Your Changing z/OS Performance Management World: New Workloads, New Skills

Agenda

- The new world of RMF monitoring
  - RMF XP and z/OSMF

- z Systems hardware functions
  - SMF logstreams and zEDC

- z/OS in the new world of cloud, mobile and analytics
  - z/OS Connect
  - IDAA and WLM
RMF Product Overview and RMF XP

RMF XP Enhancements

► RMF XP is the solution for Cross Platform Performance Monitoring
► RMF XP supports the Operating Systems running on
  ► x Blades
  ► p Blades

► In addition RMF XP supports Linux on System z
  ► LPAR Mode
  ► VM Guest Mode
RMF XP Windows Support – z/OSMF Resource Monitoring

Health Check for all Windows Systems in the Enterprise:
- Processor
- Memory
- Filesystem
- Network
RMF XP can be configured to write SMF records at interval end.

### RMF XP & SMF Records

#### RMF XP

- RMF Generic CIM Client
- RMF Distributed Data Server
- SMFWTM
- GPM4CIM

#### SMF Buffer of RMF Sysplex Data Server

SMF Type 104

---

### RMF XP & SMF Records

<table>
<thead>
<tr>
<th>AIX on System p</th>
<th>ST</th>
<th>Linux on System x</th>
<th>ST</th>
<th>Linux on System z</th>
<th>ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX_ActiveMemoryExpansion</td>
<td>1</td>
<td>Linux_IPProtocolEndpoint</td>
<td>20</td>
<td>Linux_IPProtocolEndpoint</td>
<td>40</td>
</tr>
<tr>
<td>AIX_Processor</td>
<td>2</td>
<td>Linux_LocalFileSystem</td>
<td>21</td>
<td>Linux_LocalFileSystem</td>
<td>41</td>
</tr>
<tr>
<td>AIX_ComputerSystem</td>
<td>3</td>
<td>Linux_NetworkPort</td>
<td>22</td>
<td>Linux_NetworkPort</td>
<td>42</td>
</tr>
<tr>
<td>AIX_Disk</td>
<td>4</td>
<td>Linux_OperatingSystem</td>
<td>23</td>
<td>Linux_OperatingSystem</td>
<td>43</td>
</tr>
<tr>
<td>AIX_NetworkPort</td>
<td>5</td>
<td>Linux_Processor</td>
<td>24</td>
<td>Linux_Processor</td>
<td>44</td>
</tr>
<tr>
<td>AIX_FileSystem</td>
<td>6</td>
<td>Linux_UnixProcess</td>
<td>25</td>
<td>Linux_UnixProcess</td>
<td>45</td>
</tr>
<tr>
<td>AIX_Memory</td>
<td>7</td>
<td>Linux_Storage</td>
<td>26</td>
<td>Linux_Storage</td>
<td>46</td>
</tr>
<tr>
<td>AIX_OperatingSystem</td>
<td>8</td>
<td>Linux_KVM</td>
<td>30</td>
<td>Linux_zCEC</td>
<td>50</td>
</tr>
<tr>
<td>AIX_Process</td>
<td>9</td>
<td>Linux_Xen</td>
<td>31</td>
<td>Linux_zLPAR</td>
<td>51</td>
</tr>
<tr>
<td>AIX_SharedEthernetAdapter</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIX_ActiveMemorySharing</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIX_VirtualTargetDevice</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

One Subtype per Metric Category
RMF XP & SMF Records

<table>
<thead>
<tr>
<th>Metric Category</th>
<th>ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows on System x</td>
<td>60</td>
</tr>
<tr>
<td>Windows_LocalFileSystem</td>
<td>61</td>
</tr>
<tr>
<td>Windows_NetworkPort</td>
<td>62</td>
</tr>
<tr>
<td>Windows_OperatingSystem</td>
<td>63</td>
</tr>
<tr>
<td>Windows_Processor</td>
<td>64</td>
</tr>
<tr>
<td>Windows_Storage</td>
<td>65</td>
</tr>
</tbody>
</table>

SMF Data Flow Using Log Streams

- Program requests to write a SMF record
- SMF locates correct dataspace
- Locates appropriate buffer to write the record
- If full, buffer passed to task to be written to logstream
Usage and Invocation

- Define new logstreams in system logger
- Defining new keywords in SMFPRMxx
  - LSNAME(IFASMF.q1.q2,TYPE(xx:yy))
  - DEFAULTLSNAME(IFASMF.q1.q2)
  - RECORDING(DATASET|LOGSTREAM)
    - SETSMF operator command can be used to toggle recording settings
- Creating new JCL to use IFASMF_DL with new logstreams
- Update processes to use data from logstreams, if necessary
- Activate PARMLIB changes via IPL or SET SMF=xx command

z/OS Ver 2.1 - SMF Logger Updates

- Specify log stream buffer sizes with new DSPSIZMAX parameter in SMFPRMxx
  - Support for DSPSIZMAX to be used when SMF is initialized
    - also available for z/OS V1.12 and V1.13 with the PTF for APAR OA35175
  - z/OS V2.1 supports dynamic changes via SET SMF and SETSMF
- SMF also supports the use of data compression on zEC12 and zBC12 systems with the zEDC Express feature and the zEnterprise Data Compression (zEDC) feature for z/OS V2.1.
**zEnterprise Data Compression (zEDC)**

**Capture new opportunities with lower cost of keeping data online**

- Efficiently **compress active data** using a dedicated compression accelerator
- **Industry standard compression** for cross platform data distribution

**Typical Client Use Cases:**

- **Disk savings** with improved utilization of storage tiers with DFSMSdss™ use of compression
- **Compression for sequential files** with less CPU costs
- **Shorten encryption time** with hardware compression and IBM Encryption Facility for z/OS
- **Fast, secured data transfer across the enterprise** with IBM Sterling Connect:Direct for z/OS Standard Edition V5.2
- **Transparent acceleration** of Java compressed applications

**DFSMShsm™**

Use up to 58% less disk space and up to 80% less CPU compared to using DFSMSShm with the COMPACT keyword

**BSAM/QSAM**

Compress data up to 4X, with up to 80% reduced CPU *

**Java 7**

Up to 90% reduction in CPU time with up to 74% reduction in elapsed time vs. using zlib software

**Connect:Direct for z/OS 5.2***

Up to 80% reduction in elapsed time for z/OS to z/OS file transfers

---

**zEDC and SMF Logstream Data**

- New SMFPROMxx COMPRESS keyword on LSNAME and DEFAULTLSNAME
  - A buffer of SMF records is compressed by zEDC Express before it is written to the system logger
  - SMF data is only compressed while it is resident in the system logger
  - PERMFIX to specify amount of storage used for SMF buffers that can remain permanently fixed
  - When compressed data is processed by IFASMFDL, it decompresses the SMF records for selection and writing
  - SOFTINFLATE parameter to process compressed SMF records using software algorithm, for a pre-z/OS V2.1 system or no zEDC Express
**Logstream BufferParms**

- Program requests to write a SMF record
- SMF locates correct dataspace
- Locates appropriate buffer to write the record
- If full, buffer passed to task to be written to logstream

**SMF Logstream Data Flow and zEDC**

- SMF Records
- SMF Data Space 64k Buffers
- Compression Output Buffers
- SMF Data
- IFASMFDL
- Media Manager
- Staging Data set
- Offload Data sets
- IFASMFDL
- SMF Data
- inflate
- CF
- Fewer offloads should result in CPU savings
- Logger storage requirements minimized due to compression
- zEDC
- With compatibility PTFs software inflate can be done on downlevel z/OS or processor without zEDC
- CPU Reduction due to compression performed by zEDC
Obtain PCIe Information via API

- IQPINFO – Obtain PCIe Information
  - The IQPINFO service provides PCIe related information, including any performance statistics
  - The service is described in *MVS Programming: Authorized Assembler Services Reference*.
  - The response data area of the IQPINFO service is mapped by the macros:
    - IQPYPERF PCIE Performance Data Return Area
    - IQPPFMBPCIE Function Measurement Block
- RMF Monitor III Data Gatherer collects PCIe performance statistics frequently and writes new SMF Record Type 74 Subtype 9.
- The new RMF Postprocessor PCIE Activity Report provides detailed information about PCIE Express based functions. Currently supported functions are:
  - z Enterprise Data Compression (zEDC)
  - Shared Memory Communication via RDMA (SMC-R)

RMF Postprocessor PCIE Activity Report

- Basic PCIe Metrics, e.g. PCI Load/Store and DMA Operations
- Common Request Statistics across all Personalities (Compression and future personalities)
- Compression related Statistics
SMF 30 support for zEDC

- Using zEDC, CPU time for job could go down
- How much zEDC capacity a job used?
- APAR OA45767 adds zEDC usage info to Type 30 records
  - Total number of zEDC requests
  - Total queue time in microseconds
  - Total execute time in microseconds
  - Deflate and inflate statistics for number of bytes input and output
Mainframe and **private cloud**

- **What does it mean?**
  - You want to use the mainframe to deliver service to your enterprise using the characteristics of the public cloud service delivery model. These services could be IaaS, PaaS, SaaS, etc……

- **Why would you do it?**
  - To enhance your service delivery to be like public cloud, while taking advantage of the strengths of the mainframe
  - To save money through server consolidation
  - Because someone tells you to implement cloud in your current data center

---

**The layers of IT-as-a-Service**

- **Software as a Service**
  - Business Processes
  - Collaboration
  - Industry Applications
  - CRM/ERP/HR

- **Platform as a Service**
  - Middleware
  - Web 2.0 Application Runtime
  - Development Tooling
  - Java Runtime

- **Infrastructure as a Service**
  - Servers
  - Networking
  - Data Center Fabric
  - Storage

**Shared virtualized, dynamic provisioning**
Mainframe and hybrid cloud

- **What does it mean?**
  - You want to combine the strengths of the mainframe with the benefits of the public cloud service delivery model

- **Why would you do it?**
  - To use public cloud services in a "Fit for Purpose" model that follows the “Systems of Engagement” and “Systems of Record” architecture
  - Because it sounds like a cool thing to do
  - As a compromise between an all public cloud strategy and your current mainframe-based environment
Systems of Engagement and Systems of Record

Focus on point of interaction with the SOR

- Access Clients: Client systems and devices that interact with SOE
  - Not just mobile phones... any system or device... including mainframe programs

- Systems of Engagement: Systems that incorporate technologies which encourage peer interactions
  - This can be on System z, and in fact System z can make an excellent platform for SOE

- Systems of Record: Systems that host authoritative data sources for a given data element or piece of information
  - This doesn't have to be System z, but a great deal of SOR data is on the platform

Hybrid cloud example: z/OS and Softlayer

[Diagram showing the integration of SoftLayer and z/OS]
Mainframe as a Service

Another use-case for z/OS Connect is as a standard gateway into the z/OS LPAR to expose programs as a service:

z/OS Connect provides a way to do this with a single entry point (HA is possible) and common protocol (REST/JSON)

What is z/OS Connect?

1. z/OS Connect is software function that runs in Liberty Profile for z/OS.
2. z/OS Connect is described and configured in the Liberty server.xml file.
3. z/OS Connect is designed to accept RESTful URIs with JSON data payloads.
4. One part of z/OS Connect is a servlet that runs in Liberty Profile z/OS.
5. A ‘Service Provider’ is software that provides the connectivity to the backend system.
6. z/OS Connect provides the ability to transform JSON to the layout required by backend.
7. ‘Interceptors’ are callout points where software can be invoked to do things such as SAF authorization and SMF activity recording.
8. Initially the backend systems supported will be CICS, IMS and Batch.
**Context Within Overall Mobile Architecture**

The message here is that z/OS Connect is a piece of the Mobile architecture, but in most cases will not be the only component:

- **CICS**
- **IMS**
- **Batch**

![Diagram](image)

- Users of z/OS Connect would access through normal corporate firewall infrastructure.
- IBM MobileFirst Platform to provide application management, security and operational governance for mobile applications.
- z/OS Connect would be behind the secure firewall, and on LPARs along with backend systems.

**Liberty Profile z/OS**

Liberty Profile is IBM's dynamic and composable server runtime. First shipped with Version 8.5, it is available on many platforms, including z/OS:

- Single JVM per server model
- As opposed to the multiple JVM model of traditional WAS z/OS (the CR/SR model)

- Simple configuration structure
- One XML file serves as the main configuration file

- Dynamic
- Changes to the configuration file or to the applications are detected and dynamically loaded

- Composable
- You tell Liberty Profile what features and functions you want and only that code is loaded

- On z/OS can run from UNIX shell or as a z/OS started task
- On z/OS we anticipate most will run as started task

Liberty Profile is the basis for z/OS Connect, so any discussion of z/OS Connect necessarily involves Liberty.
Audit (SMF) Interceptor

The audit interceptor writes SMF 120.11 records with the following information captured:

**Server Identification Section**
- System Name
- Sysplex Name
- Jobname
- Job Prefix
- Address Space Stoken

**z/OS Connect User Data Section**
- Arrival Time
- Completion Time
- Target URI
- Input JSON Length
- Response JSON Length
- Method Name
- Service Name
- Userid

System z mobile pricing model

- Reduce z/OS peak MSUs attributable to mobile workloads -- up to 60%
- No Infrastructure Changes Required... (such as separate LPARs)

- Customers must tag and track z/OS CPU seconds from mobile workloads.
- New MWRT tool replaces SCRT and will subtract mobile CPU seconds from peaks.
Which tracking mechanism can be used for MWP?

1. Use an Individual LPAR for mobile-only workloads

2. Use Individual Regions for mobile-only workloads

3. Use Same Regions for mobile and non-mobile workloads

The WLM Enclave

An "enclave" is a way to identify and manage individual pieces of work within the many parts of a running z/OS system.

Key points from this chart:
- An "enclave" is simply a way for WLM to understand priorities at a work unit level.
- WAS does this automatically ... if you do no other configuration it'll still do this with default values.
Assigning a Service Class to the Enclave

Subsystem Type CB - WebSphere z/OS CN and TC Classifications

Classification:
- Default service class is CBDEF5
- Default report class is EWASDEF

<table>
<thead>
<tr>
<th>Qualifier</th>
<th>Type</th>
<th>Qualifier</th>
<th>Starting Position</th>
<th>Service Class</th>
<th>Report Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN</td>
<td>DFDMSR*</td>
<td>1</td>
<td></td>
<td>CBCLASS</td>
<td>DFDMSR</td>
</tr>
<tr>
<td>CN</td>
<td>DFRSR01*</td>
<td>2</td>
<td></td>
<td>CBCLASS</td>
<td>DFRSR01</td>
</tr>
<tr>
<td>TC</td>
<td>DFRTRAN1</td>
<td>3</td>
<td></td>
<td>DFRTRAN1</td>
<td>DFRTRAN1T</td>
</tr>
<tr>
<td>TC</td>
<td>DFRTRAN2</td>
<td>4</td>
<td></td>
<td>DFRTRAN2</td>
<td>DFRTRAN1T</td>
</tr>
<tr>
<td>TC</td>
<td>DFRTRAN3</td>
<td>5</td>
<td></td>
<td>DFRTRAN3</td>
<td>DFRTRAN3</td>
</tr>
</tbody>
</table>

Enclaves created in WAS CR are classified by rules in CB subsystem type:
1. CN of DFRSR01* matches the Deployment Manager. Work there goes to CBCLASS.
2. Work in DFRSR01* cluster with a transaction classification gets CBCLASS as well.
3. Work in DFRSR01* cluster with TC of DFRTRAN1 or DFRTRAN2 get service classes as shown.
4. Work that matches the TC of DFRTRAN3 regardless of WAS CN gets service class DFRTRAN3.
5. Anything that doesn't match any specific rules gets the default service class of CBDEF5.

The transaction class name file

The file supplies a set of criteria to match requests to transaction class names, which then match with rules in the CB subsystem type.

From that we get goals and importance based on specific transactions based on criteria in the classification XML file.
**What is the WAS for z/OS Liberty profile?**

- The WAS for z/OS Liberty profile is Liberty with *optional*, independently enabled *extensions* that exploit z/OS facilities
  - Only enable exploitation of z/OS features you need
  - Only configure the z/OS functions you use
- Focus of v8.5 is basic integration and exploitation

**Feature – z/OS Workload Manager**

- Adds support to classify HTTP requests with z/OS WLM
  - Classification associates response time goals and importance to work run in WebSphere
  - z/OS workload manager will manage the resources available on the system in a way that ensures the most important work runs while attempting to meet response time goals
  - RMF reports provide information about completed transactions, response times, etc by service class

```xml
<server description="myworkloadManagement">
  <featureManager>
    <feature>zoswlm-1.0</feature>
  </featureManager>

  <wlmClassification/>
  <httpClassification transactionClass="WLPTRADE" resource="/tradelite/4"/>
  <httpClassification transactionClass="WLPDFT"/>
</server>
```
IBM DB2 Analytics Accelerator

- DB2 detects WLM service class and importance level and sends it to the accelerator with each query.

- The accelerator maps the importance level to a Netezza priority and alters the session prior to the query execution, using the corresponding priority. Also, threads scheduled will have their priorities adjusted.

<table>
<thead>
<tr>
<th>WLM Importance Level</th>
<th>Netezza Priority</th>
<th>Version 3</th>
<th>Version 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Critical</td>
<td>Critical</td>
<td></td>
</tr>
<tr>
<td>Importance 1</td>
<td>Critical</td>
<td>Critical</td>
<td></td>
</tr>
<tr>
<td>Importance 2</td>
<td>High</td>
<td>Critical</td>
<td></td>
</tr>
<tr>
<td>Importance 3</td>
<td>Normal</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Importance 4</td>
<td>Normal</td>
<td>Normal</td>
<td></td>
</tr>
<tr>
<td>Importance 5</td>
<td>Normal</td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td>Discretionary</td>
<td>Low</td>
<td>Low</td>
<td></td>
</tr>
</tbody>
</table>
Agenda

■ The new world of RMF monitoring
  ► RMF XP and z/OSMF

■ z Systems hardware functions
  ► SMF logstreams and zEDC

■ z/OS in the new world of cloud, mobile and analytics
  ► z/OS Connect
  ► IDAA and WLM