WebSphere Application Server on z/OS What Can You Do With The SMF 120s?

David Follis
IBM

August 6, 2012
Session Number 11377

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

CICS*
DB2*
GDPS*
Geographically Dispersed Parallel Sysplex
HiperSockets
IBM*
IBM eServer
IBM logo*
IMS
On Demand Business logo
Parallel Sysplex*
RACF*
System z9
WebSphere*
z/OS
zSeries*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

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

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Disclaimer

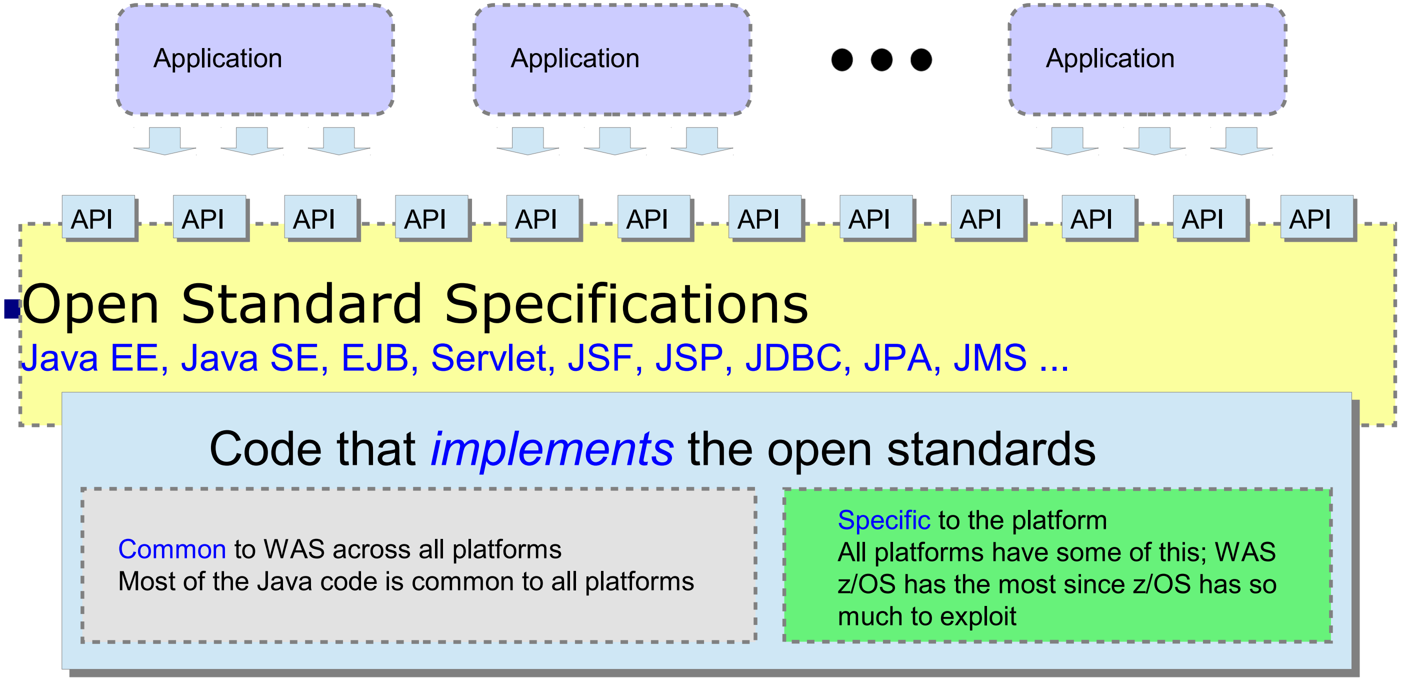
- The information contained in this documentation is provided for informational purposes only. While efforts were many to verify the completeness and accuracy of the information contained in this document, it is provided “as is” without warranty of any kind, express or implied.
- This information is based on IBM’s current product plans and strategy, which are subject to change without notice. IBM will not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation.
- Nothing contained in this documentation is intended to, nor shall have the effect of , creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of the IBM software.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

WebSphere Application Server on z/OS

Session	Day	Time	Room	Title	Speaker
11377	Monday	11:00	Grand Ballroom Salon B	What Can I Do With the SMF 120s?	David Follis
11371	Monday	3:00	Orange County Salon 2/3	Administrator Hands On Lab	David Follis / Michael Stephen / Ken Irwin
11378	Tuesday	12:15	Grand Ballroom Salon B	Spelunking the Admin Console	John Hutchinson
11375	Tuesday	4:30	Grand Ballroom Salon B	Being the Back-Up Administrator	Mike Loos
11374	Wednesday	11:00	Grand Ballroom Salon B	Liberty Profile – Rumors Dispelled	David Follis
11373	Thursday	4:30	Grand Ballroom Salon B	What's New?	David Follis / John Hutchinson / Michael Stephen
11370	Thursday	6:00	Grand Ballroom	Potpourri	Anybody
11376	Friday	8:00	Platinum Ballrom Salon 10	zWAS – In Real Life	David Follis / Rod Feak

"WAS is WAS" -- at Open Specification Layer

This is an important starting concept -- it's what makes application development a platform-neutral consideration:



Application

Application

• • •

Application

API API API API API API API API API API API API

Open Standard Specifications

Java EE, Java SE, EJB, Servlet, JSF, JSP, JDBC, JPA, JMS ...

Code that *implements* the open standards

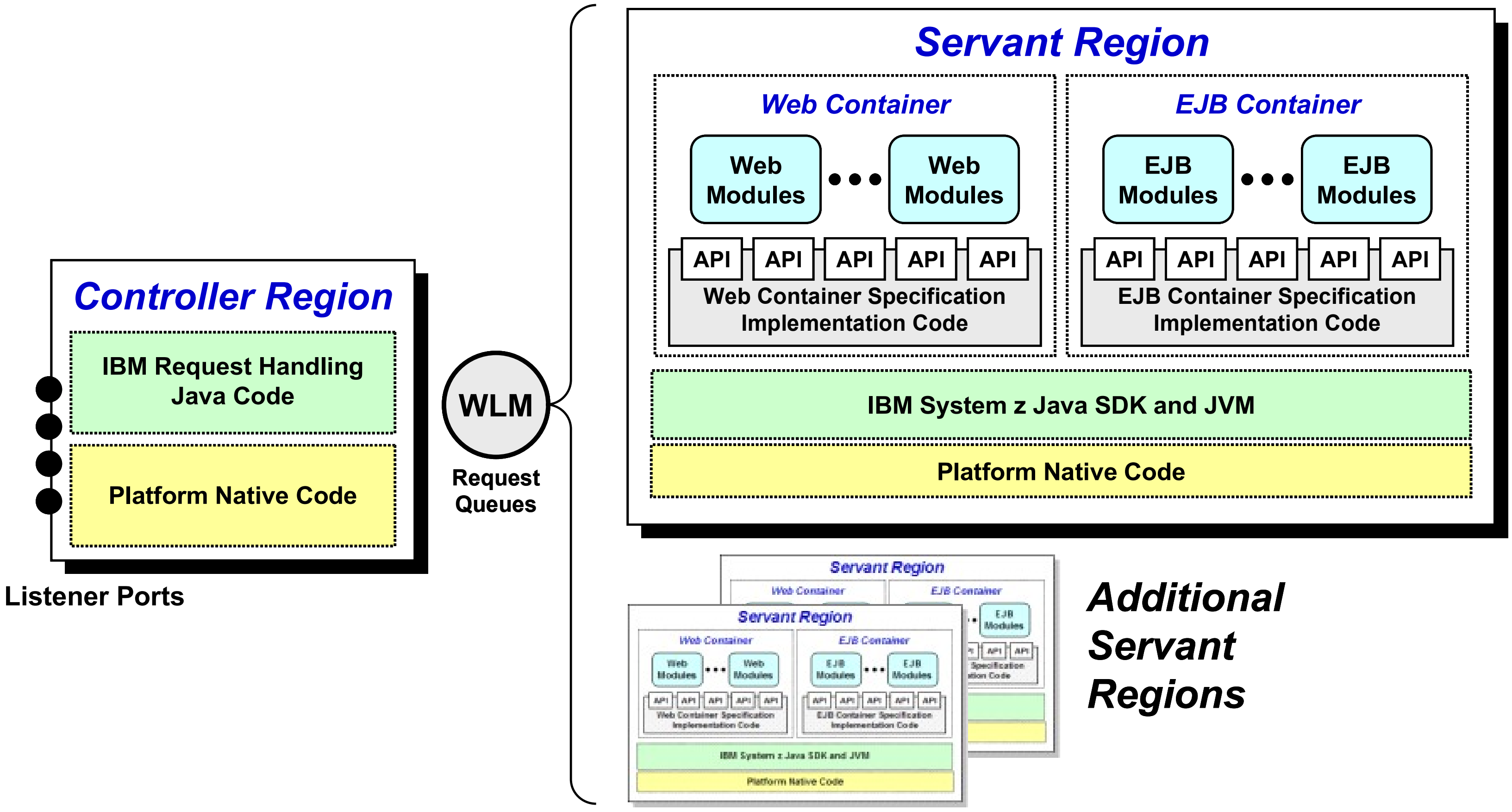
Common to WAS across all platforms
Most of the Java code is common to all platforms

Specific to the platform
All platforms have some of this; WAS z/OS has the most since z/OS has so much to exploit

WAS is common across platforms at this layer and above

Under the open specification line is where WAS takes advantage of platform attributes where they exist

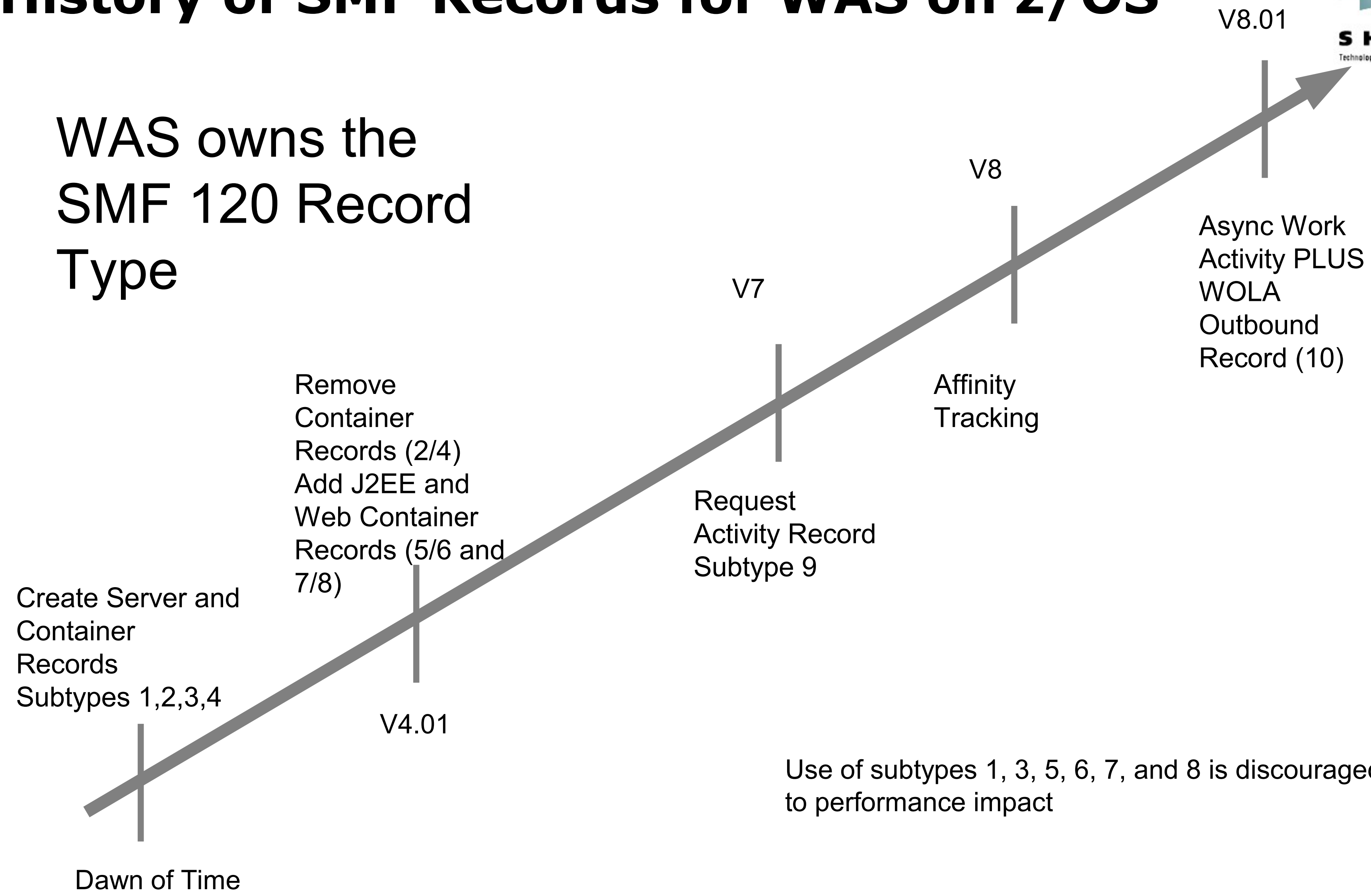
Structure of WAS on z/OS



z/OS Platform Functions: *WLM* *SMF* *SAF* *JES* *Cross-Memory*

History of SMF Records for WAS on z/OS

WAS owns the SMF 120 Record Type



Dawn of Time

Create Server and Container Records Subtypes 1,2,3,4

V4.01

Remove Container Records (2/4) Add J2EE and Web Container Records (5/6 and 7/8)

V7

Request Activity Record Subtype 9

V8

Affinity Tracking

V8.01

Async Work Activity PLUS WOLA Outbound Record (10)

Use of subtypes 1, 3, 5, 6, 7, and 8 is discouraged due to performance impact

Configuring the SMF 120 Subtype 9

Features can be turned on with variables, or dynamically:

- **Static definition using WebSphere variables:**

```
server_SMF_request_activity_enabled                 0 | 1
server_SMF_request_activity_CPU_detail              0 | 1
server_SMF_request_activity_timestamps              0 | 1
server_SMF_request_activity_security                0 | 1
```

- **Dynamically turn 120.9 records on and off, and set the level of details collected, through the MVS Modify (F) command:**

```
F <server>,SMF,REQUEST,[ON | OFF]
F <server>,SMF,REQUEST,CPU,[ON | OFF]
F <server>,SMF,REQUEST,TIMESTAMPS,[ON | OFF]
F <server>,SMF,REQUEST,SECURITY,[ON | OFF]
```

- **DISPLAY command tells you the status of SMF recording within a server**

```
F <server>,DISPLAY,SMF
```

What's in the SMF 120 Subtype 9 Record?

- Platform Neutral Server Information
- z/OS Specific Server Information
- Platform Neutral Request Information
- z/OS Specific Request Information
- Formatted Timestamps
- Network Data
- Classification Data
- Security Information
- CPU Usage Section
- User Data
- Async Work Section

Platform Neutral Server Information

- Always present, one instance
- Contains:
  - Cell name
  - Node name
  - Cluster name
  - Server name
  - Controller PID
  - WAS release (e.g. 7.0.0.19)
- Usage:
  - Record sorting by server (c/n/c/s is usually unique)
  - Change of PID for CR indicates server recycle (mostly)
  - Use release to identify perf. changes across maintenance

z/OS Server Information

- Always present, one instance
- Contains:
  - System name
  - Sysplex name
  - Controller jobname, jobid, stoken, ASID
  - Cluster and Server UUID (unique in 'universe' :-)
  - GMT offset for timestamps
  - Build level
- Usage:
  - Sorting by system, jobname
  - Stoken is a better indication of a recycle (unique in LPAR)
  - GMT offset is important for reports

Platform Neutral Request Information

- Present for regular requests (not async), one instance
- Contains:
  - Servant PID
  - Dispatch Task ID
  - TCB CPU time (all processor types)
  - Completion minor code
  - Request Type (HTTP, IIOP, etc)
- Usage:
  - More sorting
  - CPU time to run the request ON THE DISPATCH THREAD
  - Non-zero minor code indicates a problem (and probably missing data)
  - Request type helps filter data (internal vs. application)

z/OS Request Information (Part 1)

- Present for regular requests (not async), one instance
- Contains:
  - Timestamps (more on this later)
  - Servant jobname, JobID, Stoken, ASID
  - Dispatch TCB address and TTOKEN
  - TCB CPU time on zIIP, zAAP
  - Enclave Token
- Usage:
  - TTOKEN is unique for LPAR
  - Use with other fields to determine how many Servants and dispatch threads you are really using
  - Enclave token can be a correlator (not unique cross-plex, mostly unique in an LPAR)

z/OS Request Information (Part 2)

- Contains:
  - Enclave CPU times so far – IGNORE THESE
  - Enclave Delete CPU times (more later)
  - GTID value (Global Transaction ID)
  - Dispatch timeout value
  - Classification Transaction Class name
  - Flags (more on this later too)
- Usage:
  - Use the GTID to track related requests under the same global transaction (IIOP)
  - Group requests by Transaction Class or make sure your classification XML file is working as expected

z/OS Request Information (Part 3)

- Contains:
  - Granular RAS settings from classification XML:
    - Timeout dump actions
    - CPU time limit
    - DPM interval / action
    - Message Tag
  - Affinity data (more on this later)
- Usage:
  - Validate XML settings are working
  - Message tag can be a filter (e.g. by application)

Formatted Timestamps

- Contains:
  - Timestamps in human readable form
  - Optional
  - Mostly just makes the record bigger
  - Recommend you leave these off

Network Data Section

- Present for IIOP and HTTP requests, one instance
- Contains:
  - Bytes sent and received
  - Target port number (-1 for local)
  - Origin string (e.g. origin host/port or jobname/asid for local)
- Usage:
  - Sort by origin
  - Sort by target port
  - Correlate response times with size of request/response
  - Correlate long response times with network segments

Classification Data Section

- Multiple instances, depends on request type

- Contains:
  - HTTP: Target host/port and URI
  - IIOP: A/M/C names plus class and mangled method name
  - WOLA: CICS transaction name

- Usage:
  - Useful when validating classification XML is working
  - Sorting CPU and response times by specific application request (URI, EJB method)

Security Section

- Multiple instances, depends on data available, can be configured off
- Contains:
  - Server identity
  - Received identity
  - Invocation identity
- Usage:
  - Chargeback by who made the request
  - Or by the ID the request ran under (invocation vs. received)

CPU Usage Section

- Can be configured off
- Records based on container breach (e.g. servlet->EJB)
- Contains (for each thing recorded)
  - CPU time (all types)
  - Elapsed time
  - Number of invocations
  - Strings identifying the thing called
- Usage:
  - Can be used to roughly profile application
- NOTE: Enabling this introduces extra overhead

User Data Section

- Up to five sections
- Each section is 2K of user data
- Includes an identifying 'type' so you can tell how to format it
- Compute Grid uses an IBM reserved tag for 'job' data
- Can be set by the application
- Can be set by a servlet filter installed in the server
- See WP101859 for an example
- Usage:
  - Whatever you want!
  - Data we missed
  - Application specific data (function performed, etc)

Async Work Section (Part 1)

- Async Work is also a 120-9 record with:
  - Server identification sections
  - User Data, CPU Usage sections
- Contains:
  - Start and Completion timestamps
  - Servant identification (jobname, jobid, stoken, asid)
  - Execution context thread identification (tcb, ttoken)
  - Dispatch thread identification (tcb, ttoken)
  - Execution context and dispatch enclave token
- Usage:
  - Where did it run, how long did it take
  - Correlate to request that scheduled it

Async Work Section (Part 2)

- Contains:
  - Transaction class if an enclave was created
  - Daemon work?
  - Enclave CPU so far – IGNORE
  - Dispatch TCB on GPs and on zIIP/zAAP
  - Work classname and package
  - Workmanager name
- Usage:
  - CPU time for the work
  - Help figure out how to use enclave CPU time (Daemon etc).
  - What was it? (Work name, workmanager name)

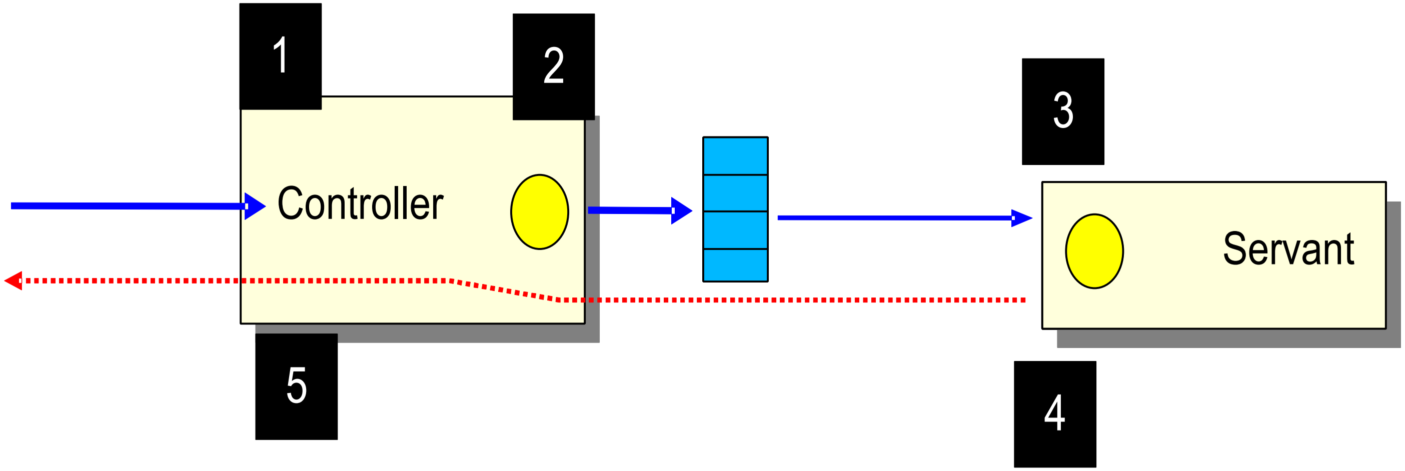
CPU Times – A closer Look

- Task CPU times
  - CPU time spent on the dispatch thread
  - Broken into 'all types' and 'just zAAP/zIIP'
  - Grabbed at slightly different times than enclave values
  - If minor code != zero, no 'end' CPU time grabbed so calculated values are usually negative
- Enclave CPU times
  - For IIOP requests to local servers enclave may propagate
    - Thus enclave values include time in other servers
  - For IIOP requests in a global transaction, enclave is reused
    - Thus values cover multiple requests, missing on first through N-1 requests

Enclave CPU values

- For Enclave CPU we get from WLM:
  - Time on all processor types
  - Time on zAAP and Time on zIIP
    - If zAAP on zIIP then zAAP time reported as zIIP
  - Normalization factors if at different speeds
    - A value of 256 means same speed
  - CPU used in service units (MSUs)
  - Response Time Ratio
    - 1<value<1000
    - Actual / Goal * 100
    - Identify requests that missed goal
    - Correlate with request/response size or time of day or origin of request, application URI or tag

Timestamps



1) Arrival Time
2) Time placed on the queue
3) Begin Dispatch
4) Complete Dispatch
5) Response sent (request finished)

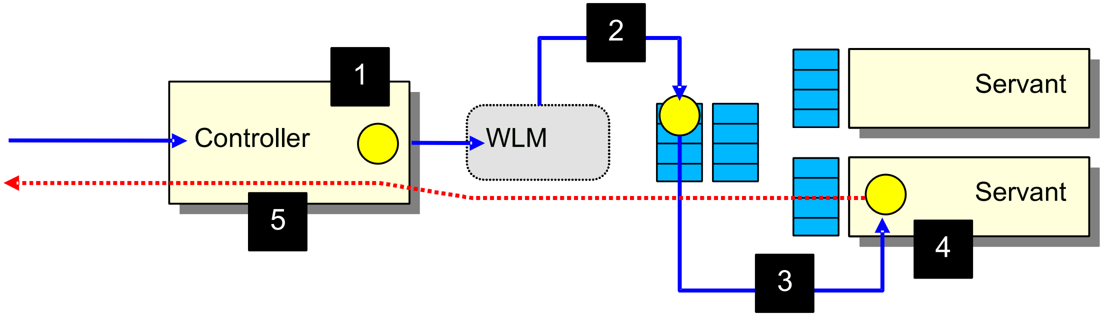
Values are in STCK format
Subtract and divide by 4096 to get (in microseconds):

- Time in CR
- Time on Queue
- Time in Dispatch
- Time to send Response
- Overall Response Time

Flags!

- Contains:
  - Created an enclave?
  - One-way Request
  - Classification Trace Match
  - SMF record on (from XML)
  - Queued with affinity
- Usage:
  - Helpful with tracking enclave propagation
  - One way requests don't have a response
  - If using classification tracing, records with flag on produced trace data too
  - Affinity – next chart...

SMF 120-9 Updates for affinity routing



- Some requests establish an affinity to a servant region

- Later requests use that affinity and must run in the same servant
  - HTTPSession and Stateful Session Beans are examples

- The SMF 120.9 record already indicates if a request ran in a particular servant because of an affinity

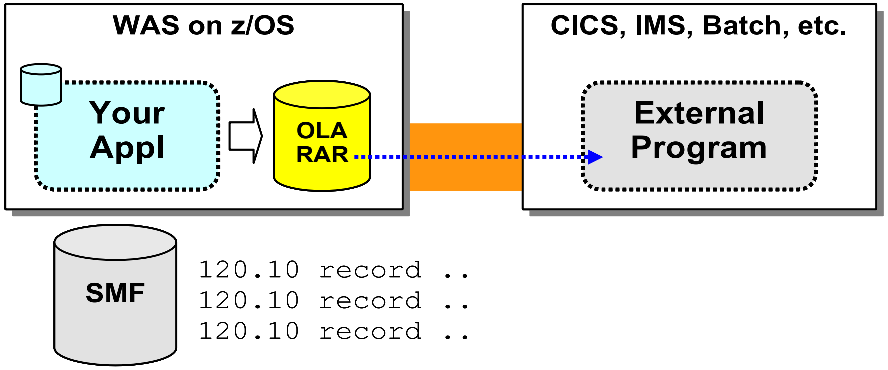
- In **Version 8** we added an affinity token to the SMF record

- Find the request that created the affinity and all the later requests that used it

Uses for Affinity Data

- Find all the requests that establish an affinity
- Find all the requests that use an affinity
- Build the chains and accumulate data for a set of affinity related requests
- Identify affinities that are never used again
- Build historical data about 'average' affinity usage and flag outlying cases
    - Shopping carts created but never buys anything

SMF 120-10 and CICS Correlation with WOLA

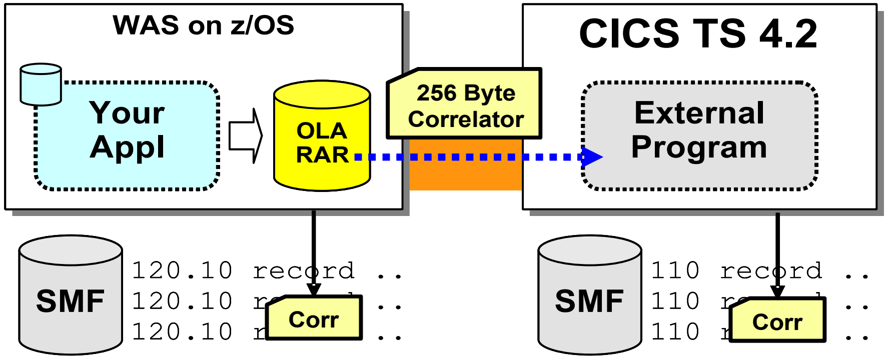


Similar to WAS z/OS 120.9 records

120.9 records inbound calls, the new 120.10 is used to record *outbound* calls

Good information about content and performance of outbound calls

InfoCenter: rtrb_SMFsubtype10



Part of the SMF 120.10 record function

256 bytes of specific information about the outbound request

With CICS 4.2 the correlator ends up in the CICS 110 records as well

InfoCenter: rtrb_SMFsubtype10

SMF 120 Subtype 10

- Contents
  - Servant jobname, jobid, stoken, ASID, process id
  - Task originating request: TCB, TTOKEN, Task ID
  - Enclave token (to correlate with 120-9)
  - Bytes sent and received
  - Timestamps for request sent and response received
  - WOLA register name and service name
  - Transaction context, Userid, CICS or IMS-OTMA info
- Usage
  - Correlate to WAS request driving into CICS/IMS
  - Correlate to CICS SMF data
  - Determine if WAS response time delays are coming from CICS or IMS reached via WOLA

This is all nice, but SO WHAT?

- Most customers use Type 30's or Type 72s to get the big picture
- WAS SMF 120's provide the details
    - Use received timestamp to calculate inbound requests per second
    - Find patterns for requests that don't meet goals or use more CPU than usual
    - Spot unusual activity in an application
        - Affinity usage (or lack thereof)
        - CPU usage detail showing different usage patterns
        - Unusually large requests/responses
    - Use a servlet filter to get your own info in the User Data
- Use the IBM SMF 120 Browser and write your own plugins!
    - WP101726 shows you how
    - Share 'em with your friends!

System z Social Media

- **System z official Twitter handle:**
  - @ibm_system_z

- **Top Facebook pages related to System z:**
  - Systemz Mainframe
  - IBM System z on Campus
  - IBM Mainframe Professionals
  - Millennial Mainframer

- **Top LinkedIn Groups related to System z:**
  - Mainframe Experts Network
  - Mainframe
  - IBM Mainframe
  - System z Advocates
  - Cloud Mainframe Computing

- **YouTube**
  - IBM System z

- **Leading Blogs related to System z:**
  - Evangelizing Mainframe (Destination z blog)
  - Mainframe Performance Topics
  - Common Sense
  - Enterprise Class Innovation: System z perspectives
  - Mainframe
  - MainframeZone
  - Smarter Computing Blog
  - Millennial Mainframer