second to understand our IPSec solution</td>
</tr>
<tr>
<td></td>
<td>DAVID WIEBOWSKI AND KEITH WELTER</td>
</tr>
<tr>
<td>67</td>
<td>SVC dumps just got smarter!</td>
</tr>
<tr>
<td></td>
<td>CHARLES MARI, ELPIDA TZORTZATOS, AND BRYAN CHILDS</td>
</tr>
<tr>
<td>69</td>
<td>Mirror, mirror on the wall, should these data sets by mirrored at all?</td>
</tr>
<tr>
<td></td>
<td>NICOLE FAGEN AND DONNY CHAN</td>
</tr>
<tr>
<td>72</td>
<td>Our contributors</td>
</tr>
<tr>
<td>75</td>
<td>Corrections and clarifications</td>
</tr>
</tbody>
</table>

**Back cover:**
Step up to the IBM System z Tech University
To boldly go...

zEnterprise gets you where you’re going

BY GREG DAVYNE

On July 22nd, 2010, IBM announced a “System of Systems.” The IBM zEnterprise System (zEnterprise 196 or z196) is a computing environment that combines unprecedented innovations with the IBM gold standard of enterprise computing—the System z mainframe. The result is a workload-optimized system: a heterogeneous computing system capable of hosting multiple workloads integrated together, yet each one optimized to meet individual business goals on an IT infrastructure managed as a single entity.

Has it all

The environment consists of the IBM zEnterprise 196 (z196) central processor complex, the IBM zEnterprise Unified Resource Manager (zManager), and built-in support for the IBM zEnterprise BladeCenter Extension (zBX) Model 002. The zBX is designed with improved scalability, performance, security, resiliency, availability, and virtualization.

The z196 Model M80 provides up to 1.6 times the total system capacity of the z10 EC Model E64, and all z196 models provide up to twice the available memory of the z10 EC processor. It is the world’s fastest and most scalable enterprise system.

The zBX infrastructure works with the z196 processor to enhance System z virtualization and management that, in turn, delivers an integrated hardware platform spanning System z mainframe and POWER7™ technologies.

zManager, delivered with the z196 server, is designed to provide end-to-end virtualization and management along with the ability to optimize technology deployment according to individual workload requirements. It provides tightly integrated, cross platform systems management capabilities and policy-based frameworks for optimizing multitier, heterogeneous workloads.

We never brag but . . .

The IBM announcement of the zEnterprise System received strong endorsement from industry analysts, the press, and IBM customers, including the following:

“The zEnterprise has the potential to change the way the organization thinks about mainframe computing; suddenly it's not just a mainframe anymore.”

- Alan Radding, Research Director, Independent Assessment at www. technologywriter.com/indassess/iazent.pdf

“Big Blue is still reinventing its big iron.”

- Computerworld

“The new IBM zEnterprise System represents a potentially revolutionary change to the platform and the next phase in the evolution of highly efficient, scalable processing opening up the possibility of hosting entire workloads on a single highly integrated system.”

- Martin Kennedy, a managing director at Citigroup

While the way an organization thinks about mainframe computing might have changed, the way in which system programmers can position z/OS™ to run on, and exploit, a zEnterprise system is very similar to what they did for earlier generations of servers, and continues to get easier.

This article looks at ways to position yourself for the zEnterprise system in terms of migration for hardware and software and then at the implementation of some of the new functions. So let’s get going!

Superb software support

All supported z/OS releases can run on the new server, even those releases that have extended defect support though a z/OS Lifecycle Extension offering. What that means is that all supported z/OS releases can participate in a sysplex with a coupling facility (CF) or coexisting z/OS image on a z196 server.

Note that not all System z196 functions are available in every z/OS release. For example:

- To exploit zManager, including network or performance management, you need to run z/OS V1R10 or later.
- To exploit the new z/OS FICON Discovery and AutoConfiguration (zDAC) support, which can save you time by discovering new and changed fabric-attached devices and can suggest configurations based on I/O policies that you set, you need to run z/OS V1R12.

For information about how the Hardware Configuration Dialog (HCD) makes use of zDAC support, see “How HCD finds paths (caution: some math involved)” on page 29 and “z/OS storage fabric discovery and auto-configuration” in z/OS Hot Topics Newsletter Issue 23, August 2010, GA22-7501-19.

Figure 1 provides an overview of the functions and supported releases for the zEnterprise. Also, note that for z/OS V1R7 and V1R8 you must have a z/OS Lifecycle Extension contract in place to acquire the required service. (A z/OS Lifecycle Extension contract is also required for z/OS V1R9 defect support.)
### z196 PSP bucket - 2817DEVICE 2817

#### Release

<table>
<thead>
<tr>
<th>Feature</th>
<th>z/OS V1R7</th>
<th>z/OS V1R7 w/zIIP</th>
<th>z/OS V1R8</th>
<th>z/OS V1R9</th>
<th>z/OS V1R10</th>
<th>z/OS V1R11</th>
<th>z/OS V1R12</th>
</tr>
</thead>
<tbody>
<tr>
<td>XL C/C++ ARCH(9) Tune(9)</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>OSA-Express3 Inbound Workload Queuing (IWQ)</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>New OSA display command</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>zdac support</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>Performance management</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>Networking management</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>IBM zEnterprise Blade Center Extension (zBX) support</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>Up to 32 HiperSockets™</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>Cryptographic exploitation</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>WP</td>
<td>WP</td>
<td>WP</td>
</tr>
<tr>
<td>Three subchannel sets</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>Extensions to zHPF multtrack operations</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>CF level 17</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>Power Save Mode</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P*</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>LE high register resolution</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>HiperDispatch serviceability</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P**</td>
<td>P**</td>
<td>B</td>
</tr>
<tr>
<td>HiperDispatch cache and affinity node changes</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>Greater than 64 CPs per LPAR</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>CPU Measurement Facility</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>RMF™ Postprocessor Crypto Hardware Activity Report: 4096-bit²</td>
<td>P**</td>
<td>P**</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>OSA-Express3 CHPID OSD maximum ports</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>HiperDispatch</td>
<td>N</td>
<td>WP</td>
<td>BP</td>
<td>BP</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>Crypto toleration³</td>
<td>WP</td>
<td>WP</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Greater than 64 CPs per server</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>IFAURR reporting</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>FICON Express8 (CHPID FC)</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>Up to 128 coupling link CHPIDs</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>BP</td>
<td>B</td>
</tr>
<tr>
<td>IB coupling links</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>New z/Architecture® instructions ⁴</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>OSA-Express3 (GbE LX and sx SX 1000BASE-T, 10 GbE LR and SR)</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>BP</td>
<td>B</td>
</tr>
<tr>
<td>Base GA1 support</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

#### Explicit z/OS support

- **N**: Not Supported
- **P**: PTFs are required
- **P****: PTFs are required for tolerance
- **WP**: Web deliverable
- **BP**: FMID is shipped in a web deliverable
- **B**: FMID is in the base product

**Key**

- **B** - FMID is in the base product.
- **P** - PTFs are required.
- **P****P** - PTFs are required for tolerance.
- **WP** - FMID is shipped in a web deliverable.
- **P*** - Support differs by release.
- **N** - The function is not supported.

---

Figure 1. zEnterprise functions and z/OS support

---

**NOTE:** For z/OS V1R7, z/OS V1R8, and z/OS V1R9 the Lifecycle Extension for each release is required for support.

1. Toleration PTFs are required to detect Power Saving Mode.
2. PTFs are required to recognize cryptographic control blocks and provide updated reports.
3. A cryptographic web deliverable is NOT required, but a toleration PTF is needed even if the web deliverable is installed.
4. Support differs by release: z/OS V1R12 includes XL C/C++ support for ARCH(9) and TUNE(9) options.
Coexistence PTFs for the zEnterprise

The following functions require that you install coexistence PTFs on non-z196 systems that plan to share resources with a system that runs on a z196 server:

- CF Level 17
- Power savings mode (if sharing SMFPRMxxx PARMLIB members)
- Cryptographic support.

Depending on your cryptography needs, consider the following:

- If you are enabling your existing level of Integrated Cryptographic Services Facility (ICSF) software to function as it did before you migrate to the zEnterprise, install one or more PTFs on a z196 server.
- If you are on z/OS V1R10, V1R11, or V1R12 and you want to exploit one of the many cryptographic enhancements introduced with the latest level of ICSF, download and install the Cryptographic Support for z/OS V1R10 - V1R12 deliverable (FMID HCR7780).

The web deliverable provides the following enhanced functions:

- An improved symmetric key store (CKDS).
- 64-bit enablement for the full set of ICSF callable services.
- Additional content in audit records.
- API support for new symmetric encryption modes introduced in the z196 server CP Assist for Cryptographic Functions (CPACF).
- New Common Cryptographic Architecture (CCA) key token wrapping mechanisms.
- Support for enhanced Personal Identification Number (PIN) protection techniques for online PIN verification (ANSI x9.8 and ISO 9564-1 PIN Management and Security standards).
- New CCA functions implementing the Elliptic Curve Digital Signature Algorithm (ECDSA), and new CCA functions for Hashed Message Authentication Codes (HMAC) using the SHA-1 and SHA-2 families of hash functions. Planned availability of this last function is the first quarter of 2011, with additional PTFs. See “Take a second to understand our IPSec solution” on page 64 for a description of some of this cryptographic support.

PSP and fix categories

As with earlier generations of servers, you can find information on hardware and software requirements in Preventive Service Planning (PSP) “buckets”:

- The UPGRADE 2817DEVICE SUBSET 2817/ZOS contains information on the z196 server.
- The UPGRADE 2458DEVICE SUBSET 2458/ZOS contains information on the zBX with pointers to where you can get additional information.

Similar to what IBM did when they introduced the z10 servers, the identification, verification, acquisition, and installation of the PTFs required for the zEnterprise System is integrated into standard SMP/E processing. You can use a simple SMP/E REPORT MISSINGFIX command on your z/OS systems to identify if fixes are missing for the z196 server:

```
SET BDY(GLOBAL).
REPORT MISSINGFIX ZONES(SOSR12T) FIXCAT(IBM.Device.Server.z196-2817).
```

You can acquire and RECEIVE the latest HOLDDATA onto one or more of your z/OS systems by using your normal service acquisition tool. Or try downloading the two year HOLDDATA file directly from the following website:

```
http://service.software.ibm.com/support/docview/390/390holddata.html
```

You must select Full from the Download NOW column (last 730 days) to receive the FIXCAT HOLDDATA, as the other files do not contain FIXCAT HOLDDATA.

When using the SMP/E REPORT MISSINGFIX command, you can use wildcarding or specify multiple categories to report on additional fix categories. For example, you can do the following for fixes:

- To report on fixes to exploit InfiniBand Coupling Links, you can specify:

```
IBM.Device.Server.z196-2817.
ParallelSysplexInfiniBandCoupling
Fix Category
```

- To report on the fixes to exploit Server Time Protocol, you can specify:

```
IBM.Device.Server.z196-2817.
ServerTimeProtocol Fix Category
```

- To report on fixes to exploit the z Manager, you can specify:

```
IBM.Device.Server.z196-2817.
UnifiedResourceManager
```

- To report on the coexistence service for cryptography support, you can specify:

```
IBM.Coexistence.ICSF.z/OS_V1R10-V1R12-HCR7780 Fix Category
```

- To report on fixes to exploit the IBM Smart Analytics Optimizer using a zBX server, you can specify:

```
IBM.Device.Server.zBX-2458.
ISAOPT fix Category
```

Note that a description of ALL fix categories is available from the following website:

```
ibm.com/systems/z/os/zos/smpes/fixcategory.html
```

For information about the REPORT MISSINGFIX command, see the description of SMP/E commands available at the following website:

```
```

Migration actions

From a migration perspective, you’ll be pleased to know that there are relatively few migration actions when you migrate from a z10 EC or z10 BC server. That is because the z196 server is the next evolutionary step from the z10 generation of servers. If you are migrating to a z196 server from a z10 EC or z10 BC server and have performed the migration actions associated with those servers, you’ll encounter fewer migration actions than if you migrate from a z9® or earlier generation of servers.
It is important to note that you can migrate directly to a z196 server without installing the intermediate (that is, the earlier generation) of servers, but you still need to ensure that any migration considerations are satisfied for the servers that you “skipped.”

The migration actions that you are required to perform on z/OS for a z196 server are very similar to what you needed to do to upgrade to a z10 server. Specifically, every new server generation introduces new machine instructions, and if you are running a multisystem sysplex, many introduce a new CF level to the mix.

New mnemonics

An important consideration involves the new mnemonics for assembler macro instructions that might collide with (that is, be identical to) the names of assembler macro instructions that you already use or provide. In the event of such collisions, the default opcode table (UNI) of the High Level Assembler (HLASM) treats the specification of these names as instructions when you install PTF for APAR PK97799, but be aware that this is likely to cause assembler error messages or the generation of incorrect object code.

If you write programs in the IBM Assembler Language, compare the names of assembler macro instructions that you use or provide to the list provided in z/Architecture Principles of Operation, SA22-7832-07. This allows you to identify any conflicts or collisions that might occur after you have installed the HLASM PTF for APAR PK97799.

If you find a conflict, take one of the following actions:

- Change the name of your macro instruction.
- Specify PARM='...OPTABLE(YOP)...' (or some other earlier opcode table).
- Create a separate ASMAOPT file containing assembler options, such as that described in the previous method. This method requires no changes to source code or JCL.
- Add *PROCESS OPTABLE(YOP) as the first statement of your source program.
- Apply the PROFILE option either in JCL, or in the ASMAOPT file. As a result, the specified or default member of the SYSLIB data set moves into the front of the source program.

- Use PUSH/POP ACONTROL to save and switch the OPTABLE dynamically if you must use both a new instruction and a macro with the same name in an assembly. For more information about HLASM’s opcode table, see HLASM Programmer’s Guide, SC26-4941-05.

When programs exploit the new machine instructions, or utilize the z/OS V1R2 XL C/C++ ARCH(9) or TUNE(9) options, those programs can only run on z196 servers; otherwise, you will receive an operation exception. This is a major consideration for program development, test, quality assurance, and production teams, as well as for anyone who is performing fallback testing or disaster recovery.

Parallel Sysplex environments

Another common migration consideration is the Parallel Sysplex® environment. Here is a point-by-point summary of those considerations for the zEnterprise System:

- IBM supports Parallel Sysplex environments that consist of the current or earlier two generations of hardware. For z196 servers, this means that z/OS images or coupling facility images on a z196 server are only supported in a Parallel Sysplex with other z196 servers, z10 EC and z10 BC servers, and z9 EC and z9 BC servers.

If you are running z/OS images on zSeries® z900, z800, z990, or z890 servers, you cannot add a z196 server to that sysplex. The earlier servers in the sysplex must be upgraded to System z9 or later to have support for the z196 server in the sysplex. Similarly, if you have any z/OS images or coupling facility images on an earlier server, and you intend to introduce a z196 server into that sysplex, you must migrate those images to a System z9 (or later) server before you can introduce the z196 server.

If you are moving your coupling facilities to a z196 server (or just to a server with a higher CFCC level), you must run the Coupling Facility Structure Sizer (CFSIZER) tool to find out if you have to increase coupling facility structure sizes. The z196 server introduces CFCC Level 17.

- The z196 does not support the Integrated Cluster Bus 4 (ICB-4) Coupling Links. Instead, use 12x InfiniBand coupling links, which are designed to replace Integrated Cluster Bus 4 (ICB-4), and to complement 1x InfiniBand and ISC-3 on a z196 server.

- You cannot connect the z196 server to a Sysplex Timer® (9007-002). The IBM Server Time Protocol (STP) feature is the follow-on to the Sysplex Timer. STP is designed to allow multiple servers and coupling facilities to maintain time synchronization with each other without requiring a Sysplex Timer. STP is a hardware feature of the z196, z10 EC, z10 BC, z9 EC, z9 BC, z990, and z890 servers.

- Even though the z196 server does not support attachment to a Sysplex Timer, it can participate in a Mixed Coordinated Timing Network (Mixed CTN) that has either a z10 or a z9 server synchronized to the Sysplex Timer. This implementation allows enterprises to migrate concurrently from an existing External Time Reference (ETR) to a Mixed CTN and from a Mixed CTN to an STP-only CTN.

HiperDispatch considerations

When you upgrade to a z196 server that uses HiperDispatch, you must enable HiperDispatch on any z196 LPAR with greater than 64 processors defined. This limit affects images when they IPL or when you dynamically add processors. When you IPL with greater than 64 processors defined and HPERDISPATCH=NO in IEAOPTxx, the system generates the following message, but continues to run with HPERDISPATCH=YES:

```
IRA865I HPERDISPATCH=YES FORCED DUE TO GREATER THAN 64 LPS DEFINED
```

On an LPAR that specifies HPERDISPATCH=NO with fewer than 64 processors, you can dynamically add more processors and continue to run in the HPERDISPATCH=NO mode.
However, you might see the following new system message:

```
ISSN126 HIPERDISPATCH MUST BE ENABLED TO CONFIGURE CPU IDS GREATER THAN 3F ONLINE.
```

Any attempt to configure additional CPUs that result in greater-than-64 CPUs online is in effect, and HIPERDISPATCH=NO is rejected with the following message:

```
IEE241I CPU(x) NOT RECONFIGURED ONLINE - REQUIRES HIPERDISPATCH ENABLED.
```

Checking the limit
To notify you that you are getting close to the 64 CPU limit and are running with HIPERDISPATCH=NO, the IBM Health Checker for z/OS check, SUP_HiperDispatchCPUConfig, is available with z/OS V1R12 and with the PTF for APAR OA30476 for on z/OS V1R11 and z/OS V1R10.

The check always succeeds for an LPAR when HIPERDISPATCH=YES is specified (that is, all CPU configurations are supported). When an LPAR runs with HIPERDISPATCH=NO, the check raises an exception if the number of CPUs is close to forcing the LPAR to IPL with HIPERDISPATCH=YES. The CPUSLEFTB4NEEDHD parameter indicates the minimum number of CPUs that can be installed and activated on an LPAR running in HIPERDISPATCH=NO. When CPUSLEFTB4NEEDHD=0, the check always succeeds. The default is 8 with values 0-63 accepted.

Using new zEnterprise functions
The last consideration for positioning z/OS for a zEnterprise System is using some of the new functions. The good news is that many of the functions are enabled or disabled, based on the presence or absence of the required hardware, required software, and in many cases configuration settings to exploit the new capabilities. A few of the functions have coexistence considerations (namely installing coexistence PTFs on systems that plan to share resources). Using the following functions requires some customization.

**CF level 17**
Requires changes in your structure sizes (see Parallel Sysplex environments) and greater than 1023 coupling facility structures function requires a new version of the CFPRM CDS.

**Power Savings mode**
A new SMFPRMxxx parmlib option, MAXEVENTINTRECS, allows governing the number of event interval records to be collected when the processor capacity changes. The default is zero. To collect extra records between regular intervals when the processor capacity changes, you must adjust the default.

**Third subchannel set**
Third subchannel set requires definitions similar to those for the second subchannel set, which you can make with HCD. The IODF statement of LOADxx allows you to select the subchannel set from which to bring primary PPRC devices online. The SCHSET parameter now allows values 0, 1, or 2. It can also be set using an operator prompt during IPL.

**z/OS FICON Discovery and Auto Configuration (zDAC)**
To use zDAC, you must first establish a policy for the discovery operation by using the HCD or Hardware Configuration Management (HCM).

You can limit the scope of the discovery, limit the proposal information, indicate the desired number of paths to discovered logical control units, and indicate the method used for device and control unit numbering. After discovery, you can select which controllers to define and accept or override the proposed values for control units and devices.

**zManager**
zManager manages System z ensembles. Ensembles are collections of one or more zEnterprise System nodes in which each node is comprised of a z196 server and its optionally attached IBM zBX server.

An ensemble can consist of a single z196 server without a zBX server attached, using an OSA loopback cable, or two to eight CPCs where at least one of the z196 servers has a zBX server attached. The resources of a zEnterprise System ensemble are managed and virtualized as a single pool of resources integrating system and workload management across the multisystem, multitier, and multi-architecture environment. Management actions for the ensemble are conducted from a designated primary Hardware Management Console (HMC).

To improve availability, you must pair this HMC with an alternate HMC, also designated to manage the ensemble, that can take over if the primary fails. Ensemble membership requires that the z196 server have two OSA-Express3 1000BASE-T features, defined as CHPID type OSM, for connectivity to the intranode management network (INMN). Communication across the INMN is exclusively for enabling zManager of the HMC to perform performance management, network virtualization management, energy management, and other management tasks. (For information about OSA-Express 3, see “This is bound to impress you: QDIO inbound workload queuing” on page 21.)

Starting with zEnterprise System, the guest platform management provider is the interface between the z/OS Workload Manager (WLM) and zManager. The interface supports platform management functions and provides policy information to WLM about the platform-wide performance goals of workloads in which the z/OS system is participating. To use the guest platform management provider, you must update your WLM definitions.

For more information, be sure to check out the article “z/OS goes hybrid with zEnterprise” on page 12.

**DB2 and IBM Smart Analytics Optimizer**
To enable DB2® for z/OS to route queries to an attached IBM Smart Analytics Optimizer, you must apply a particular PTF. For details and for additional information regarding hardware and software requirements, see the IBM Smart Analytics Optimizer for DB2 for z/OS support website at the following site:


Also, see the article “Understanding the IBM Smart Analytics Optimizer” on page 17.

**The road ahead: zEnterprise System**
All supported z/OS releases can run on a z196 server. However, full exploitation of the hardware capabilities of z196 servers depends on the z/OS release level. The
software release determines the software requirements (PTFs and in some cases FMIDs provided in web deliverables), and the functional capabilities. See Figure 1 for a summary of z/OS support for a z196 server by release.

Service support for z/OS V1R7, z/OS V1R8, and z/OS V1R9 has ended. Anyone running z/OS V1R7 or z/OS V1R8 must have a z/OS Lifecycle Extension (or extended support) contract in place to acquire the required service. (A z/OS Lifecycle Extension contract is also required for z/OS V1R9 defect support.)

Keep these points in mind about migrating to zEnterprise:

- Verify installation of all the service from the appropriate PSP buckets (with any installation or upgrade). To that end, you can use the SMP/E V3.5 function to simplify the verification and installation tasks associated with PSP bucket service.
- Perform the relatively few migration actions when you move from a z10 EC or a z10 BC server to the new zEnterprise servers. Those migration actions are very similar to what was necessary to upgrade to a z10 server.

Install the required software and activate the hardware so you can use many of the new functions for zEnterprise although expect to make some required, but minor, configuration updates to use of some of these functions.

Come along and explore the new world of zEnterprise. It will get you where you’re going...

References
- CFSIZER: [ibm.com/systems/support/z/cfsizer](http://ibm.com/systems/support/z/cfsizer)
- InfiniBand coupling links: IBM System z Connectivity Handbook, SG24-5444
- STP website and publications: [ibm.com/systems/z/advantages/pso/stp.html](http://ibm.com/systems/z/advantages/pso/stp.html)

Learn to speak fluent z: zEnterprise terms and concepts

You’d expect a new product like IBM’s zEnterprise System to come with its own lingo. Take a look at this zEnterprise glossary to understand some of these key terms and concepts (adapted from zEnterprise System Introduction to Ensembles, GC27-2609).

---

**A**
application environment. The environment that includes the software and the server or network infrastructure that supports it.

**B**
blade. A hardware unit that provides application-specific services and components. The consistent size and shape (or form factor) of each blade allows it to fit in a BladeCenter chassis.

**C**
central processor complex (CPC). A physical collection of hardware that consists of main storage, one or more central processors, timers, and channels. In the zEnterprise environment, the CPC consists of a System z zEnterprise mainframe and any attached IBM zEnterprise BladeCenter Extension (zBX). See also node and zCPC.

**D**
CPC. See central processor complex

**E**
ensemble. A collection of one or more zEnterprise nodes (including any attached zBX) that are managed as a single logical virtualized system by zManager, through the use of a Hardware Management Console.

**F**
firmware. Licensed Internal Code (LIC) that is shipped with hardware. Firmware is considered an integral part of the system and is loaded and run at power on. Firmware is not open for customer configuration and is expected to run without any customer setup.

**G**
GPMP. See guest platform management provider.

**H**
guest platform management provider (GPMP). An optional suite of applications that is installed in specific z/OS, Linux, and AIX® operating system images to support platform management functions. For example, the guest platform management provider collects and aggregates performance data for virtual servers and workloads.
Hardware Management Console (HMC). A user interface through which data center personnel configure, control, monitor, and manage System z hardware and software resources. The HMC communicates with each central processor complex (CPC) through the Support Element. On an IBM zEnterprise 196 (z196), using zManager on the HMCs or Support Elements, personnel can also create and manage an ensemble. See also primary HMC.

HMC. See Hardware Management Console.

Hypervisor. A program that allows multiple instances of operating systems or virtual servers to run simultaneously on the same hardware device. A hypervisor can run directly on the hardware, can run within an operating system, or can be imbedded in platform firmware. Examples of hypervisors include PR/SM, z/VM®, and PowerVM Enterprise Edition.

IBM Smart Analytics Optimizer for DB2 for z/OS. An optimizer that processes certain types of data warehouse queries for DB2 for z/OS.

IBM System z Application Assist Processor (zAAP). A specialized processor that provides a Java execution environment, which enables Java-based web applications to be integrated with core z/OS business applications and backend database systems.

IBM System z Integrated Information Processor (zIIP). A specialized processor that provides computing capacity for selected data and transaction processing workloads, and for selected network encryption workloads.

IBM zEnterprise 196 (z196). The newest generation of the System z family of servers built on a new processor chip, with enhanced memory function and capacity, security, and on demand enhancements to support existing mainframe workloads and large scale consolidation.

IBM zEnterprise BladeCenter® Extension (zBX). A heterogeneous hardware infrastructure that consists of a BladeCenter chassis attached to a IBM zEnterprise 196 (z196). A BladeCenter chassis can contain POWER blades or optimizers.

IBM zEnterprise System (zEnterprise). A heterogeneous hardware infrastructure that can consist of a IBM zEnterprise 196 (z196) and an attached IBM zEnterprise BladeCenter Extension (zBX) Model 002, managed as a single logical virtualized system by zManager.

IBM zEnterprise Unified Resource Manager (zManager). Licensed Internal Code (LIC), also known as firmware, that is part of the Hardware Management Console. zManager provides energy monitoring and management, goal-oriented policy management, increased security, virtual networking, and data management for the physical and logical resources of a given ensemble. IEDN. See intraensemble data network (IEDN).

Optimizer. A special-purpose hardware component or appliance that can perform a limited set of specific functions, with optimized performance when compared to a general-purpose processor. Because of its limited set of functions, an optimizer is an integrated part of a processing environment, rather than a standalone unit. One example of an optimizer is the IBM Smart Analytics Optimizer for DB2 for z/OS.

OSM. An OSA-Express channel path identifier (CHPID) type that provides connectivity to the intranode management network (INMN).

OSX. An OSA-Express channel path identifier (CHPID) type that provides connectivity to the intraensemble data network (IEDN).

Performance index. A number that indicates whether the performance goal for a service class was achieved, exceeded, or missed.

Performance policy. A description of the performance objectives and importance of a workload.

Platform management. The subset of systems management focused on hardware and virtualization management.

Primary HMC. The System z Hardware Management Console (HMC) through which data personnel create and manage an ensemble. This HMC owns configuration and policy information that zManager uses to monitor, manage, and adjust resources for all members of this ensemble.

Manage suite (Manage). The first suite of functionality associated with the IBM zEnterprise Unified Resource Manager (zManager). The Manage suite includes core operational controls, installation, and configuration management, and energy monitoring.
power system control network (PSCN). The power subsystem of the System z servers that is controlled by a fully redundant dual-Ethernet communications network. This network provides communication to all field-replaceable units (FRUs) and hierarchic control through a mirrored system of control cards and IP addresses. The PSCN provides a means for subsystems to communicate and control the dynamic parameters of system operation. The PSCN also supports error reporting, failure data collection and recovery detection, and correction of both the internal hardware and firmware of the System z servers.

**PSCN.** See *power system control network.*

static power save mode. A IBM zEnterprise 196 (z196) function used for periods of low utilization or potentially when a CBU system is sitting idle waiting to take over in the event of a failure. The server uses frequency and voltage reduction to reduce energy consumption of the system. Static power save mode is initiated by the customer using the HMC or Support Element or Active Energy Manager.

**Unified Resource Manager.** See IBM zEnterprise

**virtual server.** A logical construct that appears to comprise processor, memory, and I/O resources conforming to a particular architecture. A virtual server can support an operating system, associated middleware, and applications. A hypervisor creates and manages virtual servers.

**virtual server collection.** A set of virtual servers that are involved in supporting a workload. This set is not necessarily static. The constituents of the collection at any given point are determined by the virtual servers involved in supporting the workload at that time. See also *virtual appliance* and *virtual server image.*

**virtual server image.** A package containing metadata that describes the system requirements, virtual disks, and any goals and constraints for the virtual machine (for example, isolation and availability). The Open Virtual Machine Format (OVF) is a Distributed Management Task Force (DMTF) standard that describes a packaging format for virtual server images. See also *virtual appliance* and *virtual server collection.*

**virtual server image capture.** The ability to store metadata and disk images of an existing virtual server. The metadata describes the virtual server's storage, network needs, goals, and constraints. The captured information is stored as a virtual server image that can be referenced and used to create and deploy other similar images.

**virtual server image clone.** The ability to create an identical copy (clone) of a virtual server image that can be used to create a new similar virtual server.

workload. A collection of virtual servers that perform a customer-defined collective purpose. A workload generally can be viewed as a multi-tiered application. Each workload is associated with a set of policies that define performance and energy consumption goals.

**z196.** See IBM zEnterprise 196 (z196).

**zAAP.** See IBM System z Application Assist Processor.

**zBX.** See IBM zEnterprise BladeCenter Extension (zBX).

**zCPC.** The physical collection of main storage, central processors, timers, and channels within a System z zEnterprise mainframe. Although this collection of hardware resources is part of the larger zEnterprise central processor complex, you can apply energy management policies to the zCPC that are different from those that you apply to any attached IBM zEnterprise BladeCenter Extension (zBX) or blades See also *central processor complex.*

**zIIP.** See IBM System z Integrated Information Processor.

**zEnterprise.** See IBM zEnterprise System (zEnterprise).

**zManager.** See IBM zEnterprise Unified Resource Manager (zManager).
The IBM zEnterprise System brings a new dimension into the world of computing. In a zEnterprise System, an ensemble is a collection of one or more zEnterprise nodes managed as a single, logical, virtualized system. Each node contains a zEnterprise 196 (z196), and an optionally attached zEnterprise Blade Center Extension (zBX). This integrated hardware platform makes it possible to deploy and manage workloads across mainframe and distributed technologies and environments. Now, users can deploy an application across multiple platforms and have it monitored and managed by the zEnterprise Unified Resource Manager (zManager).

**zEnterprise: A robust hybrid**

Figure 1 shows a logical view of a single node ensemble with zManager. z/OS and Linux images running on the z196 and AIX images on the zBX can join the ensemble as virtual servers. (Think of a virtual server as any logical construct that comprises processor, memory, and I/O resources and conforms to a particular architecture. Examples in an ensemble include an operating system, associated middleware, or applications.)

**Managing it all**

You can install the guest platform management provider in z/OS V1R10, z/OS V1R11, or z/OS V1R12 to permit z/OS to participate in the zEnterprise. It is the interface between z/OS Workload Manager (WLM) and zManager. The guest platform management provider supports platform management functions by providing policy information to WLM, which is defined in the HMC, about the platform performance goals of the various workloads with which the z/OS system is involved. The guest platform management provider also collects and reports performance data for any workload with which any z/OS system is involved.

First, let’s find out how your business can benefit from using the zEnterprise System. Then, we’ll look at how to set up and configure the guest platform management provider, how to start it, and the type of data the guest platform management provider collects and sends to zManager.

**Business benefits using zEnterprise**

Multi-tier workloads and their deployment on diverse infrastructures are common today. The zEnterprise system provides a highly optimized environment for deploying such workloads with monitoring and management capability.

The z/OS system is the backbone of the software solution that can enhance SOA architectures and provide significant improvements for Java programs. In this scenario, an application running under WebSphere is deployed in an AIX system (virtual server) that runs in the zBX together with the WebSphere Application Server. The application retrieves data stored on a database server, in this case, DB2 that resides on a z/OS system. As WebSphere, DB2, and the web server that
runs with the WebSphere Application Server plugin are Application Response Measurement (ARM)-instrumented, transaction level data is made available, and transaction performance can be measured. ARM is an Open Group standard and shouldn’t be confused with the Application Restart Manager of z/OS. An ARM-instrumented application is one in which application response measurement calls are added to the source code to enable the performance of the application to be monitored by management systems.

Note that transaction level data is only available when you use ARM-instrumented middleware (like WebSphere Application Server, DB2, and the web servers associated with the WebSphere Application Server plugin). In this scenario, the guest platform management provider that runs on AIX and z/OS collects the transaction data and passes it to the HMC.

Figure 2 is an example of a transaction “hop” report that the guest platform management provider generates for the service classes within the default workload. It displays the number of hops (that is, the intermediate connections in a string of connections that link two network devices) the state of the transaction (successful, failed, stopped, and in flight), and the time spent on each tier for the transaction in a given interval.

The hop report shows that this is a three-tiered transaction:
- The data for hop 0 shows transaction on IBM WebServing Plugin (Hardware Instrumentation Data or HIS)
- The data for hop 1 shows transaction on WebSphere Application Server
- The data for hop 2 shows transaction on DB2, each with average response time.

Together with other reports, the system administrator can better understand if the performance goal of the deployed workloads is being achieved, and if not, which virtual servers or partitions are contributing to the performance problem. For more information about HIS and instrumentation measurement, see “Gain insight using HIS data” on page 24.

Setting up the guest platform management provider

Consider running guest platform management provider in a SECLABEL environment, that is, an environment that activates the security class or SECLABEL to cause the system to use security labels for authorization checks when the resource has a security label.

You must authorize the guest platform management provider as a started task and ensure that it is able to communicate with WLM and use the z/OS UNIX System Services. IBM provides a sample job HVEENV in SYS1.SAMPLIB to help you configure the security and execution environment that is needed to run the guest platform management provider. The job HVEENV performs two tasks:

Task 1: Configures the security environment, by issuing RACF commands to do the following:
- Create RACF group HVEMCA and user ID HVEMCA1.
- Authorize HVEMCA to run as trusted started task.
- Authorize HVCMCA to access OSM, the internal communication network.

Task 2: Configures the directories to store the configuration and diagnostics data and the execution environment by specifying the location of the required Java product to be one of the following:
- IBM 31-bit SDK for z/OS, Java 2 Technology Edition, V5
IBM recommends that your end users keep the RACF groups HVEMCA and HVEMCA1 as group and user name. If your installation uses a security product other than RACF, it needs to update the equivalent RACF commands. Your installation can also update the variables in task 2 to specify the location to store the configuration and diagnostics information for your specific environment.

Both methods invoke the started task HVEMCA that is located in SYS1.PROCLIB. HVEMCA creates a z/OS UNIX shell that starts the guest platform management provider. After the guest platform management provider starts, it connects to WLM and uses the z/OS UNIX file system, the directories defined in HVEENV to store the configuration, and diagnostic information. zManager must be available for the guest platform management provider to discover the support element (SE). When the SE is discovered, a policy is made available to the guest platform management provider that it passes to WLM for activation.

Starting the guest platform management provider

In z/OS V1R12, there are two ways to start the guest platform management provider:

1. Through the Workload Manager (WLM) panel
2. By issuing the following WLM MODIFY command from the operator console:

In z/OS V1R10 and V1R11, the only way to start the guest platform management provider is by issuing the MODIFY command.

How the guest platform management provider retrieves data

After the guest platform management provider starts, it periodically collects and sends the following data to the SE every 10 seconds:

- Server statistics
- Service class data
- Application data
- Application resource data.

An installation can use the reporting function in the HMC to view the data and monitor performance.

zEnterprise has it

z/OS is entering the new computing world of zEnterprise by leading the way to leverage its well-known strengths by providing the following:

- Constant Workload Manager evolution
- IBM System z Application Assist Processor (zAAP) support
- IBM System z Integrated Information Processor (zIIP) support
- System integrity with enhanced security,
- Continuous reliability, availability and serviceability (RAS).

That is just to name a few. The mainframe operating system is going hybrid in a big way, and that’s a great thing for the future of computing.

Find out more about the zEnterprise System at the following website:

ibm.com/systems/z/hardware/-enterprise/zmanager.html
The IBM zEnterprise Unified Resource Manager (zManager) introduces integrated monitoring and management of heterogeneous resources as a single, logical, virtualized system—AKA, the ensemble. To step you through setting up and managing an ensemble, the latest release of the System z Hardware Management Console (HMC) offers an interactive Ensemble Management Guide.

Intended for those not familiar with the HMC, the Ensemble Management Guide is an interactive task guide within zManager and contains step-by-step links to all the required tasks to create and manage an ensemble from one central place.

**Ensemble Management Guide**

Creating and managing an ensemble requires a combination of existing tasks and new tasks including:

- Performance management
- Integrated private data network management
- Virtual server lifecycle management
- Hypervisor management
- Integrated operational controls
- Management of System z firmware.

The ensemble management tasks are permitted to the Ensemble Administrator and Service roles defined in the HMC. These roles can be combined with other roles to create custom user IDs that span a variety of management tasks in the Ensemble Management Guide.

You can open the Ensemble Management Guide task from the Task Index or from the TaskPad area after expanding the Ensemble Management node in the zHMC navigation tree. You can link to more information about the tasks from the Guide.

---

**Figure 1. Ensemble Management Guide: summary**
The Ensemble Management Guide brings together the set of diverse tasks you need to customize your new ensemble and the environment around it. Depending on your needs, you might perform some of these tasks a number of times and others not at all. While there is no specific order in which you must perform these tasks, the Ensemble Management Guide provides you with a recommended order in which to accomplish them.

**Keeping track of your tasks**
To help keep track of which steps have been started or completed, and by whom, a free-form entry field is available in the guide for logging notes which can be displayed or hidden by clicking on the Notes icon in the upper right-hand corner.

The Ensemble Management Guide starts with some steps to consider before you create an ensemble, such as managing the users who can access the new ensemble tasks.

Next, the Ensemble Management Guide will help you to define your ensemble, directing you to an alternate HMC to be designated for backup management support, and then returning to the primary HMC to add at least one member to the ensemble—a CPC or “virtual server” to be managed as part of a single, logical, virtualized system.

The customization of an ensemble will vary based on your needs. Some configurations might need only one virtual server and have no need at all for a virtual network. Another configuration might define multiple virtual servers for their ensemble, and define specific workloads and performance policies to run those workloads on different virtual servers across an elaborate virtual network.

Once the ensemble is humming along, there are monitoring and reporting tools to help assess the health of the ensemble. The Ensemble Management Guide contains a link to the new Monitors Dashboard to view performance metrics and a link to launch and customize workload reports.

All together now
So, if you’re unsure of what objects you can include in your ensemble, or what tasks are necessary to help you achieve your desired configuration, the Ensemble Management Guide is a great tool for planning and running the necessary tasks to pull it all together.

For a glossary of enterprise terms, see “Learn to speak fluent z” on page 9.
Get Smart!
Understanding the IBM Smart Analytics Optimizer

BY KERSHAW S. MEHTA AND THERESA TAI

In today’s competitive business environment, making faster, more informed decisions to achieve an edge against the competition is the challenge every company faces. Many companies are turning to analytical business intelligence (BI) to gain insight from their exploding data volumes. Others have adopted data warehousing solutions into their information management infrastructure, deploying dedicated servers and solutions that result in duplication of operational data across a distributed environment. This in itself causes many challenges such as additional security guidelines and new operational rules to govern this data duplication.

In July 2010, IBM announced the IBM zEnterprise System (zEnterprise) that combines unprecedented innovations in mainframe technology along with a heterogeneous “compute platform” capable of hosting many workloads including one for BI by way of the IBM Smart Analytics Optimizer for DB2 for z/OS.

The zEnterprise is an integrated hardware platform made up of IBM zEnterprise 196 (z196) processors—the industry’s fastest and most scalable enterprise system and the new zEnterprise BladeCenter Extension (zBX)—an infrastructure hosting high-performance specialty processors (blades) for specific workloads, such as the IBM Smart Analytics Optimizer.

IBM Smart Analytics Optimizer technology
The IBM Smart Analytics Optimizer is a revolution in analytic technology. First, it provides a new model for data warehouses by allowing you to combine operational and BI workloads, thereby streamlining the business infrastructure. This means that you can deploy a BI workload with your existing Online Transaction Processing (OLTP) workload currently running on DB2 for z/OS.

Second, IBM Smart Analytics Optimizer exploits many disruptive hardware technologies including large number of processor cores making use of massive amounts of main memory (RAM) and fast interconnects, which are all innovative technologies in zEnterprise. The specialty blades in the zBX supply aggressive use of commodity multi-core and vector processing to achieve parallelism. Exploitation of large main memories, along with compression, allows warehouses to fit in memory when partitioned among multiple blades. In fact, because IBM Smart Analytics Optimizer performs no decompression of data and no I/O, it can produce a disk-less query response, thereby improving overall query response time. Finally, the zBX utilizes a private high-speed internal network that connects it to the z196 to establish a TCP/IP and Distributed Relational Database Architecture (DRDA®) connection between DB2 for z/OS and the IBM Smart Analytics Optimizer.

Origins
The IBM Smart Analytics Optimizer is the product of an IBM Almaden research project called Blink technology. The aim of this project was to provide consistently low, interactive query response times, regardless of any specific query structures when accessing the underlying data. The project developed technologies to accelerate processing of long-running online analytical processing (OLAP) queries by orders of magnitude. This technology implements hybrid row and column stores in main memory exploiting predicate evaluation on highly compressed data using multi-core and vector-optimized algorithms. Blink technology is a perfect application for zEnterprise.

The nature of queries
Data warehousing and BI queries are typically complex, long running, and ad hoc in nature. OLAP queries typically perform large table scans. To help relieve the cost of running these highly resource intensive workloads and provide a more predictable response time, database administrators (DBA) have used expression on indexes and Materialized Query Tables (MQT) as ways to help tune BI workloads. However, many times this is not enough—BI queries are complex and sometimes hard to tune because of their unpredictable design. The IBM Smart Analytics Optimizer provides consistent query response times for a given amount of data thereby alleviating the need to perform tuning activities.

The sweet spot
IBM Smart Analytics Optimizer was designed for OLAP-style SQL queries, typically characterized by star or snowflake schema that touch a large subset of the database (unlike OLTP queries) that look for trends or exceptions and typically involve some type of aggregation function like sum, average, or count.

Data mart definition
The first step is identifying the data to accelerate—the set of tables that are referenced by the long-running queries. The set of tables, logically related to each other, is the data mart. For example, all tables of a single star schema would belong to the same data mart.

To define the data mart, the DBA can use the IBM Smart Analytics Optimizer studio, a graphical user interface (GUI), to define the tables that belong to a data mart, together with information about their relationships. DB2 creates definitions for these data marts in its own catalog.
After defining a data mart, the DBA creates a snapshot of the tables from the operational data store (DB2 tables), and then loads it in the IBM Smart Analytics Optimizer. The “unload” process can be controlled either by the GUI, or by invoking the IBM Smart Analytics Optimizer’s administrative stored procedures.

When loading the data onto the IBM Smart Analytics Optimizer, the data transforms into a highly compressed, scan-optimized, pre-joined format and partitioned among the main memory of the available blades. In addition, the partitioned mart is also stored on a separate storage unit for failover purposes. When this process is complete, the mart is available to query.

Integration with DB2 for z/OS

After deployment, the IBM Smart Analytics Optimizer acts as another resource manager at the disposition of DB2 for z/OS, just like the Data Manager (DM), Buffer Manager (BM), the Internal Resource Locking Manager (IRLM). Use of the IBM Smart Analytics Optimizer is transparent to the query and to the application, which generates the SQL query. In fact, existing applications require no change to benefit from the capabilities of IBM Smart Analytics Optimizer.

Users and applications continue to connect to DB2 for z/OS, as they do today. Changes made internally to the DB2 optimizer (component of DB2 for z/OS) let it become aware of the IBM Smart Analytics Optimizer’s existence in a given environment. Think of the DB2 optimizer as the intelligent traffic cop that decides if a given query can run either on the IBM Smart Analytics Optimizer or by using the already well-known access paths in DB2 for z/OS.

The DB2 optimizer uses a cost-based algorithm and decision-making process for query routing; all queries run in their most efficient environment, regardless of their type (OLAP versus OLTP). Furthermore, if the zBX is unavailable or DB2 for z/OS has lost connectivity to the IBM Smart Analytics Optimizer, queries do not fail. Rather, queries run on the z196 by DB2 for z/OS accessing the operational data store.

Integration with zEnterprise

A zEnterprise System is an integrated hardware platform that makes deployment and management of workloads across mainframe and distributed technologies seamless. The IBM zEnterprise Unified Resource Manager (zManager) provides integrated management for the zEnterprise System. It can dramatically simplify operations across the multiple application environments. zManager provides many of IBM’s leading mainframe-like governance and qualities of service. For example, the System z maintenance strategy now extends to the IBM Smart Analytics Optimizer by providing operation control for the specialty blades, automated call home for hardware and firmware problems, concurrent hardware repair, concurrent firmware fixes, and driver upgrades.

For definitions of new or unfamiliar zEnterprise terminology, see “Learn to speak fluent z” on page 9.

An example – putting it all together

For instance, a retail organization wants to find out how many surfboards they sold in the state of Oklahoma in the month of January to decide whether to restock or to lower the price. This multi-dimensional query requires scanning a large subset of the sales database in order to make an informed decision. It’s highly unlikely that the DBA built an index or a materialized query table for such a dynamic query.

Given its unpredictable nature, this type of complex, ad hoc query might never be allowed to run on an OLTP system. Now this query matches the sweet spot of the IBM Smart Analytics Optimizer! Applications can generate and submit these types of queries to DB2 for z/OS, where it is redirected to execute on the specialty blades running IBM Smart Analytics Optimizer.

New beginning

IBM Smart Analytics Optimizer is an integration of IBM hardware, software, storage, and advanced technologies focused on business analytics. It augments IBM DB2 for z/OS into an enterprise data warehouse by accelerating selected queries by orders of magnitude. IBM Smart Analytics Optimizer along with zEnterprise is a game-changer for BI and data warehousing workloads.

Learn more:

- To find out more about IBM Smart Analytics Optimizer, see the following website:
  [ibm.com/systems/z/hardware/zenterprise/zmanager.html](http://ibm.com/systems/z/hardware/zenterprise/zmanager.html)
- To find out more about Blink, see the following website:
- Also, see the article “Practice safe sets (data)” on page 36.

---

Image of IBM Smart Analytics Optimizer

(Figure 1. Overview of IBM Smart Analytics Optimizer)
One of the strengths of System z is the ability to run multiple workloads at the same time across multiple z/OS images and to manage them according to business performance goals. The IBM zEnterprise Unified Resource Manager (zManager) extends this performance management to both traditional System z and BladeCenter hardware environments. zManager provides policy-based monitoring and dynamic processor adjustments to ensure that multitier applications that span System z hardware and BladeCenter hardware are managed with sufficient processor resources.

**zManager, workloads, and performance policies**

For z/OS workload management (WLM), a workload is a customer-defined collection of work to be tracked, managed, and reported as a unit. Customers use business goals or functions to define z/OS WLM workloads. Similarly, a zEnterprise workload is a customer-defined collection to be tracked, managed, and reported as a unit that reflects a business goal or function. In this case, however, the collection consists of logical constructs called virtual servers that perform a customer-defined collective purpose such as hosting a business application.

On zManager, a virtual server is a PR/SM partition, z/VM virtual machine, or a PowerVM partition. A virtual server that supports a single business application belongs to a single workload. A virtual server that supports multiple business applications can belong to multiple workloads that represent each of these business applications. You associate a performance policy with the workload, and platform resources are presented, monitored, reported, and managed for that workload.

A performance policy describes the business importance of the workload and performance objectives for the virtual servers belonging to the workload. The performance policy is a collection of service classes where a service class identifies performance goals for a subset of virtual servers within the workload. The service class also defines the importance for its virtual servers relative to other virtual servers within the workloads.

**zManager and the HMC**

The latest release of the zHardware Management Console (HMC) provides the interface to define a workload, set workload performance policy, monitor workloads, and set alerts to notify you about important performance events.

Using event monitoring, you can set thresholds and define alerts on processor utilization of virtual servers and the performance goal achievement of service classes. When you have defined these controls, you are notified by email or page when the event threshold has been reached. The alerts appear on several workload monitoring reports.

**zManager reports**

zManager provides various monitoring reports to help you understand the performance of your workloads. You can drill down from the overall workload performance health view to contributions of individual virtual servers. Several reports provide performance metrics:

- The Monitors Dashboard Report shows basic system resource utilization statistics.
- The Workloads Report shows an overview of each workload’s performance.
- The Service Classes Report shows the performance of the individual service classes in a workload.
- The Virtual Servers Report shows detailed performance statistics for the set of virtual servers which make up a workload.
- The Hypervisor Report shows detailed performance statistics of the all virtual servers running on a hypervisor.
See Figure 1 for examples.

Figure 1. zManager reports

zManager and processor resources

Today the Intelligent Resource Director (IRD) manages processor resources across z/OS logical partitions based on the z/OS workload management service policy. A partition’s processor allocation is changed by adjusting the processor processing weight of the partition.

zManager provides IRD-like management for other hypervisors supported in a zEnterprise system. zManager manages processors across virtual servers within the supported hypervisor instances; for example, PowerVM on zBX Power™ blades and z/VM.

Managing the performance of virtual servers

When a specific PowerVM or z/VM virtual server within a workload is not achieving its goals, zManager first determines whether the performance of that virtual server can be improved with additional processor resources. If performance can be improved by additional processor allocations, zManager then projects the impact of reducing processor allocations for other virtual servers under the same hypervisor. If the projected impact shows the workload will be closer to achieving the goals of its performance policy, zManager adjusts the processor allocations of the virtual servers.

You can request and view information about resource adjustments through the Resource Adjustments report in the HMC. This Report lists all successful and unsuccessful resource adjustments for a given workload, virtual server, or service class. The report identifies the donors and receivers of additional resources.

While zManager monitors the performance of z/OS LPARs and provides performance data reports for those virtual servers, it does not manage processor resources for them.

Managing workloads for the future

Every IT installation wants to make good use of its resources and maintain the highest possible throughput, to help achieve the best possible system responsiveness and ultimately meet service level agreements (SLAs). Presenting a single virtualized heterogeneous system, zManager workload management can optimize the physical and virtual system resources that support an application.
You can now improve inbound workload performance by using multiple read queues with the z/OS V1R12 QDIO inbound workload queuing (IWQ) function on OSA-Express. When IWQ is enabled, OSA-Express places inbound network traffic for certain workload types (for example, sysplex distributor and streaming bulk data) onto separate read queues that are dedicated to those workload types. This function allows the z/OS TCP/IP stack to process inbound data for these read queues on multiple concurrent threads, which, in turn, improves system throughput and scalability.

Requirements: The IWQ function requires an OSA-Express3 in QDIO mode on an IBM System z10 or z196 server.

Background
Here’s a bit of background on OSA-Express and queue processing.

Outbound traffic
Before talking about OSA-Express QDIO inbound processing, let’s look at the outbound side. OSA-Express supports four write queues that allow for outbound traffic separation based on priority. You can enable this support in one of two ways:

- Using the Policy Agent to map type-of-service values to outbound priorities
- Using the priority queuing function of the z/OS Workload Manager (WLM) to map WLM service classes to outbound priorities.

Single-read bottlenecks
However, until now, OSA-Express has supported only a single-read queue to handle all inbound traffic. Here are some of the problems this can present.

When a single processing thread handles all inbound data for the interface, z/OS Communications Server can become a bottleneck. For example, inbound interactive traffic might be delayed behind inbound streaming traffic over the same interface.

When there is a high inbound traffic rate over an interface, z/OS Communications Server might schedule multiple threads on a multiprocessor system to service the single-read queue. However, for a streaming TCP connection, these processing threads can race against one another such that the stack receives the inbound packets out-of-order, causing the stack to send duplicate acknowledgments. Sometimes, this results in the other side of the connection re-transmitting packets. When these racing threads also send new outbound data in response to inbound acknowledgements, the stack can transmit packets out-of-order with a negative impact on overall throughput.

The z/OS stack also sets I/O interrupt timing values with OSA-Express to control how frequently the adapter presents I/O interrupts to the host. The stack dynamically adjusts these values based on changing traffic patterns. However, if packets arrive over the single-read queue for a mix of workload types with different latency requirements, these timing values will not be optimal for all the traffic.

The inbound solution
When IWQ is enabled, z/OS Communications Server and OSA-Express establish multiple-read queues (a primary queue and two ancillary queues) for inbound traffic, as you can see in Figure 1. One ancillary queue is for streaming bulk data (such as FTP connections). The other ancillary queue is for Sysplex Distributor (SD) traffic. SD provides workload balancing across Transmission Control Protocol/Internet Protocol (TCP/IP) stacks in a sysplex.
The z/OS TCP/IP stack registers the routing variables (RVs) with OSA-Express to identify the specific inbound traffic the adapter needs to place on each queue. OSA-Express inspects the headers of each inbound IPv4 and IPv6 packet looking for a matching RV to determine the appropriate read queue.

The RV for bulk data is the “5-tuple” of the TCP connection, that is, (1) source and (2) destination IP address, (3) source and (4) destination port, and (5) TCP protocol. The z/OS stack identifies streaming TCP connections and registers those with OSA-Express to cause packets for these connections to arrive on the bulk queue. The RV for SD is the distributed dynamic virtual IP address (DVIPA) and TCP protocol. The z/OS stack registers each distributed DVIPA with OSA-Express to cause TCP traffic destined for such a DVIPA to arrive on the SD queue. OSA-Express places packets that do not match any registered RV onto the primary queue.

**IWQ benefits**

With IWQ, z/OS Communications Server can service the read queues concurrently on separate processing threads, which improves overall throughput. The sorting of inbound traffic onto separate queues by OSA-Express also allows the z/OS stack to more efficiently process the packets. For example, one thread can receive SD packets and forward them to target stacks using the QDIO accelerator. In parallel, another thread can receive streaming data for FTP connections. Also, by separating streaming traffic onto its own queue and servicing a TCP connection on a single processing thread, the out-of-order packet delivery problems that we discussed are minimized.

When IWQ is enabled, the z/OS stack sets different I/O interrupt timing values with OSA-Express for each read queue to provide optimal processing for the distinct workload types with different latency requirements. For example, the primary queue might consist mostly of interactive traffic that benefits from different interrupt settings rather than the streaming bulk data queue. So improve your inbound traffic, overall throughput, and reduced latency by using IWQ when you are running a mixed inbound workload over the same OSA-Express QDIO interface.

**Turn it on!**

To turn on QDIO IWQ specify the following parameters on the INTERFACE statement:

```
INBPERF DYNAMIC WORKLOADQ
```

IWQ also requires the use of a virtual MAC address (VMAC). You can request that a VMAC be automatically generated by simply specifying the following parameter on the INTERFACE statement:

```
VMAC
```

**Converting to the INTERFACE statement**

IWQ is configurable only on the INTERFACE statement. If you are using DEVICE, LINK, and HOME statements to configure OSA-Express QDIO interfaces in your TCP/IP profile, you must first convert to the INTERFACE statement.

The INTERFACE statement offers several benefits over DEVICE, LINK, and HOME statements. With the INTERFACE statement, you can configure several new performance features including IWQ. The INTERFACE statement also makes it easier to add, modify, or delete an interface on the fly using the system command VARY TCPIP,, OBEYFILE.

**More info**

The *z/OS V1R12 Communications Server IP Configuration Guide* has step-by-step instructions to help you convert your existing DEVICE, LINK, and HOME statements to the INTERFACE statement. There’s also an Interactive Problem Control System (IPCS) option that displays your current configuration using the INTERFACE statement for all OSA-Express QDIO interfaces. You can use this IPCS option as an aid for your migration to the INTERFACE statement. For more information about the IPCS option, see the *z/OS V1R12 Communications Server IP Diagnosis Guide*.

**Looking under the hood**

When you have IWQ enabled you should notice immediate benefits of having additional dedicated read queues for sysplex distributor and streaming workloads. If you are curious about what’s going on inside IWQ, you can pop...
When IWQ is enabled, the z/OS stack sets different I/O interrupt timing values with OSA-Express for each read queue to provide optimal processing.

the hood and look at a number of neat displays and statistics for the new read queues.

Display TRLE command
The Display TRLE command shows you all the read queues that are active and the type of workloads they each handle:

```
D NET,TRL,TRLE=QDIO101
```

Display OSAINFO command
The Display OSAINFO command, which is also new in z/OS V1R12 Communications Server, shows you which read queues you have active, and for each queue, the routing variables that are currently registered:

```
D TCPIP,,OSAINFO,INTFN=V403ETHG0
```

Here’s sample output from the command:

```
Ancillary Input Queue Routing Variables:
 Queue Type: BULKDATA   Queue ID: 2  Protocol: TCP
 Src: 9.67.125.40..15503
 Dst: 9.67.125.5..15044
 Total number of IPv4 connections: 1
 Queue Type: SYSDIST   Queue ID: 3  Protocol: TCP
 Addr: 201.2.10.31
 Total number of IPv4 addresses: 1
```

TNSTAT command
The VTAM® tuning statistics command (TNSTAT) is a great way to gather detailed statistics about inbound and outbound data on each queue. You’ll be able to display numerous counters, such as the number of packets and bytes received on each read queue:

```
MODIFY NET,TNSTAT,TRLE=QDIO101,CNSL,TIME=3
```

And here’s sample command output:

```
IST1233I DEV = 0E2A  DIR = RD/1  (PRIMARY)
IST1719I PCIREALO = 0  PCIREAL = 1630
IST1720I PCIVRTO = 0  PCIVRT = 0
...
IST1754I NOREADSO = 0  NOREADS = 0
IST1722I SBALCNTO = 0  SBALCNT = 214
IST1722I PACKCNTO = 0  PACKCNT = 500
IST2185I FRINVCTO = 0  FRINVCT = 0
IST1236I BYTECNTNO = 0  BYTECNT = 173982
IST1810I PKTIQDO = 0  PKTIQD = 0
IST1811I BYTIQDO = 0  BYTIQD = 0
IST924I -------------------------------------------------------------
IST1233I DEV = 0E2A  DIR = RD/2  (BULKDATA)
IST1754I NOREADSO = 0  NOREADS = 0
IST1722I SBALCNTO = 0  SBALCNT = 451
IST1722I PACKCNTO = 0  PACKCNT = 1365
IST2185I FRINVCTO = 0  FRINVCT = 0
IST1236I BYTECNTNO = 0  BYTECNT = 2033850
IST1810I PKTIQDO = 0  PKTIQD = 0
IST1811I BYTIQDO = 0  BYTIQD = 0
IST924I -------------------------------------------------------------
IST1233I DEV = 0E2A  DIR = RD/3  (SYSDIST)
IST1754I NOREADSO = 0  NOREADS = 0
IST1722I SBALCNTO = 0  SBALCNT = 965
IST1722I PACKCNTO = 0  PACKCNT = 965
IST2185I FRINVCTO = 0  FRINVCT = 0
IST1236I BYTECNTNO = 0  BYTECNT = 327084
IST1810I PKTIQDO = 0  PKTIQD = 0
IST1811I BYTIQDO = 0  BYTIQD = 0
IST924I -------------------------------------------------------------
```

The proof is in the performance IBM put IWQ to the test with various sysplex distributor, streaming, and mixed workloads and published the results in a whitepaper. You can see the performance data at the following website:

```
ibm.com/support/docview.wss?uid=swg27020476
```

External References
For more information about IWQ, see the following publications:

- z/OS V1R12 Communications Server IP Diagnosis Guide, GC31-8782-11
- z/OS V1R12 Communications Server IP Configuration Guide, SC31-8775-17.
An important feature introduced with the System z10 platform is the CPU measurement facility (CPU MF), the hardware support for collecting information about how the software and hardware interact on your system. CPU MF is supported by a new z/OS component, Hardware Instrumentation Services (HIS). This article describes the data provided by CPU MF and HIS and shows how customers can directly benefit from this information.

CPU Measurement Facility (CPU MF): sampling hardware and software data

CPU MF allows for the collection of counters that can provide insights into the interaction between software and hardware on your z10 processor and the new z196 processors. The facility also allows for the collection of instruction samples with very little overhead. Instruction samples contain all the information needed to determine the address space and address where the sampled instruction is running. In addition, the sample contains information about the home address space and the unit of work under which the code is running. Finally, the sample contains a signal which represents the frequency with which the instruction was executed.

HIS: capturing hardware events

HIS, which is supported in z/OS, captures the CPU MF data and makes it available to z/OS system programmers. Using HIS, you can capture counter data indicating the frequencies of various hardware events such as cycles, instructions, and cache misses. This information provides a unique view of the cache/memory characteristics of z10 and z196 processors. It can be used to supplement traditional performance metrics as a secondary source to help understand why performance has changed. It can also provide insights into workload and hardware interaction and, using a new relative nest intensity (RNI) metric, can provide a “hint” to help select the best workload from the latest Large System Performance Reference (LSPR) for capacity sizing. (See Now that you have the measurements . . . for more about the RNI.)

Using HIS, you can also capture instruction samples and information about the locations of various modules on your system. With these two pieces of information and proper tools support, it’s possible to determine what executable code on your system is consuming the most CPU time and how frequently that code is called.

HIS COUNTER data and its benefits

One of CPU MF modes is called “COUNTERS” because it actually counts events. These counts can be utilized to create unique performance metrics that performance and capacity analysts can utilize. The key insight that these counts provide is the interpretation of z10 and z196 cache/memory characteristics, which is not provided in any traditional systems management facilities (SMF) records (for example like those for Resource Measurement facility or RMF).

Exactly what does it count? Basic counters provide counts of cycles, instructions, and level-1 misses and cycles when instructions are not the processor level-1 cache. In addition, information in the extended counters indicates the source of a level-1 cache miss when it occurs. On a z10 machine it could be sourced from level-1.5 private, level-2 local, level-2 remote, or memory. For z196 processors, there is an additional cache level, and the miss can be sourced from level 2, level 3, level-4 local, level-4 remote, or memory. Sourcing essentially means that the instructions and data need to be retrieved from the appropriate level to ensure integrity for the cache coherency model.

From the above “counts,” the following performance metrics can be derived:

<table>
<thead>
<tr>
<th>Sample performance metrics</th>
<th>z10 abbreviation</th>
<th>z196 abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles per instruction</td>
<td>CPI</td>
<td>CPI</td>
</tr>
<tr>
<td>L1 miss percentage</td>
<td>L1MP</td>
<td>L1MP</td>
</tr>
<tr>
<td>Private cache L15 / L2 hit percentage</td>
<td>L15P</td>
<td>L2P</td>
</tr>
<tr>
<td>On chip cache L3 hit percentage</td>
<td>N/A</td>
<td>L3P</td>
</tr>
<tr>
<td>On book L2 / L4 local hit percentage</td>
<td>L2LP</td>
<td>L4LP</td>
</tr>
<tr>
<td>Off book L2 / L4 remote hit percentage</td>
<td>L2RP</td>
<td>L4RP</td>
</tr>
<tr>
<td>Memory hit percentage</td>
<td>MEMP</td>
<td>MEMP</td>
</tr>
</tbody>
</table>
Cycles per instruction is a rate of delivery metric like CPU per transaction or CPU per I/O, except that it is for the entire LPAR. There is no “good” or “bad” value, and it is often dependent on the workload mix. The L1MP shows how often a L1 miss occurs, and the above metrics “decompose” where the misses were sourced from within the cache/memory hierarchy.

This COUNTERS information can be written on an interval basis to an SMF type-113, subtype 2 record. By default the SMF records are written every 15 minutes, but with the PTF for APAR OA30486 these SMF records can be synchronized with the SMF global recording interval or written at a user defined interval from 1 to 60 minutes.

COUNTERS is enabled at an individual z/OS system/LPAR. The COUNTERS are captured for each logical processor. To enable COUNTERS, follow these four steps:

1. Ensure that the machine is a z10 General Availability 2 (GA2) or z196 processor and that it has the appropriate z/OS HIS PTF for the APARS installed.
2. Update the LPAR security tabs to allow COUNTERS (you can enable these tabs dynamically for COUNTERS).
3. Configure HIS by setting up the HIS procedure, defining an OMVS directory so that summary information can be written, and updating SMFPRMxx to include SMF type-113 records.
4. Start the HIS address space, then modify HIS to start collecting COUNTERS. The recommended MODIFY command is:

   `F HIS,B,TT='Text',PATH='/his/',CTRONLY,CTR=(B,E),SI=SYNC`

Figure 1 shows how CPU MF and HIS provide a z/OS logical view of resource usage and caching:

You essentially have no measurable overhead when you enable COUNTERS mode, and the recommendation is to keep it enabled on the key LPARS. Many customers are successfully running COUNTERS mode today in production for z10s and z196 processors.

Now that you have the measurements . . .

What can you do with these performance metrics? These metrics can be used to provide a secondary source of information as to why performance has changed. Typically performance is measured on transaction or job throughput, or CPU per transaction level from SMF and RMF. But now where there is a change in configuration or performance, the COUNTERS can be used to validate why there was a change.

For example if a customer has enabled HiperDispatch=YES, the COUNTERS performance metrics could be used to see how much the Level 2 Remote % (for the z10 processor) has been reduced as HiperDispatch attempts to align the logical processors being dispatched onto the same book. This would be additional validation in addition to traditional measurements.

Another use of the CPU MF COUNTERS was introduced with the announcement of the z196 processor. This announcement introduced three new LSPR workload categories, (Low, Average, and High) based on the RNI. RNI is a new metric to measure the distribution and latency in the processor “nest”. The “Nest” is the shared caches, memory and buses that interconnect them to the processor and private caches. It has been found that performance and capacity can be strongly influenced by how a workload interacts with the hardware, and COUNTERS and SMF subtype 113 records provide the ability to
gain insight into this interaction. RNI is calculated using the percentages of the shared caches and memory. For a z10 processor, RNI is calculated as follows. Note that this formula might change in the future:

\[
\text{z10 RNI} = \frac{(L2LP \times 1 + L2RP \times 2.4 + MEMP \times 7.5)}{100}
\]

Now customers can use RNI as a “hint” to help select their workload when referencing the LSPR or utilizing the free IBM zPCR tool for capacity sizing.

In summary, HIS COUNTER data provides new insights and capability for capacity and performance analysts to help understand why performance has changed and to help characterize a workload when utilizing the LSPR tables or the zPCR tool for capacity sizing.

HIS SAMPLE data and how to benefit from it
HIS also provides the support needed to collect hardware instrumentation SAMPLE data. Instruction addresses are sampled by the hardware on every logical processor associated with your LPAR of interest. This approach minimizes the overhead allowing for the collection of more samples with less overhead than can be achieved using software-based sampling techniques.

For example, one CPU LPAR on a full speed z10 EC machine can collect eight million samples in ten minutes while consuming only about two percent of the CPU.

On the same machine, a four-CPU LPAR can collect eight million samples in ten minutes with only one quarter (.5 percent) of this overhead.

HIS also allows you to collect a map of the locations of the various executable load modules on your system. Using this map in combination with the sample data, it’s possible to match the samples to the various modules. The number of samples in a module relative to the total samples on the system can provide a good estimate of how much of the total CPU time on the system was spent in this module. Another signal in the sample also allows you to estimate the frequency with which various instructions in the module were called. Using this signal, it’s also possible to estimate the cycles per instruction on a module or code path basis.

Find out more:
Find out more information about these functions and tools.

CPU MF is supported by a new z/OS component, Hardware Instrumentation Services (HIS).

HIS report tool
A simple example of a report generation tool for HIS sample data can be found on AlphaWorks:

www.alphaworks.ibm.com/tech/rghisd

This tool provides a basic report showing the various modules on your system, or in a selected home address space, the % CPU time, the % instructions, and the average cycles per instruction in these modules.

STG Lab Services
STG System z Lab Services can provide the skills and sophisticated tools to analyze and understand HIS SAMPLE data, helping you to identify potential tuning opportunities to make your system run more efficiently and to improve the performance on your system. Figure 2 shows a screen shot example of the tooling used with these HIS SAMPLE data services:

Figure 2. HIS sample data
Capacity Provisioning enhancements

BY MICHAEL GROETZNER, ADRIAN ALVAREZ DIZ, AND RALF THELEN

Capacity Provisioning now can manage temporary physical capacity for one or more IBM System z10™ or IBM zEnterprise 196 (z196) systems. Temporary capacity is general-purpose capacity or the number of temporary IBM System z Application Assist Processors (zAAPs) and IBM System z Integrated Information Processors (zIIPs) as described by an On/Off Capacity on Demand record. General purpose capacity is available as the number of additional general purpose processors (CP) or higher capacity levels (that is, speed) on sub-capacity models.

The following topics describe the new function for Capacity Provisioning in z/OS V1R12.

Power save mode
The z196 server supports a static power savings mode. In this mode, the capacity of the machine is reduced, thereby saving energy. Capacity Provisioning monitors the status of the power savings, the information about state changes, and the current status is made available through Capacity Provisioning console messages and reports. When you enable the power savings mode, the Provisioning Manager does not activate additional temporary capacity but continues to deactivate resources when necessary. You might observe the following new messages when you use this support:

- CPO3025W Static power save mode has been enabled for CPC name. No temporary capacity will be activated while power save mode is enabled.
- CPO3024I Static power save mode has been disabled for CPC name.

Most z196 servers allow you to enable static power saving just once a day. In this case, you might also see the following message:

- CPO3026I Enabling of static power save mode now allowed for CPC name when you can enable it again.

In addition, from the z/OS console you can use Capacity Provisioning commands to enable and disable power saving mode. The following two new Capacity Provisioning commands are available:

- ENABLE POWERSAVE CPC=cpc_name
- DISABLE POWERSAVE CPC=cpc_name.

To enable your system to accept the power saving commands, use the CPC power capping setting on the Hardware Management Console (HMC). Set the Energy Management tab on the CPC Details panel, to “Custom.”

CICS/IMS support
Do you want to manage your physical processors according to workload peaks including CICS® and IMS™ applications? In z/OS V1R12, Capacity Provisioning supports CICS and IMS transaction service classes in workload conditions within a Capacity Provisioning policy. For that reason, Capacity Provisioning uses newly introduced Resource Measurement Facility (RMF) data. This enhancement is also available on z/OS V1R10 and V1R11 through a PTF for APAR OA29641.
Moving average PI support
Do you monitor heavily fluctuating workloads on your systems or observe delayed provisioning or deprovisioning performance index (PI) actions? If so, you might smooth the PI graph by setting the SystemObservation_MovingAveragePiWeight parameter. The value range is 1 to 100. The default of 100 disables averaging. Values below 100 enable the calculation and management on behalf of moving average PI for all observed workloads.

The results of setting this value are an average calculation of indices (PI) that make it easier to define provisioning criteria. A smaller weight value produces an increased smoothing by giving more weight to older PI data, while a larger value discounts older data in benefit of the latest PI data and produces less smoothened average PI.

Using the SystemObservation_MovingAveragePiCapping parameter, the maximum PI limit is considered for the calculation of the moving the average PI. This prevents extremely high PI values from distorting computed moving average PI in the long-term. The value is 1.3 to 55. The default of 0 disables capping. Both parameter settings apply to all observed service classes.

Control center enhancements
The Capacity Provisioning Control Center (CPCC) lets you: define Capacity Provisioning policies and domain configurations on your workstation, and then upload them to z/OS.

But you might wonder...
• What if you want to check directly on your workstation which domain policy or domain configuration is currently activated in a Capacity Provisioning Manager (CPM) on z/OS?
• How can you get detailed status information of the CPM from within CPCC?

New Status dialogs in CPCC
Beginning in z/OS V1R12, connecting the latest CPCC to a z/OS V1R12 Provisioning Manager lets you to display the following status panels:
• The “Connected Domain” status as shown in Figure 1
• The “Active Configuration” status as shown in Figure 2.

The new panels help you determine whether the status of a Provisioning Manager matches your expectations. The panels are also an easy alternative to issuing status queries through the MVS™ console command interface. You can now define a domain configuration or policy and compare their values with the currently active settings on a connected domain within the same user interface GUI.

For more information about Capacity Provisioning enhancements, see z/OS V1R12.0 MVS Capacity Provisioning User’s Guide, SC33-8299-04.
Creating a robust data center is a feat. Each step requires extensive planning to create the best data solution your IT budget can allow. Recently a new function in hardware configuration definition (HCD) was introduced to discover and configure your control units. Path proposal automatically optimizes the paths of the control units for availability to reduce and remove single points of failure. This optimization requires an interesting cross of mathematical theory and engineering know-how. To help automate this process HCD introduced discovery and auto-configuration of new control units in z/OS v1R12, but how does it work?

Just so you know, after this point there will be math, so you might need to dust off those computer science textbooks.

For the purposes of single point-of-failure avoidance, a CHPID and port number are defined as a list of many hardware components, for example an I/O cage, power supply, CHPID card, switch, fabric name server, host-adapter power supply, and others. Paths can be represented as in following example where $C_i$ is a representation of a component of the I/O configuration:

$$\text{Path } P_1 := \{C_1, C_2, C_3, \ldots, C_n\}$$

A control unit (CU) is attached to the system with a set of 1 - 8 paths (CHPIDs and CU ports). Of course, one path is a single point of failure, so at least two paths should be defined. We can represent a set $S$ mathematically as follows:

$$S_i := \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\}$$

Formally $S_i$ is a $k$-combination of all the paths that attach the host to the control unit, where $k$ is a number from 2 - 8 and $i$ is a combination iterator.

Rule 1: The total single point of failure rule. Set $S_1$ is better than set $S_2$ if there are the fewest number of single failures that can affect connectivity. In other words, if there is one component (like a switch) that is common to all paths in set $S_2$, and $S_1$ does not have a common component to all paths, $S_1$ is automatically better than $S_2$.

Rule 2: The spread rule. After rule 1 is satisfied, set $S_1$ is a better set than $S_2$ if there are fewer common components within each set. In other words, if $S_1$ uses components C1 four times, and uses C2 four times, and $S_2$ uses components C1 six times, and uses C2 two times, $S_1$ is considered a better set.

Rule 3: The contention reduction rule. After rule 2 is satisfied, $S_1$ is better than $S_2$ if it uses fewer components than what was already used by previous installations. In other words, if component C1 is already used by 10 other control unit installations and component C2 has not been used by any other control unit installation, sets that use C2 are better than sets that use C1, when all other things are equal.

To summarize: math makes it so

With these representations and rules defined, you can optimize your data center by using a generic search algorithm or heuristic algorithm. Auto-configuration HCD uses a modified Djikstra’s Algorithm that computes the optimization heuristically. HCD computes the optimization automatically so that you do not have to prove a single theorem. Ain’t math grand?
ICSF and Crypto Express
Fulfilling your need for speed

BY STEVEN HART, DON AULT, AND DAVID MANNERS

“Faster than a speeding bullet”

0 K. The analogy to a speeding bullet might be a trifle overused, but the performance enhancements that cryptographic support for z/OS V1R10 - V1R12 (HCR7780) offers to Integrated Cryptographic Services Facility (ICSF) is something to brag about. With HCR7780, ICSF provides significant performance enhancements that boost cryptographic key data set (CKDS) update processing and increases your throughput to cryptographic coprocessors and accelerators.

Better throughput for ICSF processing

Updates to the CKDS VSAM data set can cause ICSF to retain the ENQ, allocation, and open processing of the data set across multiple requests. In a single system environment, the data set might remain allocated and open indefinitely. In a sysplex, requests are processed in bursts on each system before the system closes the data set and releases the ENQ.

With enhancements that HCR7780 offers for z/OS V1R10 - V1R12, cryptographic coprocessor and accelerator throughput increases are achieved through the Crypto Express interrupt architecture, which replaces an earlier polling scheme implementation for response retrieval. Additionally, ICSF in-storage CKDS cache is now stored in a tree backed by a 64-bit cell pool. The use of a tree eliminates the need for periodic refreshes of the CKDS that earlier releases of the product needed to maintain performance. These significant performance enhancements, along with the ability to scale the CKDS up to and beyond the millions-of-keys realm, make HCR7780 one of the most impressive releases of ICSF yet.

CKDS update bursts

Before z/OS V1R10 ICSF, each client separately obtained the CKDS ENQ while performing an update request. As each client obtained the CKDS ENQ, ICSF would perform the VSAM data set allocate, open, write, close, and deallocate steps for each client request. This management process of the CKDS ENQ proved to be inefficient and wasted cycles for repeatedly opening up and closing down the CKDS data set.

In z/OS V1R10 - V1R12, HCR7780 provides efficiencies by keeping the CKDS ENQ and the data set allocated and open across multiple CKDS update requests. This implementation allows a burst of CKDS update requests to be processed at a single time. For each burst the VSAM allocate and open are only performed on the first request, and the VSAM close and deallocate are only performed on the last request. The burst can span multiple clients because the ENQ is passed along to clients by ICSF to avoid unnecessarily releasing and re-obtaining of the CKDS ENQ.

Processing the CKDS update

A single instance of ICSF continually processes CKDS update requests until it detects another requestor for the ENQ on the same system or in the sysplex. ICSF performs this check by calling GRS with an ISGQUERY on the CKDS ENQ. If another instance of ICSF, a utility, or both are waiting on the CKDS ENQ, ICSF closes the CKDS and releases the ENQ to avoid locking out the other requestors. If no other requestor is interested in the CKDS ENQ, ICSF keeps the ENQ and continues to process CKDS update requests.

When ICSF completes all outstanding CKDS update requests, it checks again with global resource serialization to determine if it should give up the CKDS ENQ. If no one else wants the ENQ and there are no more CKDS update requests to process, ICSF sets a short timer to allow any incoming requests a chance to continue within the burst.

By allowing more requests to make their way in, ICSF can recognize when a client is making multiple CKDS update requests at a time. For clients starting jobs containing thousands of sequential CKDS updates, holding the ENQ over the completion of a request enables the CKDS burst support to process this environment with a significant reduction in cycles. If the timer pops and no more requests have come through, ICSF closes the CKDS data set and releases the CKDS ENQ.

What does this new processing mean to you?

Consider the scenario where a single client starts a large number of CKDS updates at a given time, and there is no contention for the ENQ from other clients. In this situation, you achieve a maximum performance boost by the CKDS update burst support. In this case, ICSF holds the CKDS ENQ, allocates and opens, and processes every request before it closes the CKDS and releases the ENQ.

Dramatic performance enhancements have been observed in this scenario, showing up to an 85% reduction in path length and a 465% performance improvement (dependant on system workload and configuration). Even with ENQ contention, you can see a significant performance boost that CKDS update burst support can provide.
Faster is better

In releases of ICSF before HCR7780, a polling scheme was used to retrieve Crypto Express responses. ICSF used a general response retrieval sub-task to continually poll (every 500 microseconds) all active Crypto Express queues for responses, for both coprocessors and accelerators (Figure 1):

![Earlier ICSF releases](image1)

This scheme contained multiple drawbacks. First, it imposed additional overhead on the system and additional CPU utilization by continually polling for responses on all Crypto Express queues. Second, response time suffered because the response retrieval subtask sometimes had to scan through all other Crypto Express queues before it reached the one that actually had a response. Third, requests sometimes had to wait an unnecessary extra 500 microseconds for the next polling interval to find their finished work. And fourth, only one response retrieval subtask existed for all Crypto Express queues, which didn’t allow for response retrievals to be performed on multiple Crypto Express cards in parallel.

![New interrupt design for Crypto Express response retrieval](image2)

This is why HCR7780 has added Crypto Express Interrupt support to leverage the architectural extension for Crypto Express I/O Interrupts introduced with the z/10 General Availability 2 (GA2).

The Crypto Express I/O Interrupt support provides ICSF with a notification each time a cryptographic request completes. These notifications remove the need to continually poll all active Crypto Express cards for responses and it allows ICSF to process a response as soon as it’s ready. No more looking around for responses and processing them one at a time.

![A quick overview of the Crypto Express I/O Interrupt support](image3)

So how does this work?

1. ICSF initialization registers a sub-second level interrupt handler (sub-SLIH) with IOS and a notification byte for each Crypto Express card.
2. When the Crypto Express completes a request, the Crypto Express Interrupt Hardware Facility (starting with the z10 processor GA2) sets a notification byte and notifies IOS. (The z/OS support for this processing is in z/OS V1R11 and later.) IOS then passes control to the ICSF sub-SLIH.
3. The ICSF sub-SLIH scans its notification bytes to determine which Crypto Express cards are ready for response retrieval. As it finds notification bytes set, it releases an associated response retrieval subtask to process the response.
4. ICSF attaches a unique response retrieval subtask for each Crypto Express card that allows response retrieval on multiple cards to run in parallel.

This new support drives cryptographic coprocessors and accelerators to 100% utilization for some workloads.

If ICSF determines the required hardware and IOS support are not available for Crypto Express Interrupt processing, the ICSF uses the polling method for processing. See Figure 2.
Client application
Call ICSF.

Notification bytes
ICSF sub-SLIH
- Determine which Crypto Express cards require attention.
- Release only the response retrieval sub-tasks for the Crypto Express cards requiring attention.

Crypto Express Interrupt Hardware Facility

ICSF Crypto Express I/O service handler
- Prepare an I/O request.
- Select a Crypto Express card.
- Start the I/O.
- Pause awaiting I/O completion.
- Handle client response.

Response Retrieval sub-task #1
Loop forever:
  - Pause awaiting completed I/O.
  - Loop until Crypto Express queue is empty:
    - Retrieve a response.
    - Release waiting client.

Response retrieval sub-task #2
Response retrieval sub-task #3
Response retrieval sub-task #4

1. Client requests an ICSF function that requires a Crypto Express card.
2. Service sends a request to the card.
3. Client requests an ICSF function that requires a Crypto Express card.
4. Crypto Express Interrupt Hardware Facility sets the notification byte.
5. IOS gets control for interrupt and calls ICSF sub-SLIH.
6. ICSF sub-SLIH interrogates notification bytes and releases the response retrieval subtask.
7. The response-retrieval subtask retrieves response from card and releases the client with results.

Figure 2. Overview of ICSF interrupt design for Crypto Express response retrieval
Moving the in-storage CKDS cache “above the bar”

Earlier releases of ICSF contained a CKDS size limitation caused by reading the CKDS into 31-bit private storage that prohibited ICSF users from growing their CKDS data set to millions of keys.

HCR7780 function in z/OS V1R10 - V1R12 addresses the CKDS size limitation by moving the in-storage copy of the CKDS to above 64-bit storage. The new implementation uses the 64-bit cell pool service (added in z/OS V1R10 and later) to manage the 64-bit storage used for the in-storage cache and a balanced binary tree to keep the in-storage copy in sorted order at all times.

This support provides benefits over earlier releases of ICSF that used an array of sorted records, followed by a sequential list of updated records. The sequential list at the bottom of the array required a linear search, which degraded performance and required periodic CKDS refreshes to be performed by the user. Because the implementation of the binary tree is always maintained in sorted order, it removes the need to perform a periodic refresh and thus helps minimize performance problems.

With the move of the in-storage copy of the CKDS to above 64-bit storage, the size of the CKDS is large enough that you probably don’t have to worry about it for years to come. However, installations need to understand and plan for the necessary system resources that are required to manage the in-storage cache of the CKDS data set. Scaling a CKDS to contain millions of keys requires system central storage to be scaled up to back the in-storage CKDS cache.

For more details on ICSF system resource planning for the CKDS, see the z/OS ICSF System Programmer’s Guide, SA22-7520.

Faster is better

With this new ICSF support you can expect to see performance benefits to your cryptographic key data set (CKDS) update processing and cryptographic coprocessor and accelerator throughput. All clichés aside, faster is better for your cryptographic processing needs.
It was hard to choose our three favorite problem management tools. After all, we work with many products, but these stand out because they can benefit your installation and possibly prevent a major outage.

IBM Health Checker for z/OS

What happens when you have a properly configured system? Nothing out of the ordinary when you are using IBM Health Checker for z/OS. Use it to identify potential z/OS configuration problems or migration issues before they affect your installation’s availability. There are new checks in every release of z/OS. For example, z/OS V1R12 introduces checks like:

- ZOSMIGV1R12_INFOPRINT_INVSIZE: This migration check verifies that Infoprint Server has enough space to create Version 2 Printer Inventory files.
- XCF_CF_MEMORY_UTILIZATION: This check, among the many valuable XCF checks, raises an exception when a coupling facility reaches the memory utilization threshold. You never want the percentage of memory use in a coupling facility to approach an amount high enough to prevent it from allocating of new structures or expanding existing structures.
- SMS_CDS_SEPARATE_VOLUMES: This check tells you when the active control data set (ACDS) and the communications data set (COMMDS) reside on the same volume. In the event of a failure, you will be thankful the ACDS and the COMMDS reside on separate volumes.

The IBM Health Checker for z/OS goes beyond problem avoidance and detection. Here is an example from z/OS V1R11:

IOS_MIDAW: This check verifies that the modified indirect addressing word (MIDAW) facility is in use, which can help your installation can make better use of I/O bandwidth. Ah…performance!

Predictive Failure Analysis

What is more effective than fixing a problem? If you are using Predictive Failure Analysis (PFA), you can treat the symptoms. A soft failure is uncommon software behavior that is typically undetectable until it causes a critical problem. You might have heard a soft failure referred to as a “sick, but not dead” system…not the kind of problem anyone wants. Luckily, PFA continues to develop checks to detect the various types of symptoms related to soft failures. For example, z/OS V1R12 introduces checks like:

- PFA_SMF_ARRIVAL_RATE: This PFA check determines if there is any potential of damage to an LPAR by checking the arrival rate of SMF records per number of CPU seconds on a system.

In z/OS V1R10 and V1R11, PFA introduced these checks:

- PFA_COMMON_STORAGE_USAGE
- PFA_FRAMES_AND_SLOTS_USAGE
- PFA_LOGREC_ARRIVAL_RATE
- PFA_MESSAGE_ARRIVAL_RATE

Runtime Diagnostics

You might have to issue innumerable commands and review countless logs when you suspect your system is acting oddly. Using Runtime Diagnostics, introduced in V1R12, makes quick work of identifying the symptoms of soft failures. After you perform the configuration of Runtime Diagnostics, anytime you suspect an issue, enter the START HZR command.

Runtime Diagnostics runs a quick analysis, and returns results to narrow down the scope of possible symptoms. Because it runs so quickly, there is no spike in system resources. Best of all, the output gives you suggestions about what to do next.

For example, you suspect there is a problem and issue S HZR. After about thirty seconds, Runtime Diagnostics returns a message stating there is possible ENQ contention giving you the exact job name of the waiter for ENQ resource. That’s just one example. There are many more.

Suggested reading

Find out how to be the most highly effective z/OS system programmer, and possibly teach your operators too, by reading the following publications:

- z/OS V1R12 Problem Management, G325-2564-07
Loosely-coupled applications that exploit XML document-based technologies for synchronous program interactions are best suited for coarse-grained interactions between disparate systems communicating over network protocols. Such applications have traditionally been considered ill-suited for interactions between applications co-located on the same z/OS system image. For example, using Web services and the SOAP protocol to exchange structured information between a CICS application and an application running under WebSphere® on the same LPAR would be inefficient because of the overhead of network stacks and protocol handlers. But now, thanks to WebSphere Optimized Local Adapters (WOLA), these overhead concerns are a thing of the past!

Introducing WOLA

WOLA is a functional component of IBM WebSphere Application Server introduced in Version 7.0. It provides a built-in high-speed, bi-directional connector for calling between applications running on WebSphere z/OS and applications in other address spaces on the same z/OS image. The WOLA APIs provide a new high-performing mechanism to support efficient integration of newer Java and Java EE applications with older native language-based z/OS applications that run in address spaces outside the WebSphere Application Server z/OS server but in the same logical partition (LPAR). This includes programs running in:

- Customer Information Control System (CICS)
- z/OS batch
- z/OS UNIX Systems Services
- Airline Control System (ALCS)
- Information Management System (IMS)

As shown in Figure 1, WOLA contains z/OS-specific optimizations that provide very low latency. It does this by utilizing a feature of WebSphere Application Server z/OS called “local communications” or “local comm,” which is based on z/OS cross memory services. Local-comm has actually been part of WebSphere Application Server z/OS since its early days, and is used for communicating between WebSphere Application Server z/OS servers in the same system. With WOLA, WebSphere Application Server is now extended to provide this cross-memory capability to virtually any z/OS address space. It provides security context propagation in each direction and provides identity assertion for the CICS and IMS environments. It also provides two-phase commit transactions support for calls between WebSphere Application Server and CICS (available in both directions).

The WOLA APIs

The WOLA APIs have no pre-requisites beyond placing the data set that contains them in the STEPLIB/LINKLIST of the job that is to call them. They are callable from COBOL, C/C++, PL/I, and High Level Assembler languages. To make calls from WebSphere Application Server to external address spaces, the programming model used by WOLA is Version 1.5 of...
the J2EE Connector Architecture (JCA). Because WOLA is a local-only technology, address spaces communicating with WebSphere Application Server must be on the same LPAR. Calls from WebSphere Application Server applications to those in an external address space can be from Enterprise JavaBeans (EJB) or a servlet. Inbound calls to WebSphere Application Server applications must target a stateless session bean that implements the WOLA classes and a preset execute() method name.

Installation, customization, and serviceability

You can use a shell script called olaInstall.sh to install WOLA in an existing WebSphere Application Server z/OS file system. This script also copies the WOLA API load modules from the WebSphere Application Server file system into a user-provided load library. A Python script, olaRar.py, is provided to tailor WebSphere Application Server and to get it ready to run the WOLA sample programs. For serviceability, WOLA provides robust and dynamic tracing support. WOLA manages a new shared 64 MB above-the-bar buffer that contains WOLA trace data. This trace data is good for debugging system and transaction work processing through WOLA. WOLA also has the ability to set different trace levels for different applications or one default setting for an entire WebSphere Application Server cell.

Extras for CICS and IMS

WOLA provides additional built-in capabilities for CICS and IMS environments including the following:

- For CICS, WOLA provides support for calling existing unchanged CICS programs using the WOLA CICS Link Server. Data is passed to/from these applications using CICS COMMAREA or CHANNEL/CONTAINER.

- For calling existing unchanged IMS applications from WebSphere Application Server, WOLA provides support for calling over IMS Open Transaction Manager Access (OTMA).

Making WebSphere Application Server z/OS more accessible from earlier environments

WOLA extends the WebSphere z/OS stack (including the EJB/Web containers, Web services, and Web 2.0 support) to classic application environments like batch z/OS, CICS and IMS. With a simple proxy implemented in WebSphere Application Server, a batch z/OS application can begin interacting with web-based applications in both directions.

For more information on WOLA, see the IBM Information Center or the Techdocs Library page located at the following website:

ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101490

Practice safe sets (data)

DB2 V10: A new standard in data protection

BY MARK NELSON, RANDY LOVE, AND GAYATHRI CHANDRAN

Ever since its introduction in 1983, DB2 for z/OS has had a rich set of features that allows clients to effectively control access to data. Over the past 27 years, these features have evolved to help you meet emerging data protection needs and regulatory requirements. With DB2 Version 10, IBM is introducing a new set of security features in DB2 that allow for a greater separation of duties, reduce the amount of authority that has to be granted to get a specific task done, and further controls for accessing production data.
Fine tuning with new system privileges

The first of these new features is a set of administrative authorities that allow for a much finer granularity when granting system-level privileges. These new authorities allow the user to:

**SECADM**: Manage security-related objects in DB2 and control access to all DB2 resources. The SECADM authority is assigned to users through DB2 system parameters.

**System DBADM**: Manage most objects in a DB2 subsystem without having the ability to access data or control access to data.

**DATAACCESS**: Access data in all user tables, views, and materialized query tables. The DATA ACCESS authority also allows executing plans, packages, functions, and procedures in a DB2 subsystem.

**ACCESSCTRL**: Grant all authorities and privileges, except system DBADM, DATAACCESS, ACCESSCTRL, and privileges on security-related objects, such as sensitive triggers and user-defined functions (UDFs).

**SQLADM**: Monitor and tune SQL without any additional privileges.

In addition, DB2 V10 introduces a new privilege, EXPLAIN, which allows the user to issue EXPLAIN ALL, PREPARE, and DESCRIBE statements without requiring the privilege to execute the SQL statement.

**Separation of duties for security administration**

One of the new DB2 V10 features allows an installation to configure DB2 to prevent users with SYSADM authority from altering authorizations, thus restricting security-related work to SECADM users. You can accomplish this by setting the field SEPARATE SECURITY ZPARM to YES on the DSNTIPPI panel, as shown in Figure 1. When SEPARATE SECURITY is set to YES, the SYSADM and SYSCTRL authorities cannot be used to affect the security characteristics of the system. Specifically:

- The SYSADM authority does not allow the management of security objects, such as roles and trusted contexts.
- The SYSCTRL authority does not allow the management of roles.
- The SYSADM and SYSCTRL authorities cannot perform grants and cannot revoke privileges granted by others.

The installation SYSADM user, however, is not affected by the SEPARATE SECURITY setting and can manage security objects and perform grants and revokes.

**Controlling cascading revocation of privileges**

You can use the new REVOKE_DEP_PRIVileges system parameter to prevent any revoke operation from cascading (that is, revoking privileges granted to others). The values supported are YES, NO, and SQLSTMT. A new dependent privileges clause, INCLUDING DEPENDENT PRIVILEGES or NOT INCLUDING DEPENDENT PRIVILEGES is also added to the REVOKE statements. The REVOKE_DEP_PRIVileges value of SQLSTMT allows revoke behavior to be specified at the SQL statement level. When you specify SQLSTMT, the default behavior is to cascade. When ACCESSCTRL, DATAACCESS, or system DBADM authority is revoked, the default is always NOT INCLUDING DEPENDENT PRIVILEGES, and you must specify the clause on the REVOKE statement.

**Row and column access control**

Views allow the restriction of entire rows and columns to users, allowing information within a table to be subset to users based on values within the table. With V10, DB2 introduces a new mechanism called “row and column access control,” which restricts access to the table based upon individual user permissions and masks as specified by a policy associated with the table. All users including the administrators are subjected to these new access controls.

The CREATE MASK statement allows a user to create a column mask object that expresses a column access control for a specific column. Consider the SQL statement in Figure 2. In this example, the column “SSN” (social security number) is returned as it is in the table if the requestor is associated with the auth ID ‘PATIENT’, ‘PCP’ (primary care physician), ‘MEMBERSHIP’, or ‘ACCOUNTING.’ For all other requestors, the value that is returned is XXX-XX-XXXX, where XXX is the last four characters of the information in the SSN column.
The CREATE PERMISSION statement allows a user to create a row permission object that describes the conditions to access the rows of data. In Figure 3, the row is returned if the role associated with the request is ‘PATIENT’ and the user is looking at his own record, or the role is ‘PCP’ and the requestor is the PCP in the row.

```sql
CREATE MASK NETHMO.SSN_MASK ON HOSPITAL.PATIENT
FOR COLUMN SSN RETURN
CASE WHEN VERIFY_TRUSTED_CONTEXT_ROLE_FOR_USER(SESSION USER, 'PATIENT')=1 OR
WHEN VERIFY_TRUSTED_CONTEXT_ROLE_FOR_USER(SESSION USER, 'PCP')=1 OR
WHEN VERIFY_TRUSTED_CONTEXT_ROLE_FOR_USER(SESSION USER, 'MEMBERSHIP')=1 OR
WHEN VERIFY_TRUSTED_CONTEXT_ROLE_FOR_USER(SESSION USER, 'ACCOUNTING')=1
THEN SSN
ELSE 'XXX-XX-' SUBSTR(SSN,8,4)
END
ENABLE;

CREATE PERMISSION NETHMO.ROW_ACCESS ON HOSPITAL.PATIENT
FOR ROWS WHERE(
(VERIFY_TRUSTED_CONTEXT_ROLE_FOR_USER(SESSION USER, 'PATIENT')=1 AND
HOSPITAL.PATIENT.USERID = SESSION_USER) OR
(VERIFY_TRUSTED_CONTEXT_ROLE_FOR_USER(SESSION USER, 'PCP')=1 AND
HOSPITAL.PATIENT.PCP = SESSION_USER))
ENFORCED FOR ALL ACCESS
ENABLE;
```

Figure 2. Creating a MASK

Figure 3. Controlling row access

After the permissions or masks are created for a table, you must activate them by using the ALTER TABLE statement for the changes to take effect.

Unlike VIEWs, which can be created by anyone with SELECT authority on the table, PERMISSIONs and MASKs are security objects and can only be created by users with the SECADM authority or by users with the SYSADM authority only if SEPARATE_SECURITY=NO has been specified.

Audit enhancements

DB2 V10 introduces new audit capabilities with the introduction of audit policies and categories. The audit policies provide the flexibility and extended function to better monitor the security plan. You can define different audit policies based on the security needs of your business. Some points to keep in mind:

- You can define audit policies on any table or tables in a schema without having to make any DDL changes or rebinding of applications. These audit records are now generated for all read and write access to a table, not just first access in a transaction.

- Audit policies can generate an audit trail of any user who uses one of the system authorities, such as SYSADM, DATAACCESS. The system writes the audit record only when the audited authority is used for access.

An audit policy is created by inserting a row in the new catalog table, SYSIBM.SYSAUDITPOLICIES. You can then enable the audit policy by issuing the START TRACE command with the audit policy name. You can disable the audit policy by issuing the STOP TRACE command with the audit policy name.

Using RACF with DB2 V10

The RACF® access control module, available in SYS1. SDNSAMP(DSNXRXAC), has been updated with DB2 V10 to support these new administrative authorities. The new administrative privileges are implemented as checks in the existing DB2 general resource classes as shown in Figure 4.

<table>
<thead>
<tr>
<th>DB2 authority</th>
<th>RACF general resource class</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESSCTRL</td>
<td>DSNADM</td>
<td>db2-subsystem.ACCESSCTRL</td>
</tr>
<tr>
<td>DATAACCESS</td>
<td>DSNADM</td>
<td>db2-subsystem.DATAACCESS</td>
</tr>
<tr>
<td>EXPLAIN</td>
<td>DSNADM</td>
<td>db2-subsystem.EXPLAIN</td>
</tr>
<tr>
<td>SECADM</td>
<td>DSNADM</td>
<td>db2-subsystem.SECADM</td>
</tr>
<tr>
<td>SQLADM</td>
<td>MDSNSM</td>
<td>db2-subsystem.SQLADM</td>
</tr>
<tr>
<td>System DBADM</td>
<td>DSNADM</td>
<td>db2-subsystem.SYSDBADM</td>
</tr>
</tbody>
</table>

Figure 4. DB2 administrative privileges in RACF

But that’s not all...

Don’t think for a minute that this is all that there is to DB2 V10. Many other new features are available, including the following support:

- Distributed identities that were introduced in z/OS V1R11
- Client digital certificates that were introduced in z/OS V1R10
- Password phrases that were introduced in z/OS V1R8
- Support for temporal table to automatically maintain different versions of the data.

DB2 enhanced security for better data protection

DB2 V10 goes a long way to ensure that your data is well protected. For more information about these functions and all of the other DB2 V10 enhancements, please visit the following website:

ibm.com/software/data/db2/zos/db2-10/
It's a wrap!

ICSF supports X9.24-compliant CBC wrapping for DES keys

BY ELEANOR CHAN, KATHLEEN FADDEN, AND KEN KERR

Cryptographic Support for z/OS V1R10 - R12 (FMID HCR7780) introduces ICSF support for the ANSI X9.24-1 standard for key management and use of symmetric keys. The standard requires the key value in a token to be bundled with other token data and encrypted with Triple DES using CBC mode. ICSF refers to this new X9.24-compliant CBC wrapping method for DES keys as the “enhanced” key wrapping method. It is for wrapping DES key tokens only.

Key tokens wrapped with the enhanced key wrapping method can coexist in a CKDS with existing DES keys, which are wrapped using ECB mode—ICSF’s original wrapping method. At startup, you can specify the default wrapping method. You can also override the default wrapping method when calling services to generate or manage symmetric keys.

The enhanced key wrapping method is supported only with cryptographic support for z/OS V1R10 - R12 with a Crypto Express3 Coprocessor (CEX3C) running the September 2010 or later licensed internal code (LIC). For this reason, there are some considerations for sharing a CKDS containing enhanced wrapped keys with earlier releases of ICSF.

Installation options data set

To specify the default wrapping method, use the new DEFAULTWRAP parameter in the installation options data set. Using this parameter, you can specify the default wrapping method to be either the new enhanced CBC wrapping method, or the original ECB wrapping method. You also specify the default wrapping method for internal and external tokens independently.

If the DEFAULTWRAP parameter is omitted from the installation options data set, the wrapping method will default to the original ECB wrapping method for both internal and external tokens.

CKDS conversion utility

ICSF provides a new utility program (the CSFCNV2 utility) that will rewrap all encrypted DES tokens in a CKDS. Using this utility, you can rewrap all encrypted DES tokens in the CKDS using the enhanced or the original wrapping method. All existing and rewrapped key tokens will be written to a new CKDS.

ICSF callable services

ICSF callable services support both methods of key wrapping for DES key tokens. Some callable services support new keywords USECONF, WRAP-ECB, WRAP-ENH and ENH-ONLY for generated or updated key tokens:

- USECONF specifies use of the wrapping method set by the DEFAULTWRAP keyword in the installation options data set.
- WRAP-ECB specifies use of the original wrapping method.
- WRAP-ENH specifies use of the enhanced wrapping method.
- ENH-ONLY, when specified with WRAP-ENH, prevents an enhanced wrapped key token from being rewrapped using the original wrapping method. The ENH-ONLY keyword causes ICSF to modify the control vector of the key token. A token with the ENH-ONLY bit in the control vector can only be used on systems that support enhanced key wrapping.

Coexistence and toleration

If you will be sharing a CKDS containing enhanced wrapped keys with earlier ICSF systems, you will need to install the PTF for toleration APAR OA33320.

The PTF for toleration APAR OA33320 will allow earlier ICSF systems to:

- Use a CKDS containing enhanced wrapped tokens
- Refresh a CKDS containing enhanced wrapped tokens
- Read a key label of an enhanced wrapped token from the CKDS.

However, the earlier ICSF systems will not be able to:

- Use an enhanced wrapped token in a callable service. The callable service will fail with a return code 8 and reason code 3079 (X’C07’).
- Use a key management callable service to create, delete or overwrite an enhanced wrapped token in the CKDS.
- Use KGUP to create, update, rename or delete an enhanced wrapped token from the CKDS.
- Reencipher a CKDS containing enhanced wrapped tokens. The master key change/reencipher must be performed on a system with Cryptographic Support for z/OS V1R10 - R12 and a CEX3C running the September 2010 or later LIC. All earlier systems sharing the CKDS will be able to load the same master key and perform a change master key to activate the reenciphered CKDS.

For more information

You can find more information on the enhanced wrapping method in the ICSF documentation:

- For the installation options data set and DEFAULT keyword, see z/OS Cryptographic Services ICSF System Programmer’s Guide, SA22-7520-15.
- For the CSFCNV2 conversion utility, see z/OS Cryptographic Services ICSF Administrator’s Guide, SA22-7521-15.
- For ICSF callable services, see z/OS Cryptographic Services ICSF Application Programmer’s Guide, SA22-7522-14.
Fast! Faster! Faster!

DFSMShsm Fast Replication recovery

BY YUFEN DAVIS AND HENRY VALENZUELA

The IBM DFSMShsm fast replication feature has been steadily enhanced since its initial introduction in z/OS V1R5 as part of a joint application-based continuous data protection (CDP) solution between IBM DFS™MS and IBM DB2. The zCDP for DB2 solution is described in the article “Safety first” in z/OS Hot Topics Newsletter Issue 16, February 2007, GA22-7501-12. In this article, we want to update you on our recent enhancements to fast replication backup and recovery.

Simplified and improved functions
The following sections provide a summary of the enhancements.

Recovering data with saved catalog information
You can use fast replication to recover data sets from a copy-pool backup version. However, before z/OS V1R11, you might have been frustrated when, if the data sets had been moved or deleted since they were backed up, recovery would fail. Beginning with z/OS V1R11, DFSMShsm optionally saves catalog information at the time of the backup in a catalog-information data set managed by DFSMShsm. The saved catalog information is then used to locate and recover the moved or deleted data sets to their original volumes.

Part of the procedure in fast replication recovery included manually unallocating catalogs residing on the copy pool volumes before issuing a volume-level recovery command. Now, when the catalog information is saved during backup, DFSMShsm automatically unallocates the catalogs before volume recovery, simplifies the recovery procedure, and reduces the chance for user error.

Volume preferencing
For optimal FlashCopy® and application performance considerations, you might have been carefully manipulating your fast replication environment on the IBM System Storage® DS8000® series to ensure same-server, source-target volume pairing. You can now relax and let DFSMShsm do the volume preferencing work dynamically during fast replication volume selection. The PTF for APAR OA32494 provides the volume preferencing function in z/OS V1R10 through z/OS V1R12.

Multi-task recovery from dump
If you’ve ever tried to recover a copy pool from tape, you might have asked, “Why can’t I do this with a single command and why is the recovery single threaded?” Well, not anymore. Beginning with z/OS V1R12, DFSMShsm supports up to 64 concurrent-volume recovery tasks for a dump initiated from a single FRBACKUP command. Of course, it still won’t be as fast as recovering from DASD.

Volume recovery tracking
Sometimes a copy pool recovery from tape might be interrupted. Don’t worry. The recovery doesn’t have to start from scratch. In z/OS V1R12, DFSMShsm keeps track of which volumes have completed recovery and when the recovery is restarted. DFSMShsm does not attempt to recover volumes that have already been completed. Working hand-in-hand with this feature is the fast reverse restore feature, which also takes advantage of volume recovery tracking so that when any copy pool recovery is interrupted, it can be restarted from DASD or tape.

Tape mount optimization
In addition to multitask volume recovery and copy-pool recovery tracking, tape mount optimization has been added in z/OS V1R12 where recovery processing scans for queued volume recovery requests that need the currently mounted tape, and processes them without demounting the tape. With the reduction of tape mounts and support for up to 255 stacked dumps on a single tape, there can be significant time and processor savings. However, consider stack limits and volume-recovery tasking levels when you create your backups to avoid underutilizing available tasks because of high stacking values. In Figure 1, recovering version 1 utilizes 10 concurrent tasks while recovering version 2 is limited to one task but uses only one tape for the backup.
FlashCopy options
DFSMShsm™ uses the following FlashCopy options when the required licensed features are available on your storage devices:

- Background COPY
- NOCOPY
- Multiple relationship FlashCopy
- Full copy
- Incremental FlashCopy
- FlashCopy to metro mirror primary or global copy primary
- Preserve mirror (also known as remote pair FlashCopy)
- FlashCopy SE
- Reverse restore
- Fast reverse restore.

Some of these features are described in the following sections.

Preserve mirror
You might want to combine FlashCopy with other copy services for your disaster recovery (DR) solutions. DFSMShsm supports the option to allow FlashCopy to mirror primary volumes using the metro mirror method. If your DR environment is sensitive to the condition that the metro mirror pair enters a duplex pending state, consider using the preserve mirror option.

Preserve mirror is also known as remote pair FlashCopy.

Instead of transferring data through the metro mirror link, the remote pair FlashCopy function sends an in-band FlashCopy command to the remote site. Therefore, the metro mirror pairs remain in full duplex and the mirror is preserved. The preserve mirror option cannot be used in combination with either FlashCopy SE or fast reverse restore options. Additional preserve mirror configuration requirements must be met to perform the operation.

When you enable one of the preserve mirror options or choose the existing support for the copy pool, DFSMShsm selects appropriate FlashCopy target volumes that meet the specified configuration requirements. Similarly, backup and recovery processing uses the specified corresponding options.

FlashCopy SE
Because space efficient (SE) volumes do not consume physical space until data is written to them, they are suitable as NOCOPY FlashCopy targets where there are few updates to the source volumes, or when the FlashCopy relationship is short-term in nature.

In z/OS V1R12, when you add space efficient volumes to the copy-pool backup storage groups defined in a NOCOPY environment, DFSMShsm automatically supports FlashCopy SE. You can create a tape copy from the space efficient DASD copy and you can recover data from the space efficient DASD copy using the fast reverse restore method. The allocated physical space for a space efficient volume is released through volume initialization. To ensure that the allocated physical space is released promptly and that the volume is ready for reuse, DFSMShsm automatically initializes the space efficient backup volume when a NOCOPY backup version is withdrawn, deleted, or replaced.

You can also use FlashCopy SE in earlier releases. The support for z/OS V1R10 and z/OS V1R11 was delivered in the PTF for APAR OA30816 with an enabling patch. However, recovery from a space-efficient backup version requires the fast reverse restore option and, therefore, is not supported before z/OS V1R12.

Fast reverse restore
Before z/OS V1R12, if you wanted to recover your copy pool from a DASD backup version, you needed to be patient if the background copy had not finished. Now, there are choices! Beginning with z/OS V1R12, you can enable copy pools to be recovered using the fast reverse restore option. With fast reverse restore, DASD recovery no longer needs to wait for physical background copy to finish thereby reducing recovery time.

The DFSMShsm fast reverse restore option is supported at the copy pool level. The option can be used in background COPY and NOCOPY environments; it can be combined with the full copy, incremental FlashCopy, and FlashCopy SE options. If a background copy has completed at the time of the recovery, DFSMShsm automatically uses the standard FlashCopy or reverse restore option to recover the copy pool.

Before you rush off to implement fast reverse restore, keep in mind that after fast reverse restore is used, the DASD backup copy are no longer valid. Do you want to wait for background copy to finish and keep the valid backup version, or use fast reverse restore, knowing that the backup version will be invalidated.
after recovery? To help you make an informed decision, we have enhanced the QUERY COPYPOOL command in z/OS V1R12, so that it returns the percent-complete value (PCT-COMP) when a background copy has not finished. The following sample output from QUERY COPYPOOL(CP1) shows that the background copy has not completed:

ARC1820I THE FOLLOWING VOLUMES IN COPY POOL CP1,
   ARC1820I (CONT.) VERSION 003, HAVE AN ACTIVE FLASHCOPY BACKGROUND COPY
   ARC1820I (CONT.) SGNAME   FR-PRIMARY FR-BACKUP  PCT-COMP
   ARC1820I (CONT.) SGRP1 SRC01B     TGT03B     025
   ARC1820I (CONT.) SGRP1 SRC02B     TGT04B     083

ARC1820I THE FOLLOWING VOLUMES IN COPY POOL CP1,
   ARC1820I (CONT.) VERSION 002, HAVE AN ACTIVE FLASHCOPY BACKGROUND COPY
   ARC1820I (CONT.) SGNAME   FR-PRIMARY FR-BACKUP  PCT-COMP
   ARC1820I (CONT.) SGRP1 SRC02B     TGT02B     098

The following sample output from QUERY COPYPOOL(CP1) shows that background copy has finished:

ARC1821I NONE OF THE VOLUMES IN COPY POOL CP1, VERSION
   ARC1821I (CONT.) 003, HAVE AN ACTIVE FLASHCOPY BACKGROUND COPY
ARC1821I NONE OF THE VOLUMES IN COPY POOL CP1, VERSION
ARC1821I (CONT.) 002, HAVE AN ACTIVE FLASHCOPY BACKGROUND COPY

More information
You can find more information in the following publications:
• DFSMShsm Storage Administration, SC35-0421
• DFSMS Using the New Functions, SC26-7473
• IBM Redbooks® DFSMShsm Fast Replication Technical Guide, SG24-7069
• IBM Redbooks DS8000 Copy Services for IBM System z, SG24-6787

Software requirements
• Volume preferencing: See the PTF for APAR OA32494 in z/OS V1R10 - z/OS V1R12.
• Preserve mirror: See the PTF for APAR OA28424 in z/OS V1R10.
• FlashCopy SE: See the PTF for APAR OA30816 in z/OS V1R10 - z/OS V1R11.
• Fast reverse restore coexistence: See the PTF for APAR OA30350.

Don’t be excluded! GRS RNL gotchas

BY JOSEPH GENTILE, TAMMY GARREN, AND STEVEN PARTLOW

D o you remember the anxiety over Y2K? For GRS, the scare nobody anticipated was really for Y2K10 (2010)! After the 2010 ball dropped in Times Square, several z/OS clients saw jobs on their production systems “drop the ball” because of data corruption. Interestingly, the installations having this problem all had one RNL entry in common—an RNL that changed the scope of some temporary data set resources on the local system, leaving the data sets vulnerable to updates from multiple systems at the same time. To solve the problem, installations had to re-IPL each z/OS LPAR in the sysplex with a dynamic ENQ exit that temporarily corrected the scope and kept further ENQs from interfering with the permanent RNL.
Gotcha one: RNL and ISPF DSList syntax confusion

Don’t confuse RNL syntax with the syntax for an ISPF Display Data Set List (DSList) function! RNLs control all ENQ resources, not just data set resources, so the pattern matching is not identical to DSList:

- DSList patterns are data set qualifier based where an asterisk (*) is a wildcard that can be any number of characters (including zero characters) for a single qualifier. The asterisk (*) is similar, except that it also matches any number of qualifiers. DSList patterns also have an implied dot (.) at the end.
- RNL processing treats (*) as any number of characters, including dots(,) so it covers multiple data set name qualifiers.

Not understanding these differences can cause the kind of confusion that led to the Y2K10 RNL problem. In that case, an RNL resource entry of TYPE(GENERIC) RNAME(SYS1*), which is the same as TYPE(PATTERN) RNAME(SYS1*), matched a resource named SYSSyyddd_xxxx when the embedded year changed from 09 to 10. Obviously, the intent of the RNL entry RNAME(SYS1*) was just to target SYS1.xxx data sets, but because the entry did not include the dot qualifier, this pattern snagged an unintended resource. To target SYS1.* data sets resources, your RNL entry should look something like TYPE(GENERIC) RNAME(SYS1.*) or TYPE(PATTERN) RNAME(SYS1.*).

Note that even the simple percent sign (%) character in ISPF doesn’t behave exactly like the RNL question mark (?) character. They both mean any one single character, but an ISPF percent sign (%) cannot match a dot whereas an RNL question mark (?) character can. For example, to include all the resources that match DSList entries for “ABC.*.XYZ”, use the following RNL entries:

| RNLDEF RNL(INCL) TYPE(SPECIFIC) QNAME(SYSDSN) RNAME(ABC.*.XYZ) |
| RNLDEF RNL(INCL) TYPE(PATTERN) QNAME(SYSDSN) RNAME(ABC.*.*.XYZ) |
| RNLDEF RNL(INCL) TYPE(PATTERN) QNAME(SYSDSN) RNAME(ABC.*.*.XYZ,*) |

Figure 1. RNL entries matching DSList entries for “ABC.*.XYZ”

Gotcha two: RNL change practices

After you define your RNL entries correctly, keep an eye out for gotchas you might find when changing the RNLs. You can change RNLs by defining a new GRSRNLxx parmlib member that is either used on the next IPL or changed dynamically using the SET GRSRNL=xx command.

The following tips can help you stay out of trouble:

- The SET GRSRNL command takes effect on all systems in the complex.
- The RNL change initiated by the SET GRSRNL command change doesn’t always complete immediately, it can be delayed by active enqueues that conflict with the RNL change. When a delay occurs, the system issues WTOR message ISG220D prompting the operator to either cancel the command or request summary for the progress of the RNL change. ENQ processing for the affected resources is suspended while an RNL change is in progress, so make sure the operator responds promptly to this message so that the RNL change isn’t unduly delayed. If the operator doesn’t respond quickly, the delay can impact performance, and potentially hang the entire complex. Note that starting with z/OS V1R12, consoles AUTOREPLY cancels any RNL change that is outstanding for longer than the time limit specified in parmlib member AUTORxx. Be sure that the RNL change completed and was not canceled by AUTOREPLY before you update the IEASYxx parmlib member.
- After the SET GRSRNL change is complete, don’t forget to update the IEASYxx parmlib member to point to the new GRSRNL.xx member. If you forget, later on when you try to IPL a new system into the sysplex, it might result in a wait state.

More information on RNLs

- MVS Initialization and Tuning Reference, Chapter 36, “GRSRNLxx (global resource serialization resource name lists),” SA22-7592
- When you issue the SET GRSRNL command to change to a new GRSRNLxx parmlib member, watch out for any exception messages from the GRS checks running in IBM Health Checker for z/OS. You can use these messages to help identify potential problems with the new GRSRNLxx member.

After you define your RNL entries correctly, and implement the RNL change practices we’ve covered, RNL gotchas can never getcha again!
By now you have probably heard raves about the new time and space-saving VSAM CA reclaim function that reduces DASD usage and data set reorganization. You can use this function with catalogs, volume catalogs, IBM DB2 databases, IBM DFSMSrmm™, and IBM DFSMShsm control data sets and more! As introduced by the article “VSAM CA Reclaim: Kiss VSAM KSDS reorganizations good-bye!” in z/OS Hot Topics Newsletter Issue 23, August 2010, GA22-7501-19, CA reclaim allows VSAM and VSAM record-level sharing (RLS) to reuse reclaimed control area (CA) space on DASD for key-sequenced data sets (KSDSs), resulting in reduced space requirements and faster data access.

Getting started
To help you get started, we’ve got answers to questions asked by customers who are already using CA reclaim. The questions cover topics ranging from setup steps, activation, toleration, and use of CA reclaim.

Customer: By default, is the CA reclaim function turned on when I receive it?
IBM: By default, CA reclaim is not turned on in parmlib member IGDSMSxx: CA_RECLAIM(NONE).

Customer: When we ran a LISTCAT job, some data sets show CA_RECLAIM(YES) even though we didn’t set up CA reclaim on the system. Does this mean that if we do enable CA reclaim processing, these data sets are part of CA reclaim unless they are no longer turned on by using the ALTER command?
IBM: Yes, the output for the access method services LISTCAT command shows CA_RECLAIM(YES) because that is the default in the data classes. However, if the D SMS,OPTIONS command output displays CA_RECLAIM=NONE, the data sets are not included in CA reclaim processing. And as you mentioned, you can use the IDCAMS ALTER command to turn CA reclaim on or off for an individual file by specifying the ALTER dsn RECLAIMCA | NORECLAIMCA command. In addition, you can disable CA reclaim processing for the entire system with the SETSMS CA_RECLAIM(NONE) console command.

Customer: It is my understanding that there are two ways to set CA reclaim on for a system: one with the SETSMS command and the other in an IGDSMSxx parmlib member. If I want to use the IGDSMSxx, I need only the CA_RECLAIM(DATACLASS) statement, is that correct?
IBM: Yes, that is true. The statement syntax is: CA_RECLAIM = {NONE|DATACLASS|DATACLASS}

In general, a data set is CA reclaimable only if all of the following conditions are true:

• The data set is a VSAM KSDS or CATALOG.
• The data set is processed on a system running z/OS V1R12 or later.
• CA_RECLAIM(DATACLASS) is specified in either parmlib member IGDSMSxx or in the SETSMS command.
• CA Reclaim(Y) is specified or defaulted to in the DATACLASS for the KSDS during IDCAMS DEFINE or DD define processing, or else a subsequent IDCAMS ALTER command is issued that explicitly specifies RECLAIMCA.

Because the CA reclaim function is the default for data classes, creating the KSDS without DATACLASS or without specifying CA Reclaim(Y) for it is the same as specifying CA Reclaim(Y) for the data class.

Customer: What does the IGDSMSxx parameter CA_RECLAIM DATACLASS actually signify?
IBM: If CA_RECLAIM(DATACLASS) is set in IGDSMSxx or displayed in the output from the D SMS,OPTIONS...
command, it means that the system will use the CA Reclaim (Y|N) option that was specified for the data class, where Y is the default.

How do I know CA reclaim is working?

Customer: Is there a command other than LISTCAT that shows the IGDSMSxx CA_RECLAIM setting for the system, as well as one that shows the data sets that are actually set up to go through CA reclaim processing?

IBM: To display the IGDSMSxx CA_RECLAIM attribute for the system, issue the command D SMS,OPTIONS command. To display information about the setting for an individual data set, either the LISTCAT or DCOLLECT command displays the attributes of the DATACLAS. To determine if CA reclaim processing occurred for a data set, review SMF type 64 records to look at fields SMF64DAU (offset X’7C’) and SMF64RLM (at +X’80’). Figure 1 shows an example of an SMF64 record where four CAs were reclaimed and two CAs were reused.

In the INDEX component of the LISTCAT output, the REC-DELETED field shows the number of CAs reclaimed and REC-INSERTED shows the number of reclaimed CAs reused by CA splits. This gives you an idea of the amount of CA reclaim activity since CA reclaim was turned on for the data set.

In Figure 2, you can see 12 records were reclaimed and five reused from the LISTCAT output of the data set shown in the SMF64 record in Figure 1. The numbers in the LISTCAT and SMF output do not match because SMF record counts are on a job basis.

CA reclaim provides a long-term solution to the KSDS problem with empty CAs in VSAM and VSAM RLS by reclaiming empty CAs for automatic reuse.
Customer: So, if I do a SETSMS command or IPL the system with the CA_RECLAIM(DATACLASS) IGDSMSxx parameter, this will enable CA reclaim for all KSDS data sets that have the data class CA-Reclaim(Y) parameter specified, correct?

IBM: Yes.

Customer: Okay, if I don’t want certain data sets to do CA reclaim, I just use the ALTER command to turn it off for a particular data set. I’m assuming that after the ALTER command processing is completed, it carries across all IPLs until someone alters the data set.

IBM: Your assumption is correct. The ALTER NOCLAIMCA command turns off a bit in the VSAM volume data set (VVDs) of the catalog for the data set. The bit setting is maintained for the life of the data set until either the data set is deleted or the bit is turned on by the ALTER RECLAIMCA command.

Customer: Can I use CA reclaim for SMS and non-SMS KSDS data sets?

IBM: Yes.

Customer: Why should I enable CA reclaim for a data set?

IBM: CA reclaim provides a long-term solution to the KSDS problem with empty CAs in VSAM and VSAM RLS by reclaiming empty CAs for automatic reuse. While CA reclaim is not an online reorganization, it does greatly reduce the need to do online reorganization of the KSDSs and helps achieve 24x7 availability. It helps performance too. In fact, CA reclaim is typically turned on for most KSDSs.

CA reclaim works with any KSDS including catalog data sets, alternate index and base clusters, SMS- and non-SMS managed, and temporary data sets. You cannot use CA reclaim for data set types other than the KSDS or a KSDS defined with IMBED.

To help you decide whether a data set should use CA reclaim, use the IDCAMS EXAMINE DATATEST command that issues message IDC01728I to show the number of empty unreclaimed CAs:

If message IDC01728I shows a large number of empty CAs, the data set might be a good candidate for CA reclaim. If CA reclaim is turned on, it might still show a number of CAs that have not been reclaimed because CA reclaim processing does not reclaim empty CAs from before CA reclaim was turned on.

When will CA reclaim help performance?

The smaller a KSDS index structure VSAM has to traverse to search a record, the faster the VSAM direct and sequential requests are processed. If a KSDS has empty CAs from ERASE requests, CA reclaim removes them from the index structure, shrinking the structure size. In a situation where all the records are erased with CA reclaim, you can reduce processor usage in comparison to the same situation without CA reclaim. During testing, processor utilization was reduced by as much as 99.47% as shown in Figure 3.

When will CA reclaim help performance?

On the other hand, if many of the erased records are reinserted after VSAM has claimed them, you might see performance degradation in Figure 4.

Performance results with CA reclaim can vary depending on how many empty CAs are created by the VSAM ERASE requests and whether or not the erased data records are reinserted. In most cases, you might see the following performance changes:

- If you do issue some VSAM ERASE requests and empty CAs are reclaimed, you should see performance improvements less than 99%.
- If no or few records are erased, enabling CA reclaim does not affect system performance.
Imagine this: You are the quality control person on an assembly line making widgets. You have just finished verifying the correct operation of a batch of 1,000 of these widgets when the plant manager walks up to you and hands you another widget that needs to be inspected, but says, “Besides checking out this one, I need you to re-inspect the thousand that you already checked out because our process requires you to do inspections in batches.” If such an inefficient processing rule would make you cringe, you can appreciate the value of the new fragment parsing support in XML System Services for z/OS.

Parse and perform validation
This support, introduced in z/OS V1R12 and rolled back to V1R10, provides the ability to parse and perform schema validation on just one section, or fragment, of an XML document that consists of a large number of similar repeating fragments. The pattern of XML documents that consists of a list of repeated instances of the same type of fragment is a common one. For example, an XML document might contain a summary of all the accounts in a bank, where the associated schema includes a section that describes the XML requirements for each of the individual accounts.

Assume all the existing accounts in the document were already validated. If you wanted to add another account to the document, but weren’t sure if the new account information was valid relative to the schema, most XML parsers would require you to re-validate the entire document. With the new z/OS XML support, you can now just validate the fragment being added, avoiding the unnecessary re-validation of the fragments that already were in the document.

Context matters
In order to properly validate the fragment before parsing it, the parser must understand the context in which the fragment exists. This check is accomplished by passing a fragment context through the new GXLHXEC_CTL_LOAD_FRAGCONTEXT control operation before it is parsed.

The fragment path represents the path from the root element of the complete document to the root element of the fragment, which consists of prefixes and local names. After loading the fragment context, users need to load an Optimized Schema Representation (OSR) by calling the GXLHXEC_CTL_LOAD_OSR control operation as they would if performing a validating parse on an XML document.

GXLHXEC_CTL_FRAGMENT_PARSE
With the OSR loaded, the caller can enable fragment parsing mode with the GXLHXEC_CTL_FRAGMENT_PARSE control operation. With fragment parsing enabled, the input buffer is expected to begin with specific values. XML system services actually supports parsing two different types of fragments:

- Element fragments (that is, a single element including its descendants). If the element fragment is to be parsed, the input buffer must begin with the < character after an ignorable whitespace.
- Attribute value fragments. If the attribute fragment is to be parsed, the input must begin with either single-quotes (‘) or double-quotes (") after ignorable whitespace.

Calling applications do not need to change output handling for fragment mode parses because the output is similar to what is produced by document mode parses. For example, to parse the first child element in Figure 1, the specified fragment path should contain “/root/child” and the input buffer contain the entire element including the end element tag. To parse the name of the second child, the fragment path contains “/root/child/name” and the input buffer would contain the attribute value including the beginning and ending quotation marks.

OSR requirements
Fragment parsing is only supported with validating parses. You must ensure that the OSR for fragment parsing is generated by the z/OS V1R12 OSR generator, or the alternate OSR generator with the required rollback PTF for APAR OA32251 installed on the system.

Piece of cake
Our widget quality control person should become much more efficient using the new fragment parsing support in z/OS XML System Services, since he or she can now inspect one widget without having to also inspect all previous widgets.
Ask Mr. Catalog
Answers to common catalog questions

BY STEPHEN M. BRANCH (MR. CATALOG) AND TAN Q. NGUYEN (SPECIAL GUEST)

We have some exciting news for you in this installment of Ask Mr. Catalog! In z/OS V1R12, we addressed long standing catalog security requirement MR119084438. Isn’t that great? What, you’re saying that “MR119084438” doesn’t mean anything to you? Then let me put it this way; now you can perform REPRO MERGE CATs without needing RACF ALTER authority to the individual data sets being moved from one catalog to another.

The problem was that REPRO MERGE CAT is not just one command; it’s a series of locates, defines and deletes. Information for an entry is gathered from the source catalog, added to the target catalog with a DEFINE RECATALOG, and finally DELETE NOSCRATCHed from the source catalog. So, in order to perform a REPRO MERGE CAT, a catalog administrator must have RACF ALTER authority to the data set (which means access to the contents of the data sets). Many security administrators are uncomfortable with letting catalog administrators access this data.

Recently, a client pointed out that if we added a facility class similar to STGADMIN.IGG.DELETE.NOSCRATCH, REPRO MERGE CATs could be done without ALTER authority to the data set. STGADMIN.IGG.DELETE.NOSCRATCH permits a user to DELETE NOSCRATCH a catalog entry without RACF authority to the data set.) The client’s idea was to use the existing class STGADMIN.IGG.DELETE.NOSCRATCH in combination with a new facility class for DEFINE RECATALOG to bypass RACF checking. We thought this was a great idea! In fact, we liked it so much that we’ve implemented it in PTF for APAR OA33013.

Actually, my colleague Tan Nguyen was the developer who implemented it. Tan, what can you tell us about PTF for APAR OA33013?

The PTF for APAR OA33013 introduces our new RACF facility class, STGADMIN.IGG.DEFINE.RECAT to bypass DEFINE RECATALOG authorization. When a user has READ authority to STGADMIN.IGG.DEFINE.RECAT, they can perform a DEFINE RECATALOG without any RACF authority to the data set. If the data set is not SMS managed, UPDATE authority to the target catalog is also required.

To perform the REPRO MERGE CAT command without RACF ALTER authority to the individual data sets, the user must have the following RACF authorizations:

• ALTER authority to both catalogs: source and target
• READ authority to the existing RACF facility class: STGADMIN.IGG.DEFINE.RECAT
• READ authority to the new RACF facility class: STGADMIN.IGG.DEFINE.RECAT

(Q) How can I recover a catalog with a damaged index?
(A) Depending on the damage to the index, you might be able to use REPRO NOMERGE CAT to recover the catalog. REPRO NOMERGE CAT opens a catalog for sequential processing as a data set. If the sequence set of the catalog is undamaged, REPRO NOMERGE CAT will be able to copy all of the entries to a new catalog.

If this fails, a forward recovery of the catalog using ICFRU or a similar utility is your only recourse.

(Q) How does the GDG wrap flag work?
(A) Data sets in a Generation Data Group (GDG) have generations numbered 0001 through 9999. There are only four decimal digit positions for generations, so once you reach 9999, the numbers must wrap back to 0001 (there is no generation 0000).

To make sure that the numbers stay in order once they wrap, the system sets a flag that makes the new generation behave as if they were in the ten thousands. For example, if we have generation 9998 and then added three new generations, they’d be numbered 9999, 0001 and 0002. The...
When you add a new generation, the system checks all generations for the wrap flag. Once catalog processing detects that wrap flags are set for all generations, catalog processing turns the flag off for all generations. Using the previous example (where we have generations 9998, 9999, 0001, and 0002), and supposing the GDGLIMIT is 4, another generation would be added as 0003, while generation 9998 would be rolled off. Since 9999 still exists and does not have the wrap flag on, the wrap flags for all generations remain untouched. Now, let’s say we add yet another generation. It would become 0004, and generation 9999 would roll off. All the wrap flags are now on, so all of generations are in ascending sorted sequence without regards to the wrap flag, and the wrap flag is reset off for all generations.

Problems can arise when we create absolute generations before all of the wrap flags are on. This most commonly happens when we recall old HSM migrated versions. Constantly recalling older migrated versions inhibits the wrap flag reset logic and can eventually lead to a condition in which catalog processing cannot tell the intended order of the generations. Avoid recalling older versions if at all possible.

(Q) Can a master catalog be changed without an IPL?
(A) No. The master catalog is determined by the entry in the SYSCATnn member of SYS1.NUCLEUS or the LOADxx parmlib member in SYS.PARMLIB. The member is only accessed at system initialization time. Therefore, to change the master catalog, the member must be changed and the system IPLed.

(Q) Can a master catalog use CDSC? ECS? or is it restricted to ISC?
(A) Yes, the master catalog can use the Catalog Data Space Cache (CDSC). In the past, using CDSC for the master catalog was not advisable because most of the information for a sparse master catalog is already in main memory. I recommend a sparse master catalog, meaning that you have only entries that are user catalog connectors, aliases associated with user catalog connector aliases, and system data sets. The reason I recommend this is that building the Multi-level Alias control blocks during IPL or catalog restart requires reading the entire master catalog sequentially. The larger the catalog, the longer it will take to read it. The In-storage Catalog Cache (ISC) is the default caching method if CDSC is not used. If you choose to store other data set entries in the master catalog, CDSC can prove beneficial.

Extended Catalog Sharing (ECS) and CDSC are not mutually exclusive. Whenever possible, I recommend using ECS because it avoids SYSZVVDS reserves and VVDS reads for every catalog access. This is true for any shared catalog regardless of whether it is a user catalog or a master catalog. In short, ECS improves catalog sharing performance, and CDSC improves catalog entry reuse performance on the same system.

Go to the zFavorites website where you will find the latest information for all of your System z needs including product documentation, software, ISV development marketing info, education, links to FREE downloads, and much much more!
There are a whole bunch of different components to a storage area network fabric. There are cascaded fabric switches, interswitch links, switch ports, control units, control unit ports, wave division multiplexors, host bus adapters and channel extenders — and that’s just a few of them! In addition, some fabric providers have sophisticated algorithms for balancing work, and vendors have different schemes for assigning routes and buffers. With all that complexity built in, what happens when something in the fabric goes wrong? How do you diagnose the problem?

The complexity of the network fabric can make it hard to determine the root of a problem and the symptoms accompanying the problem might not be very clear. Clients using a Fibre Channel connection (FICON®) might see I/O errors in the form of interface control checks or perhaps missing interrupts. And I/O service time degradations might occur without any easily identifiable symptom other than the average I/O service times increasing as reported by z/OS RMF. Out of all that variation, however, one consistent symptom of fabric problems stands out: a high initial command response (CMR) time for some paths to a control unit while the CMR time for other paths to the same control unit are normal.

What is CMR time?

System z I/O architecture, as well as FICON and new System z High Performance FICON channel architectures, have industry leading instrumentation built in to identify the components of I/O service time. Among these components is the CMR time, which represents the round trip through the fabric alone.

RMF reports CMR time two different ways:

- The first way is the Device Activity Report showing the average CMR time for a device over an interval of time. However, in the case where just one or two of the paths to the device have performance issues, the average CMR time for the entire device might be only slightly degraded and thus not very useful.
- The second way is the Control Unit Queuing Report showing I/O measurements by channel path. Fabric problems might be more obvious in this report because it shows the performance of individual channel paths.

Wouldn’t be nice if z/OS would monitor the CMR times for each path to a control unit, and identify and alert the system programmer in real time when a fabric or device problem is causing inconsistent CMR times? Well now it does! After it detects that a control unit has inconsistent CMR times, IOS cannot only alert you, but also present a report showing the information that is needed to narrow down the problem.

Use IBM Health Checker for z/OS!

So, here’s how we’re going to get this information to you...z/OS Health Checker! I’m sure by now you love what IBM Health Checker for z/OS can tell you about the status of your system. If you haven’t tried it yet, you should! And now we’ve added a new check for the IBM IOS component called IOS_CMRTIME_MONITOR.

This check runs at intervals monitoring and analyzing the average CMR time returned by the channel subsystem across all paths to every logical control unit (LCU). The check issues an exception WTO message to alert you to the problem when it finds any control unit with a path that has an average CMR time that is the highest among the other paths to the control unit. and meets both the following conditions:

- The average CMR time for this path is greater than the high CMR time THRESHOLD parameter.
- The average CMR time for this path exceeds n times that of the average CMR time for an attached path with the lowest average CMR time where n is the RATIO parameter value.

You can use check parameters to:

- Set your own high CMR time threshold (THRESHOLD parameter)
- Set your own high ratio limit between the highest average CMR time and the lowest average CMR time among paths to the control unit (RATIO parameter)
The exception WTO message for the check is:

IOSHC112E Analysis of command response (CMR) time detected one or more control units with an exception.

If the check finds that there’s not enough I/O activity on a path to give a useful average CMR time calculation and comparison, that path is not included in the analysis for the interval. That means you don’t need to worry about unusual I/O producing false exceptions.

Accompanying the IOSHC112E exception message, the Command Response Time report shows information about the control units and their associated paths where this condition has been detected. You can use these reports to help determine the source of some of your fabric issues. Figure 1 shows an example of the report.

Automate to respond to fabric problems quickly

The final icing on the cake is to use the check to respond to the fabric problem quickly. For example, you can automate responses on the check exception to alert the system programmer of a fabric problem occurring in real time. There are many ways to automate responses to a check exception, including sending e-mail messages or setting off beepers.

Using the IOS_CMRTIME_MONITOR check, you can foresee possible fabric problems before they have a noticeable effect on your system. To get the check running, all you have to do is IPL with the PTF for APAR OA33367 and start the IBM Health Checker for z/OS! IOS adds the check for you automatically with our recommended default parameters and starts monitoring for fabric problems immediately. How much better can it get?

Using the IOS_CMRTIME_MONITOR check, you can foresee possible fabric problems before they have a noticeable effect on your system.

Find out more

To find out more about the IOS_CMRTIME_MONITOR check, see the PTF for APAR OA33367, which provides the updates to the following publications:

- IBM Health Checker for z/OS User’s Guide, SA22-7994. This book can show you how to set up and use IBM Health Checker for z/OS, including tips on automating off check exceptions.
- z/OS MVS System Messages, Volume 9, SA22-7639.

Using the IOS_CMRTIME_MONITOR check, you can foresee possible fabric problems before they have a noticeable effect on your system.

To find out more about the IOS_CMRTIME_MONITOR check, see the PTF for APAR OA33367, which provides the updates to the following publications:

- IBM Health Checker for z/OS User’s Guide, SA22-7994. This book can show you how to set up and use IBM Health Checker for z/OS, including tips on automating off check exceptions.
- z/OS MVS System Messages, Volume 9, SA22-7639.

Using the IOS_CMRTIME_MONITOR check, you can foresee possible fabric problems before they have a noticeable effect on your system.

Find out more

To find out more about the IOS_CMRTIME_MONITOR check, see the PTF for APAR OA33367, which provides the updates to the following publications:

- IBM Health Checker for z/OS User’s Guide, SA22-7994. This book can show you how to set up and use IBM Health Checker for z/OS, including tips on automating off check exceptions.
- z/OS MVS System Messages, Volume 9, SA22-7639.

Using the IOS_CMRTIME_MONITOR check, you can foresee possible fabric problems before they have a noticeable effect on your system.

Find out more

To find out more about the IOS_CMRTIME_MONITOR check, see the PTF for APAR OA33367, which provides the updates to the following publications:

- IBM Health Checker for z/OS User’s Guide, SA22-7994. This book can show you how to set up and use IBM Health Checker for z/OS, including tips on automating off check exceptions.
- z/OS MVS System Messages, Volume 9, SA22-7639.

Using the IOS_CMRTIME_MONITOR check, you can foresee possible fabric problems before they have a noticeable effect on your system.

Find out more

To find out more about the IOS_CMRTIME_MONITOR check, see the PTF for APAR OA33367, which provides the updates to the following publications:

- IBM Health Checker for z/OS User’s Guide, SA22-7994. This book can show you how to set up and use IBM Health Checker for z/OS, including tips on automating off check exceptions.
- z/OS MVS System Messages, Volume 9, SA22-7639.

Using the IOS_CMRTIME_MONITOR check, you can foresee possible fabric problems before they have a noticeable effect on your system.

Find out more

To find out more about the IOS_CMRTIME_MONITOR check, see the PTF for APAR OA33367, which provides the updates to the following publications:

- IBM Health Checker for z/OS User’s Guide, SA22-7994. This book can show you how to set up and use IBM Health Checker for z/OS, including tips on automating off check exceptions.
- z/OS MVS System Messages, Volume 9, SA22-7639.
Longing for larger catalogs?

BY RICH LUNGERHAUSEN

The z/OS Hot Topics Newsletter has presented several articles in past issues about Extended Address Volume (EAV) and extended addressability storage. In this article, we want to tell you about new enhancements that allow for larger catalogs that exceed 4 gigabytes (GB). With z/OS DFSMS V1R12, new function allows MVS user and master catalogs to exceed 4 GB and to reside in the cylinder managed space that is available on an EAV.

For the basics about EAV, see the following z/OS Hot Topics newsletters:

- “All about EAV” and “The three phases of EAV” in Issue 19, August 2008, GA22-7501-15
- “New EAV free space” in Issue 20, March 2009, GA22-7501-16.

More from your user, master catalogs, and VVDSs!

Now, in z/OS V1R12, DFSMS not only does MVS allow you to exceed the user and master catalog 4 GB size in the cylinder managed space on an EAV, but you can even place your Virtual Storage Access Method (VSAM) Volume Data Sets (VVDSs) in extended addressability. And a 21 cylinder VVDS is a really big VVDS!

The way we were

Without getting too technical, when z/OS V1R10 arrived on the scene, the underlying allocation method for data sets in extended addressability storage was implemented using a couple of new format data set control blocks (DSCBs), specifically the format 8 and 9 DSCBs and was restricted to new VSAM data set definitions. Since then, the types of data sets and the control over their allocation have been evolving.

For example in z/OS V1R11, the EATTR parameter was added for designating extended attribute DSCBs, making it possible to decide by individual data set definition whether the data set should be excluded from extended addressability storage (EATTR value of NO), or placed there if needed or appropriate given specific conditions (EATTR value of OPT for optional).

The way we are: using the EATTR parameter

Besides using EATTR on a DEFINE request for an individual data set, you can also assign the data set an EATTR value using a JCL parameter, as well as through storage management system (SMS) constructs when the system creates the data set.

However, when it comes to catalogs and a VVDS, the only way you can ensure that they are allocated in extended addressability storage is by specifically using the setting of EATTR(OPT) on a define request. Below is an example of how to define a new user catalog that is eligible for placement in extended addressability storage:

```
DEFINE USERCATALOG(NAME(UCAT01) - 
                    ICFCATALOG - 
                    SHAREOPTIONS(3 4) - 
                    STORCLAS(S1P03S02) - 
                    EATTR(OPT) - 
                    VOLUME(CTXOL1) - 
                    CYLINDERS(1 1) - 
                    DATA(CYLINDERS(300 20)) - 
                    ICFCATALOG - 
                    DEFINE CLUSTER (NAME(SYS1.VVDS.V1P0301) - 
                                    EATTR(OPT) - 
                                    VOLUMES(1P0301) - 
                                    NONINDEXED - 
                                    CYLINDER(12 5))
```

Note that we’ve defined this particular catalog with a primary data allocation of 300 cylinders. The allocation must be at least as much as the break point value (BPV) specified for SMS in order for SMS to even consider placing the data set in extended addressability storage. If you define a new catalog with EATTR(OPT) and you don’t want it to reside in extended addressability storage, check the BPV specified in the IGDSMxx member in SYS1.PARMLIB. (For more information on the BPV setting, see z/OS DFSMS Implementing System-Managed Storage, SC26-7407.)

Explicitly defining a VVDS with EATTR(OPT)

We can explicitly define a VVDS with an EATTR(OPT) setting. At this time there is no facility that allows for an implicitly defined VVDS to be created with EATTR(OPT), and it’s unlikely that you would want to do so, because in most cases a VVDS does not require the large amount of space that extended addressability storage can provide. Below is an example of explicitly defining a VVDS that can reside in extended addressability storage:

```
DEFINE CLUSTER (NAME(SYS1.VVDS.V1P0301) - 
                EATTR(OPT) - 
                VOLUMES(1P0301) - 
                NONINDEXED - 
                CYLINDER(12 5))
```

In the environment where this VVDS has been defined, the BPV is set at the default of 10 cylinders. Consequently, our allocation with a primary of 12 cylinders will exceed the BPV, and allocation will almost certainly occur in extended addressability storage. For SMS-managed volumes an explicit VVDS definition must occur when the volume is empty before any other data set allocations are made on the volume. Only for non-SMS volumes is it possible to define a VVDS after other data sets have been allocated on the volume, and so it’s quite likely that even in that case, the VVDS could be placed in the cylinder-managed storage. Ultimately, Direct Access Device Space Manager (DADSM) decides where a given data set is allocated. If there isn’t enough room in extended addressability storage, it might allocate the VVDS in track-managed space, but that would be a pretty unlikely scenario.
Best practices

Best practices suggest that you still segregate applications data sets into separate catalogs. This makes a lot of sense when it comes to disaster recovery, and the ability to get mission critical applications back up and running faster. You don’t want to be in the position of having your system flat on its back because the giant catalog that carries your personnel system also has your online customer transaction server data sets and nothing can come up until everything is restored.

In some environments, usage of the multi-level alias facility has been necessary so that very large catalogs can be split into separate catalogs. For example, if your installation named all production data sets with a high-level qualifier of PROD and a second level qualifier with an application name (like PROD.PAYROLL and PROD.CUSTOMER), a single level alias for PROD might result in a very large catalog becoming full or difficult to manage. In such an environment, using the multi-level alias facility provides a way to segregate the data sets into separate catalogs.

The upshot is that you shouldn’t define larger catalogs simply to avoid using the multi-level alias facility. As long as you ensure that multi-level alias data sets are cataloged in the proper catalog, there’s virtually no performance issue with accessing them.

Wrapping things up

The new facilities in z/OS DFSMS V1R12 provide new “super powers” for catalog managers. Larger catalogs and VVDSs are now not just possible but more easily implemented, making the management process a lot less daunting. Just remember to use a little common sense and continue to follow best practices. If you do, rather than having a red face and a big ‘L’ on your forehead, you’ll more likely have clients viewing you as if you had a big red ‘S’ on your chest!

Defining catalogs to reside in greater than 4 GB storage

In addition to allowing a catalog or VVDS to reside in extended addressability storage, you can also define a catalog with the extended addressable (EA) attribute so that it can exceed the old 4 GB limit. For the catalog to be EA, you must ensure that it is defined as 

extended format

(EF). Remember that EF functions like VSAM striping and compression are not supported for catalogs. The technique for defining EA catalogs is to assign the catalog to an SMS data class that has a data set name type of 

EXT specified as Required or Preferred on the ISMF DATA CLASS DEFINE/ALTER panel. The EA attribute value must be Y (Yes). More information on EA catalogs can be found in the z/OS DFSMS V1R12 version of 

z/OS DFSMS Managing Catalogs, SC26-7409 (look for “Estimating Space for an Extended Format BCS”). Note that you cannot define the VVDS as an EA catalog.

Ready for action: some thoughts on growing your storage

So with that for background now we’re ready to go out and start defining truly gargantuan catalogs and VVDSs. One big advantage to defining a catalog or VVDS in extended addressability storage is that there won’t be a problem if either needs to extend in size. Certainly that’s a major concern, and more than one major outage as been the result of a catalog or VVDS having no more room to grow. So take advantage of placing your catalogs in extended addressability storage when the opportunity is there.

But remember the old adage that “bigger isn’t always better”. Certainly the ability to define larger catalogs has its place, and being able to define a catalog or VVDS in extended addressability storage can help avoid the pesky problem of running out of space. But all those old sound issues about good catalog management and concern for catalog performance don’t automatically go away just because we can now make them bigger.

The big storage picture: management and performance

As you consider defining larger catalogs, keep in mind the big picture with respect to how your installation operates. Now that you can define larger catalogs, you might be tempted, to put all your eggs in one basket (so to speak). Certainly it’s a lot easier to manage one catalog on an EAV volume than 15 catalogs spread across multiple volumes that are not on an EAV. However, if you have multiple applications all utilizing the same catalog, that might not be the best configuration from a variety of perspectives including performance and reliability.
Supersizing your data?

OAM’s the place with the space

BY KEVIN GOLDSMITH

hat’s right, now you can super-size your data in Object Access Method (OAM). Previously, the maximum object size was 256 MB, but now OAM can manage objects up to 2000 MB!

Overview

OAM is part of DFSMSdfp™, a base element of z/OS. Technically, OAM is an access method, but it is also much more. Do you:

• Need a place to store your data for access at some later time
• And have it automatically managed through a life cycle including creating multiple backups on removable media,
• And have it moved up and down a storage hierarchy that you create
• And the storage hierarchy can consist of any combination or disk, optical, and tape destinations depending upon your access requirements,
• And at the end of the life cycle have the data expired and automatically deleted?

It’s a mouthful, but this is what OAM can do for you. An object is an unstructured data string and OAM does not care what it contains—it can be an x-ray, Adobe PDF, scanned image, your entire mp3 collection—whatever content you would like. OAM maintains metadata about each object in an IBM DB2 table that provides an object directory.

With the OSREQ application programming interface (API), you can use OAM to store objects, retrieve objects, query the metadata about objects, change the way the object is managed, or manually delete the object. You can write your own application program or use one of several solutions that already use the OSREQ API, including IBM DB2 Content Manager and IBM DB2 Content Manager OnDemand.

OAM is very flexible and exploits the power in the SMS storage group, storage class and management class constructs, and associated ACS routines to conveniently manage objects throughout their life cycle according to the policy you have established based on your unique requirements.

About larger object sizes

There are several different ways that OAM applications provide a data entity to OAM:

• The data entity is provided in its entirety as a single object to OAM. For example, consider an architectural drawing that has been scanned in black and white and results in a file that is 250 MB in size.
• Multiple small data entities are bundled together and this conglomeration is provided as a single object to OAM. For example, consider check images, each about 4 KB in size, so roughly 65,536 4 KB check images are in one 256 MB object. The application might need to track where each individual entity resides within the OAM object and some applications store locator metadata in the object itself.
• The data entity is larger than the maximum OAM object size and must be separated into multiple OAM objects. For example, consider a 1000 MB digital x-ray image that is divided into four separate 256 MB objects. In this case, the application might need to track the individual OAM objects to later reconstitute the data entity from multiple OAM objects.
• When multiple data entities are bundled together, now many more data entities can be included in each bundle, greatly reducing the number of OAM objects that are needed. With our 4 KB check image example, the number that can be bundled in a single 2000 MB object is more than half a million!
• For a large data entity, rather than having to separate it, it might now fit in its entirety within a single OAM object. If it must be separated, it will result in much fewer OAM objects. That 1000 MB digital x-ray image would now easily fit in a single OAM object.

The amount of data in the world is increasing at a staggering pace. Your cell phone bill can enumerate every call you made and received, new medical procedures can provide detailed digital imaging of your inner workings, and social networking sites can capture all the aspects of your life in words and pictures. OAM is ready to handle this growth in the amount and size of data entities.

Let’s see how the new 2000 MB object size helps in each of the following cases discussed in About larger object sizes.

• When a data entity is provided in its entirety, the size of the object can be increased several fold. That scan of the architectural drawing might have been constrained by the old maximum object size of 256 MB, limiting it to a black and white rendition, and now perhaps that same architectural drawing can be scanned in full color to more accurately preserve its original content.
• When multiple data entities are bundled together, now many more data entities can be included in each bundle, greatly reducing the number of OAM objects that are needed. With our 4 KB check image example, the number that can be bundled in a single 2000 MB object is more than half a million!

Configuration and usage

Now that you know how using larger object sizes can make life easier, there are a few steps that you must take to configure OAM before you can use it.

OAM configuration

Figure 1 illustrates how you can configure OAM to use object sizes up to 2000 MB.

Configuration and usage

Now that you know how using larger object sizes can make life easier, there are a few steps that you must take to configure OAM before you can use it.

OAM configuration

Figure 1 illustrates how you can configure OAM to use object sizes up to 2000 MB.
For step 1, specify the maximum object size on the SUBSYS statement for the OAM1 subsystem in the IEFSSNxx member of parmlib. The SUBSYS statement allows the initialization parameters to be specified for the OAM1 subsystem using the INITPARM keyword. For the maximum object size, you must specify a value for MOS=x.xxxx where x.xxxx is the number of megabytes for the largest object that can be stored in your implementation (up to 2000).

For step 2, for each object storage group in which objects greater than 256 MB will be used in the disk level of the OAM storage hierarchy, specify the DB2 tables for object storage. In z/OS V1R8 and later, you can optionally use DB2 LOB tables to store objects, but if you are going to use objects greater than 256 MB, you must use DB2 LOB tables. You'll need to review your DB2 configuration to ensure that the LOB tables have been defined and OAM has been configured to use the LOB tables (including LOB=x on the SUBSYS statement).

Step 3 is required if objects greater than 256 MB will be used at the tape level of the OAM storage hierarchy. Processing of objects at the tape level of the OAM storage hierarchy is performed in the OAM address space using 64-bit addressable virtual storage so you must specify a MEMLIMIT value to provide adequate virtual storage above the 2 GB bar to the OAM address space. You can use several mechanisms for providing a MEMLIMIT value, but the easiest is to specify REGION=0M on the OAM procedure CBRAPROC. This is the default.

One more related item, however, that simply cannot be overlooked is that because of the increased use of virtual storage, a robust auxiliary paging subsystem must be in place backed by DASD and sufficient real memory.

OSREQ application programming interface
When you have defined your configuration for objects up to 2000 MB in size for use, it is time to get busy on the application programming interface. One of the challenges of storing an object up to 2000 MB in size is the virtual storage demands in the application address space. Not to worry though because the OSREQ API now provides store sequence functions named STOREBEG, STOREPRT, and STOREEND that accommodate the application programming environment. For objects greater than 256 MB, the store sequence functions allow an application program to provide the object to OAM in “parts”. The store sequence is initiated with a STOREBEG invocation to begin the sequence, followed by one or more STOREPRT invocations to provide each part of the object followed by a STOREEND invocation to complete the storage of the object.

With this mechanism, the application address space never has to materialize the entire object at one time.

Availability
Now that you are ready to store even more, here’s the fine print:

- In z/OS V1R10, OAM supports objects up to 2000 MB only at the disk level of the OAM storage hierarchy. This means no backups or transitions are possible for objects greater than 256 MB.
- In z/OS V1R11, OAM added support for objects up to 2000 MB at the tape level of the OAM storage hierarchy, facilitating full support including backups and transition for objects greater than 256 MB.
- Objects greater than 256 MB are not supported at the optical level of the OAM storage hierarchy.

Resources
- For details on specifying a MEMLIMIT value, see z/OS MVS Extended Addressability, SA22-7614.
- For the OAM configuration steps to enable objects up to 2000 MB in size, see z/OS DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support, SC35-0426.
- For details on the OSREQ API, see z/OS Object Access Method Application Programmer’s Reference, SC35-0425.

The following example for an object stored to disk illustrates an application address space that receives data from a client over a network. As each part of the overall object arrives in the application address space, it is provided on a STOREPRT invocation until the entire object has been received. Application processing is similar for an object stored to tape.
Today’s trend is all about consolidating workloads onto the zEnterprise machine to simplify and to help contain costs. For those of you taking advantage of the processing capacity and memory of the zEnterprise, we want to make sure you know we also enhanced allocation software. If, as many customers do, you have tens of thousands of VSAM data sets representing DB2 tables, you should see significant improvements in the time it takes to restart a z/OS image because we shrunk the time needed to allocate and open the data sets you need to get the workload running again.

To improve allocation speeds, we created a test case that replicates DB2 processing in terms of number and type of data sets, catalog settings, allocation open parameters, and more. We compared how the test case behaved accessing the first data set and the 96,000th data set. Then, using this data, we redesigned system algorithms that didn’t measure up, and tested and retested, until the 96,000th data set access performed as well as the first. Finally, we ran our code in a real DB2 installation. Our test environment consisted of one freshly IPLed LPAR with four dedicated CPs running in a single system sysplex with a coupling facility and global resource serialization in STAR mode. The following settings might help you achieve similar results:

- In DIAGxx, include VSM USEZOSV1R9RULES(NO), if possible. This setting modifies virtual storage management to use improved algorithms for getting and freeing storage across the system.
- In GRSRNxx, convert SYSTOC & SYSZVVDs reserves to ENQs. This reduces hardware contention caused by the RESERVE service.
- Enable Enhanced Catalog Sharing (ECS) for the Master Catalog and all User catalogs that contain DB2 data sets. Further, distribute DB2 data sets across multiple user catalogs for better parallelism of accesses.
- In ALLOCxx, enable the z/OS V1R12 MEMDSENQMGMT function. Programs that enable MEMDSENQMGMT will find that it changes the way MVS Device Allocation manages its internal structures for SYSDSN ENQs. (This function is separately enabled by programs such as DB2 to reduce potential intersects with non-IBM code.)
- For DB2 9, install the PTFs for APARs PM00068, PM17542 and PM18557 when available. This allows DB2 to use a ZPARM MAXDS value of 100,000 without requiring updates to the GRSCNFxx parmlib member, and also enables its use of various z/OS V1R12 functions, including MEMDSENQMGMT.

The result from this testing was as dramatic as we had hoped:

**Table 1. Results of allocation testing**

<table>
<thead>
<tr>
<th>z/OS</th>
<th>Options</th>
<th>#DS</th>
<th>Elapsed time</th>
<th>CPU busy %</th>
<th>Throughput</th>
<th>ITR: trans/ CPU-sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.11</td>
<td>w/VSM USEZOSV1R9RULES(NO)</td>
<td>96000</td>
<td>8 min, 12 sec</td>
<td>57.50</td>
<td>195.27</td>
<td>339</td>
</tr>
<tr>
<td></td>
<td>SYSTOC &amp; SYSZVVDs ENQs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ECS enabled for MasterCat&amp;Userscats</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB2 V9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PM00068+PM17542+PM18557</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.12</td>
<td>w/VSM USEZOSV1R9RULES(NO)</td>
<td>96000</td>
<td>4 min, 2.7 sec</td>
<td>73.14</td>
<td>395.55</td>
<td>545</td>
</tr>
<tr>
<td></td>
<td>SYSTOC &amp; SYSZVVDs ENQs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ECS enabled for MasterCat&amp;Userscats</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MEMDSENQMGMT=ENABLE in ALLOCxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB2 V9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PM00068+PM17542+PM18557</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you compare our test case figures for z/OS V1R11 and V1R12 in the ‘Elapsed time’ and ‘CPU Busy %’ columns, you’ll see a big improvement - the elapsed time went down, while the CPU consumed to accomplish that workload went up. The last column in the table shows the improvement clearly: z/OS V1R12 is 60% more CPU-efficient (545-339)/339 and runs in half the time! As a result, z/OS V1R12 on the zEnterprise operates more quickly, makes better use of available CPU to reduce elapsed recovery time, and gets the workload restarted with fewer delays to incoming transactions.

Implementing the settings we’re recommended in this article and installing z/OS V1R12 can help improve your mean time to recovery after an outage, whether planned or unplanned. But when performance is involved, whether it’s cars or computers, remember that “your mileage may vary!”
What is Kerberos?
As the IBM website “A Kerberos Primer” reminds us (ibm.com/developerworks/ibm/library/it-kerbero.html):

"Kerberos was the mythological three-headed dog that guarded the entrance to the underworld. Unless you could get past Kerberos, you could not enter (or leave!) the underworld. In much the same way, Kerberos guards the entrance to services on the networks.”

Most Kerberos implementations are very good in an environment with applications that run on a single system. As an authentication protocol Kerberos identifies principals (users and services) by requiring them to present proof of identity. But when requests for service from the application go to multiple instances of the application on multiple systems, few implementations of Kerberos perform as well as they might.

In z/OS V1R12, z/OS Kerberos allows multiple instances to behave as one when accepting Kerberos requests. Dynamic virtual IP addressing (DVIPA) environments, where multiple systems are viewed as a single image, make this even more important. In this article we will describe how to enable Kerberos support for multiple instances of an application on multiple systems. See Figure 1 for an overview.

Kerberos and encryption
Kerberos was designed with every instance of a server having its own encryption keys. This means that when a client wants to talk to a server, the client needs to know which instance of the server it is going to talk to, and then needs to get the appropriate ticket for that server. This ticket is only usable by the one instance of the server because it is encrypted with the encryption keys that are specific to that instance of the server.

This has prevented Kerberized applications running in a DVIPA environment from taking advantage of the benefits provided in this single system view of multiple systems. A connection destined for a server on one image of a sysplex cannot be redirected to any other image of that sysplex, even when it is the same application, because the receiving server will not know the encryption keys of the other server and the incoming ticket will be unreadable.

New for z/OS V1R12!
When running on z/OS V1R12, you will now be able to configure Kerberos to allow servers to decrypt tickets from other instances of the same application. When your security administrator uses the SKRBKDC started task to decrypt service tickets, the administrator can define RACF profiles to allow the specified user to use encryption keys for other instances of the same application server. This change has been accomplished by enhancing the use of the profiles defined in the RACF KERBLINK class that define how Kerberos principal names are mapped to RACF user IDs.

Here’s how it’s done: Activate the KERBLINK class and issue RACF PERMIT commands against the proper profiles to allow multisystem applications to accept tickets from other instances of the same application. You can activate this support through the Kerberos configuration options.

Here is an example of the Job Control Language (JCL) for the FTP started task SKRBKDC:

```/FTPD EXEC PGM=FTPD,REGION=4M,TIME=NOLIMIT,PARM=(,ENVAR(‘_EUV_ENVAR_FILE=/etc/skrb/ftp.envar’),POSIX(ON) ALL31(ON)/PORT 21)```
The simplest way to configure Kerberos for an application server is to allocate an envar file for that application server. This allows you to modify the configuration at any time without changing the JCL. The JCL shows an FTP example, but it should work for any application server.

### Setting up Kerberos in a multisystem environment

There are two types of setups. The RACF administrator can:

- Use a keytab file for the application server to use on every image.
- Use the SKRBKDC started task for decrypting service tickets.

#### Setup using a keytab file

For the first setup, the RACF administrator maintains a keytab file that contains the key for every instance of the application server whose service tickets that instance might need.

To allow the application to use its own keytab file (and not the system default), specify the following string in the envar file for the application server.

```
KRB5_KTNAME=/etc/skrb/ftp.ktf
```

#### Setup using the SKRBKDC started task

An alternate setup can be better because there is no keytab file to be maintained. In this setup, the RACF administrator needs to have the application server use the SKRBKDC started task by specifying the following string in the envar file for the application service.

```
KRB5_SERVER_KEYTAB=2
```

Also, it's a good idea that the RACF database be shared across all images in the sysplex and that the sysplex is part of one realm for Kerberos implementation in a multisystem environment. Only the instance is allowed to change, and the primary (application name) and realm must be the same.

For example, if your application server (ftp) is running under RACF userid FTP1 on sysplex image SYS1 (hostname sys1.example.com) and you want to allow it to use service tickets destined for the same application server but on sysplex image SYS2 (hostname sys2.example.com), do the following:

1. Issue the PERMIT command:

   ```
   PE ftp/sys1.example.com
   CLASS(KERBLINK) ID(FTP1)
   ACCESS(READ)
   ```

   This command states that userid FTP1 is allowed read access to the principal ftp/sys2.example.com.

When you have issued all the RACF commands and KRB5_SERVER_KEYTAB has been set to 2, your distributed system application should be viewed as a single application in the eyes of z/OS Kerberos.

When running on z/OS V1R12, you will now be able to configure Kerberos to allow servers to decrypt tickets from other instances of the same application.

2. Usually you also want to have the opposite READ allowed as well for `sys1.example.com` (assuming the application server on SYS2 is running under RACF userid FTP2), so issue the following PERMIT command:

   ```
   PE ftp/sys1.example.com
   CLASS(KERBLINK) ID(FTP2)
   ACCESS(READ)
   ```

You now have the basics needed to open your Kerberos applications to a multisystem world. ■
Are you still haunted by RACF “ghost” profiles? These are RACF profiles that were intended to be generic profiles but were created before generic command processing was turned on for the class. As a result, a discrete profile was created instead of a generic profile. What’s worse is that the discrete profile still contains generic characters (such as *, **, and %), so it looks like a generic profile. Because it’s a discrete profile that looks generic, it can be an administrative annoyance after the class is enabled for generic command processing.

We talked about these ghost profiles in “RACF ghost hunters” in z/OS Hot Topics Newsletter Issue 18, February 2008, GA22-7501-14. In that article, we described how to identify and remove ghost profiles. So why are we talking about them again? Because in z/OS V1R12, RACF has introduced new ways to make it easier for you to identify and remove these insubstantial pests.

Warding off phantoms

In z/OS V1R12, RACF makes it easier to identify ghost generic profiles. In fact, you’ll be warned as soon as one might have been created. If a profile containing generic characters is created and generic processing is not enabled for the class, RACF issues the following warning message:

ICH10321I The profile name GHOST.PROFILE1.* contains generic characters, but generics are not enabled for class FACILITY. A discrete profile has been created.

This is a warning to delete the profile now unless you intended to create a discrete profile. The ghost does not have the chance to haunt your system. You can then enable generic processing for the class and create the true generic profile you intended.

Divining apparitions

In z/OS releases before V1R12, a SEARCH command could reveal a ghost profile but the RLIST command would not. This was a source of confusion and one of the reasons these profiles were dubbed “ghosts”. Consider the following RLIST command, for example:

RLIST FACILITY GHOST.PROFILE1.*

In z/OS releases before V1R12, the following message was returned for a ghost profile.

ICH13003I GHOST.PROFILE1.* NOT FOUND

With z/OS V1R12, the same command returns the following message:

CLASS NAME
----- ----
FACILITY GHOST.PROFILE1.* (UNUSABLE)

The (UNUSABLE) tag identifies a ghost profile, and indicates that it cannot be used for authorization checking. The SEARCH command results include the (UNUSABLE) tag as well. For example:

SR CLASS (FACILITY) NOMASK
GHOST.PROFILE1.* (UNUSABLE)
DISCRETE.PROFILE1 GENERIC.PROFILE1.* (G)
** (G)

In the preceding example, two true generic profiles are also shown. These true generic profiles are identified by the (G) tag, but the false “ghost” generic is identified by the (UNUSABLE) tag.

Dispelling spirits

In earlier releases, getting rid of a ghost profile was difficult. You had to disable generic processing, delete the profile, and then reenable generic processing. It was advisable to schedule these steps to minimize risk, because authorization checking might be affected when involving generic profiles sharing the same class or POSIT value.

Now, with z/OS V1R12, you can simply add the new NOGENERIC keyword on the RDELETE command, as shown in the following example, to delete ghost profiles without disabling generics for the class, and without affecting a true generic profile with the same name.

RDELETE FACILITY GHOST.PROFILE1.* NOGENERIC

Who ya gonna call?

Armed with these new capabilities for divining and dispelling ghost profiles, you are ready to defend your system against them. For more information on the commands described in this article, see z/OS Security Server RACF: Command Language Reference, SA22-7687. Let the ghost busting begin!
No need for workarounds

Greatly improved RACF generic profile loading

BY RUSSELL HARDGROVE AND GEORGE MARKOUIZOS

To process an authorization request if a discrete profile does not exist for the resource, RACF creates a list in storage of all generic profiles that could potentially be the covering profile. RACF then searches this profile list to identify the best-fit generic profile that protects the resource. The act of reading the generic profiles from the RACF database and placing them into storage is called generic profile loading.

- If processing an authorization request for a resource in the DATASET class, all generic profiles under the data set high-level qualifier (HLQ) are loaded.
- If processing an authorization request for a resource in a general resource class (that has not been RACLISTed or GENLISTed), all generic profiles for the general resource class are loaded.

Generic profile loading has been a concern for customers whose system performance has been affected when many generic profiles had to be loaded, and, in some cases, repeatedly reloaded. Although some workarounds could be employed to mitigate performance problems, these approaches were often not ideal. For this reason, RACF has introduced changes in z/OS V1R12 intended to improve generic profile loading.

What was the problem?

Each profile list created during generic profile loading is associated with either a data set high-level qualifier (HLQ) or a general resource class. The performance problems that were a concern in releases of z/OS before V1R12 can be attributed to the processing involved in building the profile lists and to situations where they were no longer cached and had to be rebuilt (sometimes repeatedly, which could result in thrashing). Rebuilding a profile list might be necessary because, for each address space, RACF would maintain only four profile lists (the four most recently used). If another profile list had to be built (to process an authorization request for a resource that was not represented by the four cached profile lists), the least recently used profile list would be deleted.

The I/O process used to build a profile list can be time consuming, reading each profile individually and using MVS services (such as the GETMAIN and ENQ macros). If a resulting profile list was long (because of the number of generic profiles) and if it needed to be rebuilt often, the ensuing overhead could be noticeable for the process (such as a batch job or TSO/E logon) and could even affect system performance.

Also affecting performance was the time it took to search the profile list to find the profile that protects the resource. The profile lists were sequentially searched, so a longer list took a longer time to search.

Profile lists were also stored in the extended local system queue area (ELSQA), so ones that listed a large number of profiles might use up this fixed storage. Many large profile lists spread over many address spaces can noticeably affect fixed frame storage.

What could be done?

So, what if you did have a large number of generic profiles, and were experiencing system performance problems that were associated with loading the generic profiles? What could be done? Well, you could:

- Split the RACF database to reduce I/O contention.
- Rename data sets to reduce the number of generic profiles under a single HLQ
- Analyze your existing generic profiles in an effort to reduce their numbers.
- Construct a RACF data set naming-conventions table to rearrange the data set names. By doing this, you can, in effect, change the high-level qualifier of the data sets in an attempt to greatly reduce the number of generic profiles read in to create the profile list.

While these workarounds were useful to help performance, none were perfect solutions. Splitting the database did not address the overhead associated with using the MVS services. Renaming data sets takes great effort, reducing the number of profiles might compromise good (granular) security, and implementing a data set naming-conventions table is a complex task.
What's the solution?
In z/OS V1R12, RACF has made changes intended to improve RACF generic profile loading.

• In earlier releases, a profile list was built by reading each profile individually. Now, thanks to modifications to the ICHEINTY macro in RACF, multiple profiles are read in one I/O operation by retrieving an index block of profiles. An index block can contain hundreds of profile names, so this greatly speeds up the process, and, by doing so, makes fewer calls to MVS services.
• Instead of using ELSQA storage, the profile lists are now in 64-bit above-the-bar storage. This removes the impact on fixed frame storage.

• The RACF SET command has a new operand called GENERICANCHOR that enables you to configure the number of profile lists maintained by RACF for an address space. A generic anchor is a header that points to a profile list created during generic profile loading. Using the GENERICANCHOR operand, you can set a system-wide (all jobs) value from a minimum of 4 (the default if not specified) to a maximum of 99. You can also specify the number of profile lists RACF maintains for a specific job name.
• After a profile list is built, it is no longer only searched sequentially to identify the covering profile. Instead the search is both binary and sequential. So the search for a best-fit profile can be performed more quickly.

Forget those workarounds
Now that the issues of time and processing overhead in building profile lists, the issue of fixed storage, and the issue of time to find the best-fit profile from a long list have all been addressed, you can forget about those workarounds. Just forget them. In fact, see where they’re listed earlier in this article? Grab a pen and cross them out.

To read more about the RACF SET command and its new GENERICANCHOR operand, see z/OS Security Server RACF: Command Language Reference, SA22-7687.

Where's the bottleneck?
Improve performance by understanding program flow with timed event data

BY SCOTT TUTTLE AND RITA BEISEL

You’ve been asked to improve the performance of a key authorized application. You pore over the code and make some clever changes, but to your dismay, it’s still as slow as molasses in winter. Understanding performance can be a tricky business. With the z/OS Timed Event Data Service, however, you now have a new tool at your disposal to understand where those pesky bottlenecks are hiding.

Beginning with z/OS V1R12, the IEATEDS macro and the IEAVTEDS REXX™ exec can help you determine elapsed times in your authorized applications and help you understand and solve such problems as:

• Serialization contention
• Inefficient algorithms
• I/O delays
• Any other cases where elapsed times are not what you expect.

How to get started
Suppose you have a slow-running, three-module authorized application, where module A calls modules B and C, like this:

Module A pseudo-code:
Initialize variables
Call Module B
Call Module C
Cleanup
Return
To find any potential bottlenecks, perform the following steps:

**Step 1: Add IEATEDS macro invocations at key points to determine overall elapsed times of events**

To determine the elapsed times for modules B and C, add an IEATEDS Request=REGISTER invocation at the beginning of module A and IEATEDS Request=RECORD invocations around the calls to modules B and C:

**Module A pseudo-code:**

Initialize variables

If MyTedToken = zeros then /* not yet registered? */

IEATEDS Request=REGISTER, MaxEvents=F’64’,

TedToken=MyTedToken, ...

Set MyEventThread = STCK TOD value

IEATEDS Request=RECORD, EventType=START,

EventThread==MyEventThread,

EventDesc==CL32’Calling ModB’,

TedToken=MyTedToken, ...

Call Module B

IEATEDS Request=RECORD, EventType=MID,

EventThread==MyEventThread,

EventDesc==CL32’Back From ModB, calling ModC’,

TedToken=MyTedToken, ...

Call Module C

IEATEDS Request=RECORD, EventType=END,

EventThread==MyEventThread,

EventDesc==CL32’Back From ModC’,

TedToken=MyTedToken, ...

Cleanup

Return

**Things to note:**

- Elapsed time is wall clock time, not CPU time.
- The REGISTER request allocates a Timed Event Data Table in common storage above the 2 GB bar for the events you plan to record. The REGISTER should only be done once per IPL, and MyTedToken needs to be placed into an area that persists across each run of the authorized application.
- Set MyEventThread = STCK obtains a unique value to use for the EventThread. This will help later when analyzing the Timed Event Data Report in a spreadsheet program by allowing you to sort on the EventThread to isolate the events for each invocation of the authorized application.
- Each RECORD request places a time-stamped event into the Timed Event Data Table with the data you specified and additional data collected by the service.

**Step 2: Obtain the report**

First, run your authorized application a few times to cause the events to be recorded to the Timed Event Data Table. Then, run the IEAVFTED REXX exec to format the data and create the Timed Events Data Report. You can optionally load the report data into a spreadsheet program.

**Step 3: Find the bottlenecks**

Use the “Thread Prior Event Delta” field in the Timed Events Data Report to find the elapsed time between events. If a large event appears to be slow, drill down and add more IEATEDS Request=RECORD invocations for a more granular picture of elapsed time. For example, if module C is showing a long elapsed time, and it has code that obtains an ENQ and another section with a processing loop, add IEATEDS Request=RECORD requests around these events and go back to step 2.

**Summing up:**

Divide and conquer those bottlenecks!

Using this iterative approach of adding IEATEDS RECORD invocations to determine elapsed times will help you divide and conquer those bottlenecks and allow you to make informed decisions toward improving the performance of your authorized applications.

For more information about the IEATEDS macro, see *MVS Programming: Authorized Assembler Services Reference, Volume 2*, SA22-7610.
**Keep data spikes in sight**

*Flood insurance included*

BY ANTHONY SOFIA

In z/OS V1R12, System Management Facilities (SMF) can detect and inform you of SMF-recording flood conditions for both data set and log stream recording. Every time a program uses one of the SMF write macros (SMFWTM or SMFEWTM), SMF looks for a flood condition. Then, if SMF detects flooding, it performs the policy action that you define.

The advantage of this support is that it can detect and suppress floods of SMF records without requiring additional long-term storage and post-processing. In addition, you can set up your flood policy to discard SMF records that are not critical to your business during a flood condition while preserving the critical records.

It’s easy to set up an SMF flood automation policy in the SMFPRMxx parmlib member.

**Gathering statistical data for your policy**

To get the information to help you build your policy, create a flood statistics report using parameter FLDSTATS(xxxx) on the SMF dump programs (IFASMFDL or IFASMFDP). The report displays statistics for each record type matching the filters that you specify in on the SMF dump program.

Be careful when you specify new keyword FLDSTATS(xxxx)—although the xxxx value specifies an interval, it’s not the regular SMF interval. Instead the FLDSTATS(xxxx) interval is the number of SMF records processed before the dump program performs the next flood calculation for the report.

For each record type matching your dump program filters, the flood statistics report shows the number of intervals processed and the average, low, high, and standard deviation times for all the intervals. SMF creates a unique report for each z/OS image.

**Setting up a policy**

After you gathered some statistics, use the new FLOODPOL statement in parmlib member SMFPRMxx to define your flood automation policies. Each FLOOD-POL statement creates an individual policy for specified SMF record types and includes subparameters defining the flood conditions you want SMF to look for, as well as the action you want the system to take when a flood condition occurs.

You can dynamically change the policy statements using a SET SMF=xx command to specify a different SMFPRMxx parmlib member with the policy values that you desire.

**Policy actions defined in SMFPRMxx**

Each FLOODPOL policy includes an action that you want the system to take when it detects a flood condition. For example:

- **MSG** specifies that the system issues a message to signal the start and the end of the flood.
- **DROP** specifies that the system issue messages to signal the start and end of the flood, and drops or rejects any record types specified in this FLOODPOL statement that experience flood conditions. While a flood condition is active, SMF write macros SMFWTM or SMFEWTM issue return code X’34’ for each rejected record. Even though records are dropped, IEFU83, IEFU84 or IEFU85 exits still get control.

When a flood condition ends there are two different ways that you can determine how many records were dropped during the flood:

- Look for message IFA783I that is issued when the flood condition ends. IFA783I displays the time that the flood ended, the number of records dropped, and the SMF record type or types affected.
- The system issues SMF record type 7 that now contains the number of records dropped, the SMF record types affected, and the start time of the flood. The end time of the flood can be determined from the SMF record timestamp (or from message IFA783I).

**Enabling the facility**

By default, the SMF record flood support is not enabled. You can enable this support using the FLOOD(YES) parameter in SMFPRMxx. If you disable SMF record flood support after having it enabled, the system resets all FLOODPOL policies and ends any detected flooding conditions.

**More information**

For further information, see the following z/OS V1R12 publications:

- For the FLDSTATS option and updated SMF Type 7 record, see z/OS System Management Facilities, SA22-7630.
- For the FLOOD and FLOODPOL statements in parmlib member SMFPRMxx, see z/OS MVS Initialization and Tuning Reference, SA22-7592.
- For messages IFA780A, IFA781I, IFA782A, and IFA783I see z/OS MVS System Messages, Vol 8 (IEF-IGD), SA22-7638.
IP Security might sound pretty generic, but to the Communications Server it means something very specific. It’s our implementation of the IP Security Protocol Architecture (IPSec). IPSec is comprised of a set of protocols defined by the Internet Engineering Task Force (IETF).

IPSec is similar to Transport Layer Security (TLS) and its predecessor Secure Sockets Layer (SSL), only more powerful. While TLS and SSL are capable of securing Transmission Control Protocol (TCP) communications at an application level, IPSec is capable of securing any IP communications at a host-to-host level.

This article explores some of the major components and functions of IPSec and describes significant enhancements for z/OS V1R12.

**Significant enhancements for IPSec: understanding the components**

The z/OS V1R12 Communications Server IPSec enhancements are the most significant since IPSec was integrated into the Communications Server. Before diving into the details of these enhancements, let’s examine the major components used by IPSec. Understanding these components is essential to successfully deploying IPSec.

At the heart of IPSec is the IP stack. As packets flow through an IPSec-enabled stack, IP filter policy is consulted to determine what should be done with the packet. The IP filter policy contains IP filter rules. Each rule specifies one of three actions:

1. Allow the packet to flow.
2. Discard the packet.
3. Apply IPSec.

A security association (SA) identifies what to do when the stack must apply IPSec to a packet. You can create an SA manually or dynamically. A manually-created SA is fully defined within the IP filtering policy, but use is limited. A dynamically-created SA is created using an IPSec protocol known as the Internet Key Exchange (IKE).

**IKED**

The Internet Key Exchange Daemon (IKED) creates dynamic SAs on behalf of a stack by negotiating with a remote security endpoint. IKED consults key exchange policy and IP filter policy during a negotiation. A key exchange policy contains a set of key exchange rules that are analogous to the IP filter rules of the stack. These rules identify the security endpoints of an Internet Key Exchange (IKE) negotiation.

**IKED, IKE, and certificates**

IKE supports the authentication of security endpoints using certificates. IKED can be configured on a per-stack
basis to use its own certificate services or the services of a Network Security Services Daemon (NSSD). When IKED is configured to use NSSD certificate services on behalf of a stack, IKED acts as a network security client. IKED also provides an IPSec network management interface (NMI) through which network management applications can manage IP filtering and IPSec on local TCP/IP stacks.

**NSSD and IPSec**

NSSD provides a centralized certificate repository and the following services to multiple IPSec-enabled stacks:

- Certificate storage on a SAF key ring
- Use of SERVAUTH profiles to control the access of network security clients to certificates
- Digital signature generation and verification for network security clients
- A network management interface (NMI) through which network management applications can manage IP filtering and IPSec on remote NSS IPSec clients, or monitor NSS clients that are connected to the local NSS server.

The Integrated Cryptographic Service Facility (ICSF) is used by the stack, IKED, and NSSD to perform cryptographic operations. IKED also uses the SSL for certificate operations. Both ICSF and System SSL use the CP Assist for Cryptographic Function (CPACF) and Crypto Express2 or Crypto Express3 cards as appropriate.

**Policy agent**

The policy agent manages IPSec policy files. It monitors the files for updates, parses them, and provides policy to the stack and IKED.

**IBM Configuration Assistant**

The IBM Configuration Assistant for z/OS Communications Server is an optional graphical user interface that you can use to generate policy files. It simplifies configuration by hiding the complexity of individual configuration files. The Configuration Assistant runs under the IBM z/OS Management Facility (z/OSMF) or standalone.

The IBM Configuration Assistant for z/OS Communications Server provides you with a policy editor where you can use to generate policy files. It also provides an optional graphical user interface that simplifies configuration by hiding the complexity of individual configuration files.

**z/OS Communications Server**

z/OS Communications Server is an integrated stack and IKED.

**The Configuration Assistant**

The Configuration Assistant parses them, and provides policy to the policy agent. The policy agent manages IPSec policy files. It monitors the files for updates, parses them, and provides policy to the stack and IKED.

The z/OS V1R12 Communications Server IPSec enhancements are the most significant since IPSec was integrated into the Communications Server. When downloaded from the z/OS Communications Server product support webpage:


**TRMD**

The traffic regulation manager daemon (TRMD) is used by the stack to write messages to the syslog daemon (syslogd).

**z/OS coupling facility**

The z/OS coupling facility is used by the stack to store information about security associations for dynamic virtual IP addresses (DVIPAs) in a sysplex. This information facilitates both the distribution and takeover of IPSec SAs associated with DVIPAs.

**Commands of interest**

You can use the `ipsec` command to perform IPSec-related management tasks. Use the `nssct1` command to display information about network security clients that are connected to NSSD.

**IPSec for z/OS V1R12: Understanding the enhancements**

In z/OS V1R12, many of these components were enhanced to support the following features:

- New cryptographic algorithms
- New authentication algorithms
- New Diffie-Hellman Group algorithms
- IKE signature algorithms

Table 1 summarizes the new cryptographic algorithms supported in z/OS V1R12:

<table>
<thead>
<tr>
<th>New encryption algorithms</th>
<th>New authentication algorithms</th>
<th>New Diffie-Hellman Group algorithms</th>
<th>IKE signature algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES_CBC (256 bit)</td>
<td>AES_GMAC_128</td>
<td>256-bit random ECP group</td>
<td>ECDSA-256</td>
</tr>
<tr>
<td>AES_GCM_16 (128 bit)</td>
<td>AES_GMAC_256</td>
<td>384-bit random ECP group</td>
<td>ECDSA-384</td>
</tr>
<tr>
<td>AES_GCM_16 (256 bit)</td>
<td>AES128_XCBC_96</td>
<td>521-bit random ECP group</td>
<td>ECDSA-521</td>
</tr>
<tr>
<td></td>
<td>HMAC_SHA2_256_128</td>
<td>2048-bit MODP Group with</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HMAC_SHA2_384_192</td>
<td>256-bit prime order subgroup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HMAC_SHA2_512_256</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Summary of z/OS V1R12 cryptographic algorithms

It will probably take time for the industry to fully move to IKEv2. IKEv1 continues to be around for the foreseeable future. You can use both protocols simultaneously in IKED. If a peer initiates SA negotiations using IKEv2, IKED automatically responds by using IKEv2 without any additional configuration. If you want to have IKED initiate SA negotiations using IKEv2, you must update your configuration.
IKED, IKE, and the new cryptographic algorithm support

IKED supports a subset of some of the new encryption and authentication algorithms summarized in the table to protect IKE traffic.

The new encryption and authentication algorithms are available to the stack when you apply IPSec to a packet. Also, a subset of the new encryption and authentication algorithms are available with IKED when you want to protect an SA negotiation. Although encryption and authentication algorithms with longer lengths are thought to be more secure, they are also more CPU-intensive.

IKED uses the new Diffie-Hellman (DH) groups when generating cryptographic keys for SAs. Don’t be alarmed by the small bits size of the Elliptic Curve Groups (ECP). For a given number of bits, ECP is generally considered to be more secure than MODP groups.

IKED uses the new Elliptic Curve Digital Signature Algorithms (ECDSA) for peer authentication. They are only available when you use IKEv2. ECDSA authentication has security and performance advantages compared to RSA authentication.

To generate cryptographic keys, IKE uses the DH key exchange. DH “groups” control how secure the DH key exchange is.

IKED now supports three Elliptic Curve Groups:
- 256-bit random ECP group
- 384-bit random ECP group
- 521-bit random ECP group.

IKED also supports one new Modular Exponential (MODP) group: 2048-bit MODP Group with 256-bit Prime Order Subgroup.

FIPS-140

Customers in government agencies and some private sector industries might need to comply with Federal Information Processing Standards such as FIPS-140. To address this need, the Communications Server now supports a FIPS-140 operational mode. In this mode, all cryptographic operations are supplied by cryptographic modules (ICSF PKCS#11 interface and system SSL) that are designed to address FIPS 140 requirements. Weaker cryptographic algorithms are prohibited and minimum key lengths are enforced. Direct hardware instructions and some optimized interfaces also are not allowed. As a result, the FIPS-140 mode might not perform as well as the non-FIPS-140 mode, so you should only enable it if you have a compelling business need.

Certificate trust chains

The Network Security Services daemon (NSSD) now supports certificate trust chains for creating and verifying digital signatures. When IKED is configured as a network security client, it can use the certificate trust chain support in NSSD to maximize interoperability with non-z/OS IKE peers with no additional configuration.

When IKED verifies a digital signature, it should ensure that the signing certificate is still valid by consulting a certificate revocation list (CRL). IKED can request that NSSD retrieve and consult CRLs when it verifies a digital signature. A certificate usually contains the information necessary to retrieve the CRL that governs it.

IKEv2 and ECP DH and ECDSA algorithms

If you are using an ECP DH group with IKEv2, it is a good idea to pair it with an ECDSA algorithm that uses the same number of bits so that the IKE peer authentication and key generation provide similar strengths. For example, if you are using the 256-bit random ECP group for DH, consider using ECDSA-256 for authentication.

Acknowledgments

We hope this information helps accelerate your use of the features added in z/OS V1R12. We’d like to thank Chris Meyer, Scott Moonen, and Allen Bailey for their technical and editorial reviews of our article.
If SVC dump were a truck, you could say that there has been a redesign of what’s under the hood and that the dashboard also now sports some pretty spiffy new knobs! SVC Dump went into the z/OS V1R12 development shop for an overhaul, and not only got its engine rebuilt, but also got some smart new external control features that will delight you system programmers and operation managers out there! These improvements work together to make SVC dump processing smarter and faster, while making z/OS and the sysplex more resilient during dump processing.

Enhanced external control knobs for the sysprog

All you experienced z/OS SVC dump takers out there are familiar with the CHNGDUMP SET, SDUMP,… command, and how it allows you to configure SVC dump processing smarter and faster, while making z/OS and the sysplex more resilient during dump processing.

Existing CHNGDUMP keyword MAXSPACE is another way to restrict DUMPSRV’s use of virtual storage, and the system will always honor the MAXSPACE specification when it is more restrictive than AUXMGMT.

With AUXMGMT=OFF, SDUMP virtual storage management reverts to previous z/OS behavior. Controlled use of auxiliary storage is available only via MAXSPACE and SRM monitoring for critical auxiliary storage shortage.

AUXMGMT=ON is the default setting, which makes availability a higher priority than first-failure data capture; however, the system takes a complete dump as long as the installation has sufficient auxiliary storage.

A system taking an SVC dump can get partitioned out of its sysplex environment if a slowly progressing global data capture leaves the system non-dispatchable for over 15 seconds. To avoid this problem, you can now specify new keyword MAXSNDSP on the CHNGDUMP command to modify the system non-dispatchable time criteria from the default value of 15 seconds.

So what’s new under the hood?

We made improvements to SVC dump’s internal algorithms to reduce memory pressures when capturing exit data. The z/OS real storage manager (RSM) now invokes a “Smart Copy” algorithm such that, if the source data is on auxiliary storage, z/OS performs an I/O operation directly into the SDUMP buffer, (rather than allowing it to be paged into real as any normal reference would cause it to be, and then subsequently copying it into the SDUMP capture data space).

No new dumps are allowed to start when auxiliary storage utilization is at 50% or higher.

Dump data capture is halted when auxiliary storage utilization reaches 68%.

After the 68% threshold is reached, new dumps are not allowed to start until the auxiliary storage utilization drops below 35%.
Furthermore, z/OS now uses a special RSM service such that if the source data did not appear referenced before the capture, it remains unreferenced after the capture. Also, the data in the SDUMP capture space is made “to look old” so that it is a top candidate for being paged out. In these ways, captured data (which is not likely to be referenced again) will not cause other, more important data, to be paged out.

GRS tweaks for GRSQ
Specifying GRSQ on an SVC Dump has always provided valuable information for serialization-related problems. The scalability of today’s systems can translate to an enormous number of ENQ-related control blocks being gathered for this GRSQ processing. In addition to taking advantage of the smarter dump algorithms, GRS now compresses certain storage buffers to mitigate the amount of paging encountered at dump capture time. These buffers were originally created to improve mainline ENQ performance. However they could ironically slow down dump processing due to an inordinate amount of paging. The system now tunes itself based on internal thresholds, building up the buffers with spikes of ENQ activity and compressing them as necessary. This compression was rolled down to z/OS R9 in a PTF for APAR OA29329. In z/OS R12, GRS has additional support to disable these buffers in cases of real or auxiliary storage shortages.

Find out more
You can read about the new keywords on CHNGDUMP in the z/OS V1R12 MVS Commands Reference, SA22-7627-24.

For more information on tailoring GRSQ-related information in SVC dumps, see the GRSCNFxx parmlib description in z/OS V1R12 MVS Initialization and Tuning Reference, SA22-7592-21.

These changes have resulted in dramatic performance improvements, particularly for the capture time phase for address spaces with a high percentage of data on auxiliary storage, and especially in environments with no real storage constraints. Some examples:

- For a 40GB system with 60% of its data on auxiliary storage, a dump that took 55 minutes on z/OS V1R11 now takes only 4 minutes on z/OS V1R12
- For a 40GB system with only 5% of its data on auxiliary storage, there is still a 50-80% improvement in overall dump processing time, in both constrained and non-constrained environments.
Mirror, mirror on the wall, should couple data sets be mirrored at all?

BY NICOLE FAGEN AND DONNY CHAN

Should disaster-recovery-site couple data sets be mirrored or copied from the primary site couple data sets (CDSs)? The answer is—it depends. Mirroring is a hardware solution that duplicates data from one disk to another disk. Mirroring can be done synchronously or asynchronously. Copying, on the other hand, refers to the process of making a point-in-time copy of a data set to disk or tape backup. To avoid high-impact failure scenarios, including sysplex outages, careful planning is required if CDSs are mirrored or copied.

Mirroring or copying CDSs
CDSs are mirrored or copied to simplify the configuration and help reduce recovery time at the disaster recovery (DR) site. You can simplify the configuration by allowing all volumes on a device to be mirrored so you do not have to manually exclude a selection of volumes on which CDSs reside. Another way to simplify the configuration is to ensure the record or policy-definition data in the CDSs noted in Table 1 is available at the DR site.

For example, mirroring or copying the coupling facility resource management (CFRM) CDS ensures the structure definitions and coupling facility (CF) definitions are available at the DR site. Another example of simplification is when mirroring sysplex failure management (SFM) allows the system weights and other SFM settings to be available at the DR site. See Table 1.

<table>
<thead>
<tr>
<th>CDS type</th>
<th>Data contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSPLEX</td>
<td>• Record for each system in the sysplex</td>
</tr>
<tr>
<td></td>
<td>• Names of the functional CDSs last used</td>
</tr>
<tr>
<td></td>
<td>• Cross-system coupling facility (XCF) groups and group members</td>
</tr>
<tr>
<td></td>
<td>• Write to operator with reply (WTOR) ID bit map.</td>
</tr>
<tr>
<td>CFRM</td>
<td>• CF definitions, node descriptor information, and CF name</td>
</tr>
<tr>
<td></td>
<td>• Structure definitions</td>
</tr>
<tr>
<td></td>
<td>• Status of structure allocations and connections.</td>
</tr>
<tr>
<td>SFM</td>
<td>• Names and weights of systems</td>
</tr>
<tr>
<td></td>
<td>• Actions to take if there is a failure.</td>
</tr>
<tr>
<td>Automatic restart manager (ARM)</td>
<td>• Where applications should restart</td>
</tr>
<tr>
<td>Workload Manager (WLM)</td>
<td>• Service class definitions</td>
</tr>
<tr>
<td>BPXMCDS</td>
<td>• Mounted file systems</td>
</tr>
<tr>
<td>System logger (LOGR)</td>
<td>• CF structure definitions defined to LOGR</td>
</tr>
<tr>
<td></td>
<td>• Log stream definitions</td>
</tr>
<tr>
<td></td>
<td>• Log stream state information</td>
</tr>
<tr>
<td></td>
<td>• DSEntext records.</td>
</tr>
</tbody>
</table>

Table 1. Summary of z/OS V1R12 cryptographic algorithms
Synchronous mirroring and the sysplex CDS
Avoid synchronous mirroring of sysplex couple data sets because sysplex performance is impeded when you access CDSs that are mirrored synchronously. XCF maintains the primary and alternate CDSs. The write to the CDS is not complete until the local copy and the remote copy of the CDS are successfully written. The additional overhead of synchronous mirroring can cause delays in applications that need to access the CDS. If a delay occurs or the there are long busy conditions on the mirrored copy, I/O delays are reported for attempts to access CDSs. In some cases, the mirroring technology might be suspended. In other situations, if the delays are long enough, a permanent I/O error condition against the CDS can occur. XCF takes the CDS that encountered the error out of service. If both the primary CDS and the alternate CDS encounter a permanent I/O error, the functions provided by the CDS will cease immediately. If it is the SYSPLEX CDS or the CF defined in the active CFRM policy. Global resource serialization is not able to allocate the ISGLOCK structure and the system will terminate with a X’0A3’ wait state. Other unpredictable results might also occur.

How to avoid the problem: Do not allow DR site systems to access the primary site DASD directly.

Asynchronous mirroring or tape-based replication
It is possible to asynchronously mirror CDSs, but careful planning and setup is required. To ensure the resiliency of the primary sysplex CDSs, carefully consider each of the following scenarios that will affect mirroring.

Crossing CDSs
Situation: The CDSs are asynchronously mirrored or copied to the DR site. Both the primary site CDSs and the DR site CDSs reside on devices with like names. Initially, the DR site systems do not have access to the primary site DASD.

Pitfall: Days, weeks, and months pass and a subset of the DR site systems are configured to have full access to the DASD at the primary site. The systems at the DR site are IPLed and some systems access the CDSs on the primary site DASD while other systems access CDSs on the DR site DASD. The result of this mixed-CDS access includes the loss of the CDS from the primary site if the primary site is active, sysplex outages, data integrity problems, and other unpredictable errors.

How to avoid the problem: Do not allow DR site systems to access the primary site DASD directly.

Hide and seek
Situation: The SYSPLEX CDS and CFRM CDS are mirrored from the primary site to the DR site. The only CFs defined in the CFRM policy reside at the primary site. Further, the defined CFs are only accessible at the primary site.

Pitfall: When the first system is IPLed at the DR site, the system discovers that there is no physically connected CF defined in the active CFRM policy. Global resource serialization is not able to allocate the ISGLOCK structure and the system will terminate with a X’0A3’ wait state.

How to avoid the problem: If CDSs are to be mirrored, the CFs at the DR site must be defined in the active CFRM policy.

Ghosts and vanishing CFs
Situation: The CFRM CDSs and SYSPLEX CDSs are asynchronously mirrored or copied from the primary site to the DR site. CFs for the primary site and CFs for the DR site are declared in the same CFRM policy. The DR site does not have connectivity to the primary site CFs.

Pitfall 1: If the systems are started at the DR site using the mirrored or copied CFRM CDSs, there can be residual data. XCF will clean up all the information in the active policy for the structures that are not available at the DR Site. It is unlikely that XCF will be able to clear up all of this stale data. As a result, the displays of structures might contain “ghost” entries for the instances of the structures that resided in the CFs at the primary site. The “ghost” is the instance of the structure that is noted in the CFRM CDS but cannot be accessed at the DR site because the primary site CFs are not available.

Pitfall 2: Time passes and configuration changes to the sysplex are made, including CFs from the primary site being connected to the DR site systems. A system at the DR site is IPLed, and it uses the CFs from the primary sysplex without any warning! If all CFs are taken from the primary sysplex and there is an ISGLOCK structure, all systems are terminated with a X’0A3’ wait state. Other unpredictable results might also occur.

How to avoid the problem: Never allow a DR site (or second sysplex) using a mirrored CFRM CDS to obtain access to primary site CFs.

DR site: Unique CDSs
It is true that some aspects of the policy definitions in the mirrored or copied CDSs can be used at the DR site. The CDSs also contain “live” state information that is only useful in the sysplex in which it was created. When the data from the primary site sysplex is detected, the DR site systems will update the CDS with meaningful data and delete obsolete data (when possible). The sysplex CDS data is overwritten with new data from the DR site systems and DR site XCF groups. The BPXMDCDS couple data set is updated to reflect the DR site mounted files.

As you can see, there are potential pitfalls associated with mirroring or copying CDSs. At a minimum, exercise caution when you mirror CDSs for certain configurations at the DR site. You can also consider the alternative to mirroring, creating and initializing unique DR site CDSs that mimic the primary site CDSs. Then, whenever updates are made to primary site CDSs, ensure that you make the same updates to
the DR site CDSs. Finally, you must use a unique COUPLEExx parmlib member at the DR site naming the CDSs formatted specifically for the DR site.

**LOGR CDS: The exception**

To every rule, there is an exception, in this case, the LOGR CDS. If the system logger log-stream data is being mirrored from the primary site to the DR site, and the data in the log stream is to be used at the DR site, the LOGR CDS must be mirrored. The LOGR CDS contains valuable information that the system logger uses to determine from what location it retrieves connector data and into which offload data set it offloads data. Without the LOGR CDS, an application cannot connect and access data previously written to a log stream.

Mirroring system logger data and the LOGR CDS means making sure all of the data components in the system logger environment are time consistent. Synchronously, the mirroring of the system-logger offload data sets, staging data sets, LOGR CDS, and MVS catalogs ensures that the DR site has all of the data necessary to be used immediately if the DR site is activated.

**Final reflection**

CDSs are being mirrored or copied to reduce recovery by simplifying the DR site initialization process. Mirroring CDSs avoids the need to perform policy updates manually at the DR site every time an update against the primary site CDS occurs. On the other hand, there are potential risks if the installation does not take precautions to avoid the pitfalls when mirroring or copying CDSs. Plan wisely and always consider the CDSs when making changes to the primary site or DR site.
Don Ault is a Senior Technical Staff Member for z/OS in Poughkeepsie, New York. In his 36 years with IBM, he has worked on z/OS components SDUMP, IPCS, GRS, XCF, Unix System Services, RSM, 64-Bit Virtual Resource Access Control Facility (RACF) and most recently ICSF restructure and performance enhancements. Currently he works on ICSF design, development, and service.

Arthur Bariska is Senior Technical Staff Member (STSM) working in System Verification Test for DFSMS for over half his 36 year career. As an ASTQB Certified Test Professional, Art has published a number of articles on testing.

Rita Beisel is a technical writer at IBM. She is currently writing for z/OS MVS and has written for Resource Access Control Facility (RACF), z/VM, DB2, and DFSMS in the past. She also likes to do interface design, usability testing, and surveys.

Laura Bostian is a senior software engineer in Poughkeepsie, New York, working on simplification of the user experience on System z. She has been with IBM for 22 years working in z/OS development and most recently in various customer engagement programs.

Stephen M. Branch is an IBM Senior Software Engineer at Silicon Valley Lab whose 25 year career has dealt with all aspects of DFSMS. He is the author of the Catalog Search Interface (CSI). Stephen holds a Master of Science Degree from Louisiana State University.

Wendy Brinkman is a WebSphere Application Server System z tester. She has performed focused testing for the WOLA enhancements in WebSphere Application Server 7.0.0.12 and has worked with WebSphere Application Server on z/OS for 14 years. Wendy also worked in development on zHardware and has tested 3033, 3081, 3090 systems before moving to software development. Wendy joined IBM in 1981.

Eugenia Brooks is an Advisory Programmer working in System Verification Test for DFSMS for over 4 years. She was level 2 support for VSAM, RLS, Catalog and AMS for the first 21 years of her career.

John Burg is an Executive IT Specialist in the IBM Advanced Technical Skills organization. With over 32 years of IT experience, his focus is on helping customers with z/OS performance and System z capacity sizing.

Bill Carey is a Senior Software Engineer in the System z Strategy and Technology organization. His areas of responsibility have included XML technology strategy and design for the last ten years.

Donny Chan is the system logger team lead at the Poughkeepsie Support Center in Poughkeepsie, New York. He graduated from the University of Illinois and has been with IBM for over ten years.

Eleanor Chan has been a function tester in the Integrated Cryptographic Service Facility (ICSF) for over 11 years before moving to Parallel Sysplex/Coupling Facility for four years. She has recently returned to ICSF where she is a function test team leader.

Gayathiri Chandran is the security team lead for DB2 for z/OS at IBM Silicon Valley Laboratory in San Jose, California. She has led many security related projects such as data stream encryption support, network trusted contexts and roles, and new DB2 authorities in DB2 for z/OS.

Gisela Cheng is an Advisory Software Engineer for z/OS at IBM in Poughkeepsie, New York. She has worked in various OS system development teams, as well as application development and software integration area. Gisela holds a Master of Science degree from Union College.

Bryan Childs is a Senior Software Engineer at IBM in Poughkeepsie, New York where he leads the Global Resource Serialization (GRS) Development Team. Bryan holds an M.S. degree in Software Development from Marist College where he now teaches various online certificate classes on z/OS.

Matthew Cousens is the test team lead for z/OS XML System Services and the XML Toolkit for z/OS. He has worked on testing XML on z/OS since 2007 first concentrating in functional test and recently expanding his role to include stress testing.

Yufen Davis is a Senior Software Engineer for DFSMS/Hshm Development in Tucson. She has been with IBM for 22 years with experience in DFMSHshm and DFMS/Hshm. Her focus areas include copy services and fast replication backup and recovery.

Greg Daynes is a Senior Technical Staff Member and the z/OS Installation Deployment Architect in IBM at Poughkeepsie, New York. His current responsibilities include z/OS software acquisition, installation, migration, deployment, and maintenance strategies. Greg is a frequent speaker at SHARE and the IBM System z Technical University (formerly zExpo).

Adrian Alvarez Diez is a Software Engineer at IBM Lab in Boeblingen, Germany. He holds a degree in Computer Science from the University of Stuttgart, Germany. Since joining IBM, Adrian has worked for various z/OS projects, such as WLM.

Jodi Everdon is an Advisory Software Engineer at IBM Poughkeepsie. She is a managing editor for z/OS Hot Topics, the Information Planner for z/OS MVS, and the author of z/OS Problem Management.

Kathleen Fadden is the Integrated Cryptographic Service Facility (ICSF) Level 3 Team Leader. She has been working in ICSF for over 15 years. Previously, she worked on the Resource Access Control Facility (RACF) service team and in various VM test groups.

Nicole M. Fagen is a Senior Software Engineer. She works in the Poughkeepsie, New York Support Center and is the team lead for Parallel Sysplex support. Nicole’s experience includes over ten years debugging complex problems related to Parallel Sysplex.

Marisa Freidhof is a Software Engineer for the I/O Supervisor component of z/OS. Her expertise includes IOS Level-3 Service, specializing in many areas of the IOS component. Marisa holds a B.S. degree in Computer Science from Binghamton University.

Tammy Garren is a Staff Software Engineer in z/OS Global Resource Serialization (GRS) development and the GRS test team lead. She holds a B.S. degree in Biology from Russell Sage College and has been with IBM for 28 years.

* Certified Information Systems Security Professional
** Certified Secure Software Lifecycle Professional
Joseph Gentile has worked at IBM since 2008, contributing to z/OS development as a Global Resource Serialization (GRS) function tester. He holds a B.S. degree in Computer Science from Binghamton University.

Kevin Goldsmith is a software engineer at IBM in Tucson, Arizona. In his 29 years with IBM, Kevin has worked in MVS development, Tivoli®, storage hardware and is currently a technical lead in the Object Access Method (OAM) component of DFSMS.

Michael Groetzner is an Advisory Specialist for z/OS Programming at IBM Germany. His expertise includes z/OS Capacity Provisioning, Michael holds a degree in Computer Science from the University of Cooperative Education in Stuttgart, Germany.

Jeff Haggar works on the design and development of z/OS Communications Server in Research Triangle Park, North Carolina and has expertise with networking hardware and software on the System z platform. Jeff holds a B.S. degree from Rensselaer Polytechnic Institute. He is an IBM Master Inventor and has several patents in I/O and networking technologies.

Russell Hardgrove has worked in the IT industry for 37 years and for IBM in Resource Access Control Facility (RACF) Level-2 Support since 1999. He is a frequent contributor to the RACF-L discussion list, is a frequent presenter at various RACF user's groups and has presented at the Vanguard Enterprise Security Expo and the IBM SecureWorld® conference.

Steven Hart, CISSP®, is a Staff Software Engineer for z/OS Cryptographic Services in Poughkeepsie, New York. In his seven years with IBM, he has worked on Integrated Cryptographic Services Facility (ICSF), Trusted Key Entry (TKE), Encryption Facility for z/OS, and Tivoli Key Lifecycle Manager (TKLM) for z/OS. Currently he works on ICSF design, development, and service.

Hugh Hockett is an Advisory Software Engineer and an IBM Master Inventor in Research Triangle Park, North Carolina. He works on the design and development of networking software for z/OS Communications Server and has a B.S. degree from Virginia Tech.

Steven Jones works in the Allocation and Scheduler components of z/OS. He's done a little of everything over his 23 years at IBM Poughkeepsie: function test, development, design, and even a couple of shifts of System Test. Contact him via email at sbj@us.ibm.com.

Ken Kerr is an Advisory Software Engineer at the IBM Poughkeepsie, New York Development Lab. He has been on the Integrated Cryptographic Service Facility (ICSF) development team for over 15 years.

Larry Law is an IBM Senior Software developer with expertise in VSAM and VSAM RLS. Larry has worked for IBM for 29 years and was the lead developer for the CA Reclaim enhancement discussed in this issue’s article “What can VSAM CA reclaim do for you?” on page 44.

Kristine Logan has been working on z/OS documentation for an unbelievable 24 years. Working on IBM Health Checker for z/OS has been one of the most interesting projects of her career, with its emphasis on customer feedback and teamwork.

Randy Love is a Resource Access Control Facility (RACF) function test team leader since 1992 and has worked on z/OS RACF releases including RACF remote sharing function, multi level security and support for all DB2 releases since DB2 V5.

Rich Lungerhausen is an Advisory Software Engineer for DFSMS Catalog Development in San Jose, California. He recently worked on implementation of Catalogs in Extended Addressable Storage (EAS) for z/OS release 12 and has been a member of the Catalog Development team for the last 5 years. Rich holds an MBA from the University of Colorado.

David Manners is a Senior Software Engineer and has worked at IBM for 27 years. His z/OS component experience includes RSM, XCF/XES, consoles, allocation, SDUMP, Global resource serialization (GRS), and most recently on ICSF design, development and service. His IBM career has also included stints in management, the z/OS porting center, and as a designer for the Enterprise Workload Manager (EWLM).

Charles Mari is a software engineer in the Systems and Technology Group at IBM Poughkeepsie. He currently works in the Storage Management area of z/OS Core Technologies.

George Markouzos, CISSP®, has been with Resource Access Control Facility (RACF) development for about 20 years and has participated in many development projects including the team lead for the support for z/OS UNIX File System access control lists, multilevel security, installation-defined RACF checks for IBM Health Checker for z/OS, ID propagation, and generic profile load performance.

Kirsten Brunner McDonald is a Senior Software Engineer working on mainframe simplification. She has been with IBM for 22 years working on z/OS workload management, WebSphere Application Server, System z Capacity Provisioning, z/OS Management Facility, and the z Hardware Management Console.

Kershaw Mehta is a Senior Technical Staff Member in the System z Strategy and Design Group. He has development experience in z/OS UNIX, Language Environment®, and Unicode Services and is now responsible for their future direction and strategy.

Jim Mulvey is a WebSphere Application Server developer and the architect for the Optimized Local Adapters feature and has worked for WebSphere Application Server for several years. Before that Jim worked as a developer on the z/OS Language Environment and Problem Determination Tools teams. Before coming to IBM in 1997, Jim worked as an independent consultant for several years, specializing in MVS and CICS systems programming.

Mark Nelson, CISSP®, CSSLP**, has worked in Resource Access Control Facility (RACF) Design and Development for 24 years. His accomplishments include the RACF database and SMF unload utilities, the RACF/DB2 external security module, and RACF support for IBM Health Checker for z/OS. An award-winning speaker, Mark co-authored the book, Mainframe Security for the Security Expert: An Introduction to RACF.

* Certified Information Systems Security Professional
** Certified Secure Software Lifecycle Professional
George Ng is an IT Specialist for IBM Systems and Technology Group Lab Services at IBM in Poughkeepsie, New York. His areas of focus include z/OS Parallel Sysplex, high availability, and performance.

Tan Q. Nguyen is an IBM Staff Software Engineer at Silicon Valley Lab whose more than 4 year career has dealt with all aspects of DFSMS and DASD Common Services. He specializes in ICF Catalog. Tan has a Bachelor of Computer Science Degree from San Jose State University.

Jun Ogata is a Staff Software Engineer who has been function testing RACF for more than 15 years. He has also tested other z/OS components, most recently XML System Services for the last two releases. Jun strives to demystify the magic of RACF.

Steven Partlow is an Advisory Software Engineer in the z/OS Storage Management Development Team. Steven holds a B.S. degree in Computer Science from the University of Illinois at Urbana-Champaign.

Eric Rosenfeld, CISSP®, joined IBM as a software engineer in 1985 and has been developing security products for 18 years. He is currently a developer for Resource Access Control Facility (RACF) and is the development product owner for z/OS Network Authentication Service.

Tom Ross started at IBM as a contractor in 1999 and spent 5 years working on the z/OS Fault Analyzer project where he picked up a strong understanding of z/OS and its internals. For the last 6 years, he has been developing and supporting IBM’s Network Authentication Service (z/OS Kerberos).

Matthew Sabins has been a developer for IOS for eight years, has degrees in both mathematics and computer science from Clarkson University. In his free time he does triathlons, plays football, and writes jokes.

Hiren Shah is a senior software engineer working on the zEnterprise Unified Resource Manager. Past projects have included the Enterprise Workload Management Product and the z/OS Workload Manager.

Anthony Sofia is a software engineer at IBM. Since joining IBM in 2004, he has worked in the z/OS I/O supervisor and z/OS system management facility (SMF) components.

Bob St. John is a Senior Technical Staff Member in System z Performance and Performance Tools from the IBM lab in Poughkeepsie, New York. He has over 28 years of experience with MVS and z/OS, improving the performance of various IBM software products and the tools used to analyze and improve performance.

Gayle Steinbugler is an advisory software engineer in Research Triangle Park, North Carolina. During 28 years with IBM, Gayle has provided user experience design for numerous products including: Retail Store Solutions, BladeCenter, Integrated Solutions Console, IBM Systems Director, i-, p- and zHMCs, and z/OSMF.

Theresa Tai is a Senior Software Engineer, Senior Certified IT Specialist at IBM STG, System z New Technology Center in Poughkeepsie, New York. Theresa's focuses are on workloads and IT infrastructure for the System z Enterprise and zEnterprise BladeCenter Extension (zBX), including the WebSphere Application Server, Portal Server, IBM Specialty Engine, Java, and Information Management software workloads.

Ralf Thelen is the Workload Manager (WLM) test team lead and the z/OS Quality Focal Point at IBM Boeblingen, Germany. He works on the design and execution of algorithm verification tests for z/OS Capacity Provisioning and WLM/SMR. Ralf has been with IBM for 21 years and holds a degree in computer science from The University of Bonn, Germany.

Scott Tuttle has been with IBM for 33 years and is currently a software developer at the IBM Poughkeepsie Lab in Poughkeepsie, New York. He has worked in many of the z/OS components, including Global Resource Serialization (GRS), XCF/XES, consoles, allocation, supervisor, and RSM. He enjoys the challenge of creating and using new techniques and tools that improve quality and help make coding and testing easier.

* Certified Information Systems Security Professional
** Certified Secure Software Lifecycle Professional

Anthony Sofia

Bob St. John

Gayle Steinbugler

Theresa Tai

Ralf Thelen

Scott Tuttle
Clarifications, corrections, and announcements

Visit the Hot Topics website for clarifications and corrections to articles from our past issues:

ibm.com/systems/z/os/zos/bkserv/hot_topics.html

The information in this Newsletter is presented as is and is based on the best knowledge of the authors. No warranty is provided (neither expressed nor implied). IBM may not offer the products, features or services discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM representative for information on the products or services available in your area.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.
Upcoming events:
Step up to the IBM System z Tech University

IBM System z Technical Symposium - April 27-29, 2011
(Orlando, FL)
New in 2011, this 3-day event will broaden your knowledge and expertise on System z technology from both hardware and software perspectives. This event is a shorter version of the IBM System z Technical University offered in October.

IBM System z Technical University - October 3-7, 2011
(Fontainebleau Hotel, Miami, FL)
This 4.5-day event will have more than 300 sessions that extensively cover a broad range of zEnterprise, z/OS, z/VM, zVSE and Linux on System z topics, multiple training levels (beginner to advanced), best practices and a solution center/expo. In 2011, we will significantly expand our coverage of System z software, including DB2, CICS, MQ, WebSphere, IMS, Cognos and more.

For details and a full list of IBM Systems technical events world-wide, visit the following website:
ibm.com/systems/services/conferenceseries