WebSphere Application Server:
z/OS Exploitation and Differentiation

David Follis
WebSphere Application Server for z/OS Development
IBM Corporation

Thursday, March 3, 2011, 3:00 PM to 4:00 PM
Session 8384

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

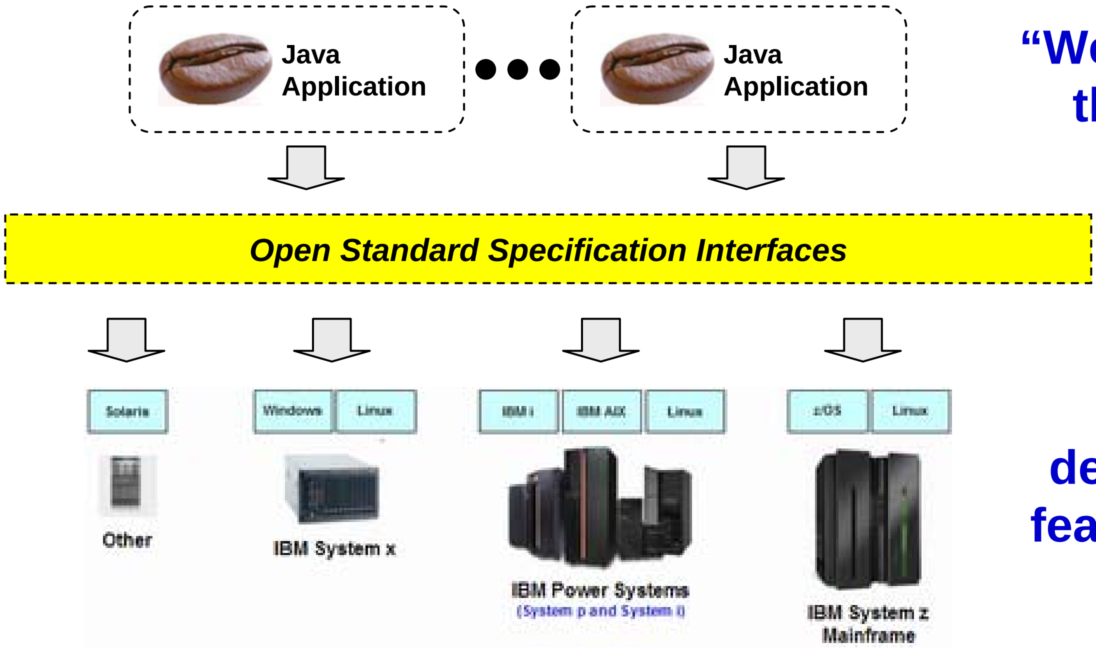
- The information contained in this documentation is provided for informational purposes only. While efforts were many to verify the completeness and accuracy of the information contained in this document, it is provided “as is” without warranty of any kind, express or implied.
- This information is 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 Sessions

Room	Day	Time	Title	Speaker
208B	Monday	11:00	Lab	Multi
201A	Monday	11:00	The Value of the WebSphere Application Server Job Manager	Loos
205A	Monday	4:30	WebSphere Application Server for z/OS -- I am No Longer a Dummy but...	Hutchinson
205B	Tuesday	9:30	Performance Tuning for WebSphere Application Server for z/OS - Practical Advice	Everett
205A	Wednesday	4:30	WebSphere Application Server for z/OS: Tools and Tricks (Potpourri)	Loos and Co.
205A	Wednesday	6:00	WebSphere Application Server for z/OS: Helping Customers Help Themselves	Stephen
206B	Thursday	8:00	Securing WebSphere Application Server for z/OS	Kearney
206B	Thursday	9:30	Application Improvement and Savings Through Simplification	McCorkle
206B	Thursday	11:00	WebSphere Application Server for z/OS: Batch	Bagwell
206A	Thursday	12:15	WebSphere Application Server 101	Stephen
206B	Thursday	1:30	WebSphere Application Server for z/OS: Availability Considerations	Bagwell
206B	Thursday	3:00	WebSphere Application Server: z/OS Exploitation/Differentiation	Follis
206B	Thursday	4:30	Performance Tuning for WebSphere Application Server for z/OS - WAS and WLM Interactions and Concepts	Follis

Very Important Starting Concept

This point can’t be stressed enough -- the differentiation is *not* in the open standard specification support offered. *That is common across platforms!*



Java Application ••• Java Application

Open Standard Specification Interfaces

Java EE, Java SE, EJB, Servlet, JSP, JDBC, JCA, JMS, Web Services, etc.

“WebSphere is WebSphere” above the specification interface line

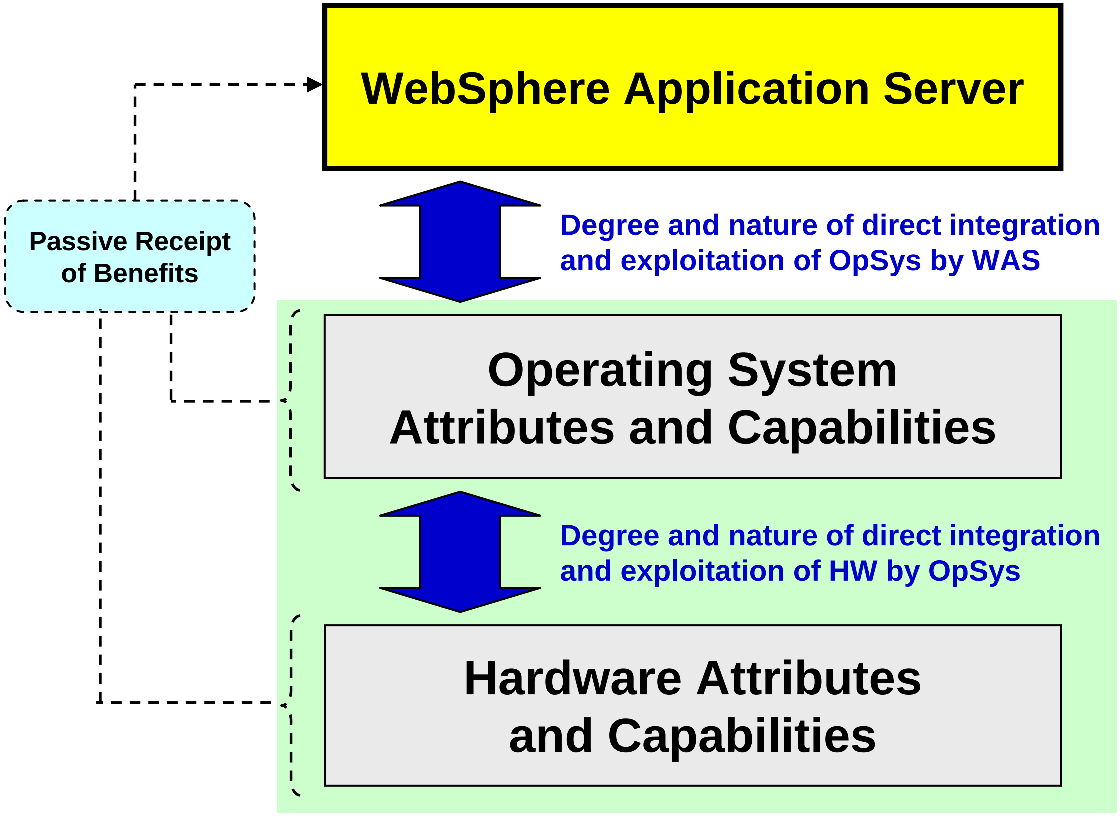
How it’s *implemented* is dependent on the platform … its features, functions, attributes and qualities of service

Starting with V6.0 the code base merged into one with a single source

Problems with code divergence solved. Code has ability to detect platform and invoke platform-specific exploitation as appropriate.

Multiple Levels of Exploitation Taking Place

We need to understand that there are benefits from the hardware design, benefits from the operating system design, *and benefits from the integration between the two*



WebSphere Application Server
Degree and nature of direct integration and exploitation of OpSys by WAS
Passive Receipt of Benefits
Operating System Attributes and Capabilities
Degree and nature of direct integration and exploitation of HW by OpSys
Hardware Attributes and Capabilities

Not all hardware designs are equal

Not all operating systems are equal

Not all operating systems have the same degree of integration with the hardware

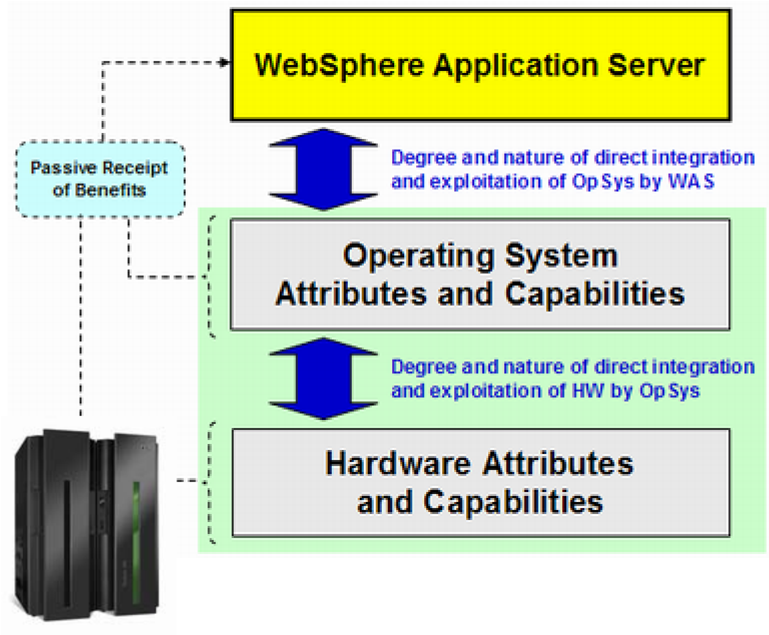
Example: z/OS only runs on System z … there are no tradeoffs to enable multi-platform flexibility. The OS is optimized for the hardware; the two are developed jointly

That doesn't mean System z and z/OS are appropriate for all cases

Nor does it mean other platforms and operating systems can do what System z and z/OS can do

Passive Benefits Fall Into Several Categories

Programs that run on System z and z/OS receive passive benefits in a couple of different areas:


WebSphere Application Server
Passive Receipt of Benefits
Degree and nature of direct integration and exploitation of Op Sys by WAS
Operating System Attributes and Capabilities
Degree and nature of direct integration and exploitation of HW by Op Sys
Hardware Attributes and Capabilities

Hardware

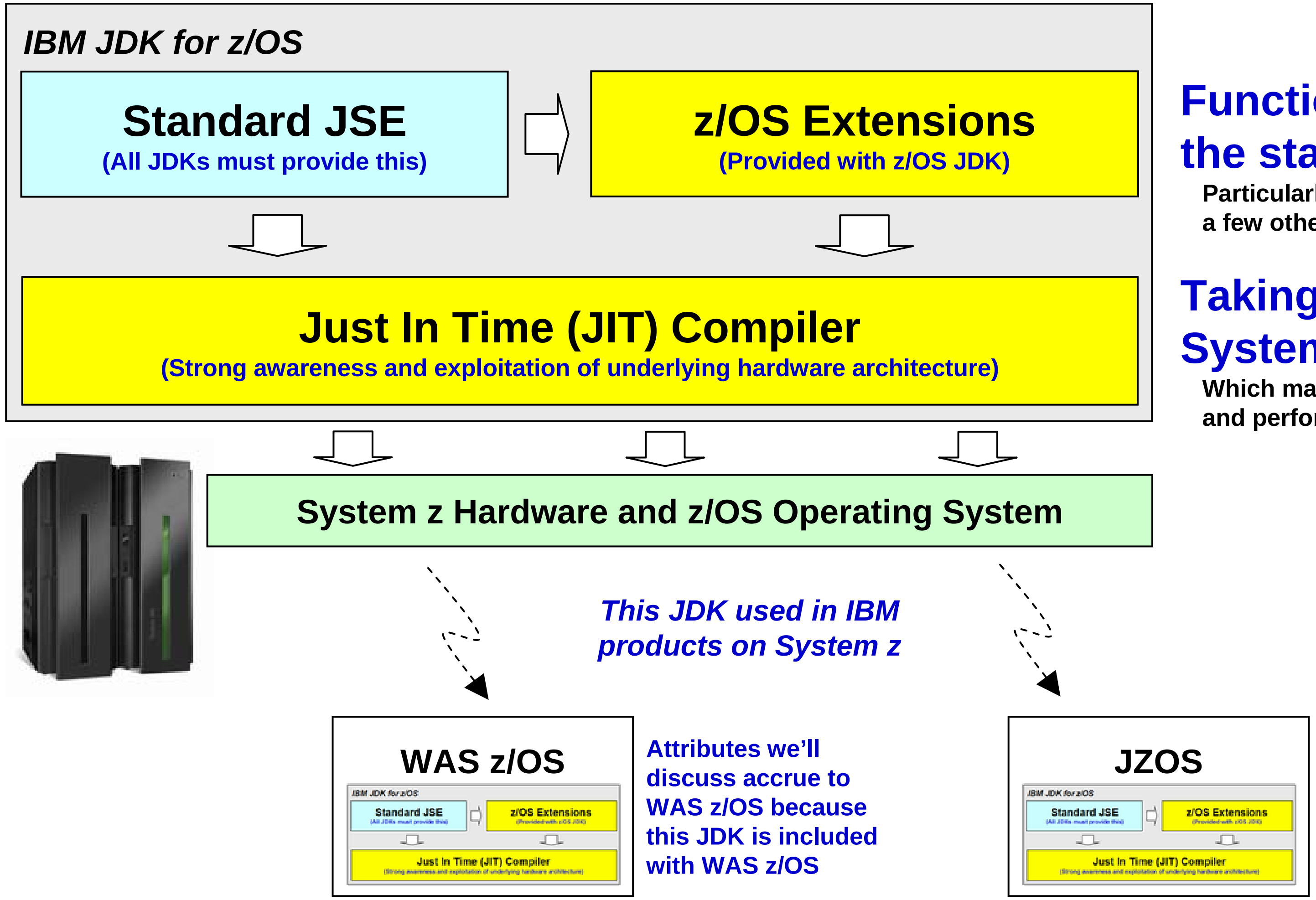
- Inherent maturity and stability of design
- Redundancy and flexible updates
- Balanced design offers very high throughput
- Mature and proven virtualization through LPAR

Operating System

- Tight integration with server hardware design
- Extremely mature architecture
- Storage protection
- Workload Manager (WLM)
- Intelligent Resource Director (IRD)
- Local TCP optimization
- Mature systems management tools
- Proven disaster recovery capabilities

Java Exploitation of System Z and z/OS

"Java is Java" but not all JDKs are the same, and not all JDKs are fully aware of and exploit the underlying platform



IBM JDK for z/OS

Standard JSE
(All JDKs must provide this)

z/OS Extensions
(Provided with z/OS JDK)

Just In Time (JIT) Compiler
(Strong awareness and exploitation of underlying hardware architecture)

System z Hardware and z/OS Operating System

This JDK used in IBM products on System z

WAS z/OS

Attributes we'll discuss accrue to WAS z/OS because this JDK is included with WAS z/OS

JZOS

Functionality *beyond* the standard JSE

Particularly in the area of security, but a few other things as well as we'll see

Taking advantage of System z hardware

Which manifests itself in efficiency and performance

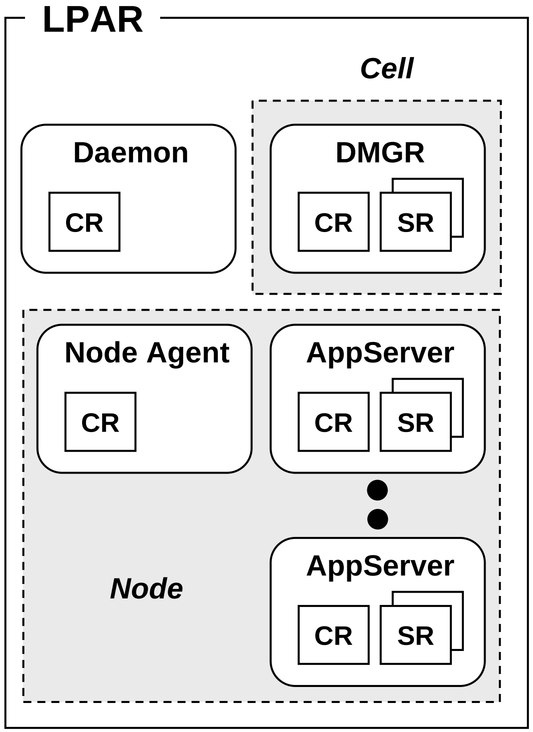
JZOS is a batch Java launching tool.

JZOS extends the z/OS-specific functionality even further

WebSphere Exploitation of System Z and z/OS

Exploitation of JES and Common z/OS Facilities

And that means that existing z/OS system programmers will be comfortable with the essential operations of WAS z/OS … it maps to their present skills



- WAS z/OS runtime implemented as a series of started tasks
- Standard JCL and START commands employed
- JCL START procedures maintained in PROCLIB
- Output written by default to JES
- JES manages output and storage
- Started tasks and address spaces displayable like any other
- Started and stopped like any other
- Able to use MODIFY commands for dynamic operations
- Configuration held in HFS or ZFS file systems
- Allows system automation tasks to control operations

This is all standard stuff. The key is that WAS z/OS was implemented to be compatible with existing z/OS skills, and to take advantage of existing z/OS facilities. WAS z/OS is *not* merely UNIX processes running in USS.

Handling of output

- Standard Out (stdout) and Standard Error (stderr) delivered to the SYSPRINT and SYSOUT DD cards in the JCL Procedure
- Standard JES mechanisms like spinning the output for purge/print work
- Can be redirected anywhere a DD card can go
  - Commonly to a file in the USS File system to be ‘like distributed’
- WAS ‘error’ messages can be redirected from SYSOUT to a z/OS Logstream

Display Command

- Syntax: MODIFY <server>,DISPLAY,HELP
- Basic command indicates server is alive and tells you the service level
- DISPLAY,SERVERS lists all the servers in this cell in this sysplex
- DISPLAY,SERVANTS lists the ASIDs of the servants owned by this controller
- DISPLAY,LISTENERS displays information about TCP/IP ports we are listening on
- DISPLAY,CONNECTIONS displays information about connections through TCP/IP
- DISPLAY,TRACE tells you what tracing options are on

Display Command (more)

- DISPLAY,JVMHEAP gives you information about the current JVM heap allocation
- DISPLAY,WORK displays counters showing how many requests have been processed by the server
  - A subcommand, DISPLAY,WORK,CLINFO shows information about how the WLM classification XML file has been used
- DISPLAY,ERRLOG shows the last ten entries written to the error log
- DISPLAY,MODE shows the bit mode the server is executing in (31/64).
- DISPLAY,THREADS shows active dispatch threads, timed out requests, etc.
- DISPLAY,ADAPTER shows information about WOLA

Display Command (more)

- DISPLAY,OLATRACE shows information about WOLA tracing
- DISPLAY,WLM shows the current min/max number of servants
- DISPLAY,SMF provides information about SMF 120-9 recording
- DISPLAY,FRCA shows statistics about the WAS FRCA cache
- DISPLAY,DPM shows the current settings for the dispatch progress monitor

Modify Command

- Dynamically change tracing (native and Java)
- Reset to initial trace configuration
- Force all tracing off
- Trigger a dump of java call stacks, a javacore, heapdump, or Java TDUMP
- Switch tracing from buffer to SYSPRINT and back
- Dynamically change the min/max number of servant regions
- Change timeout dump and delay options
- Configure the Dispatch Progress Monitor
- Change SMF recording options
- Pause/Resume listeners (more on this later)

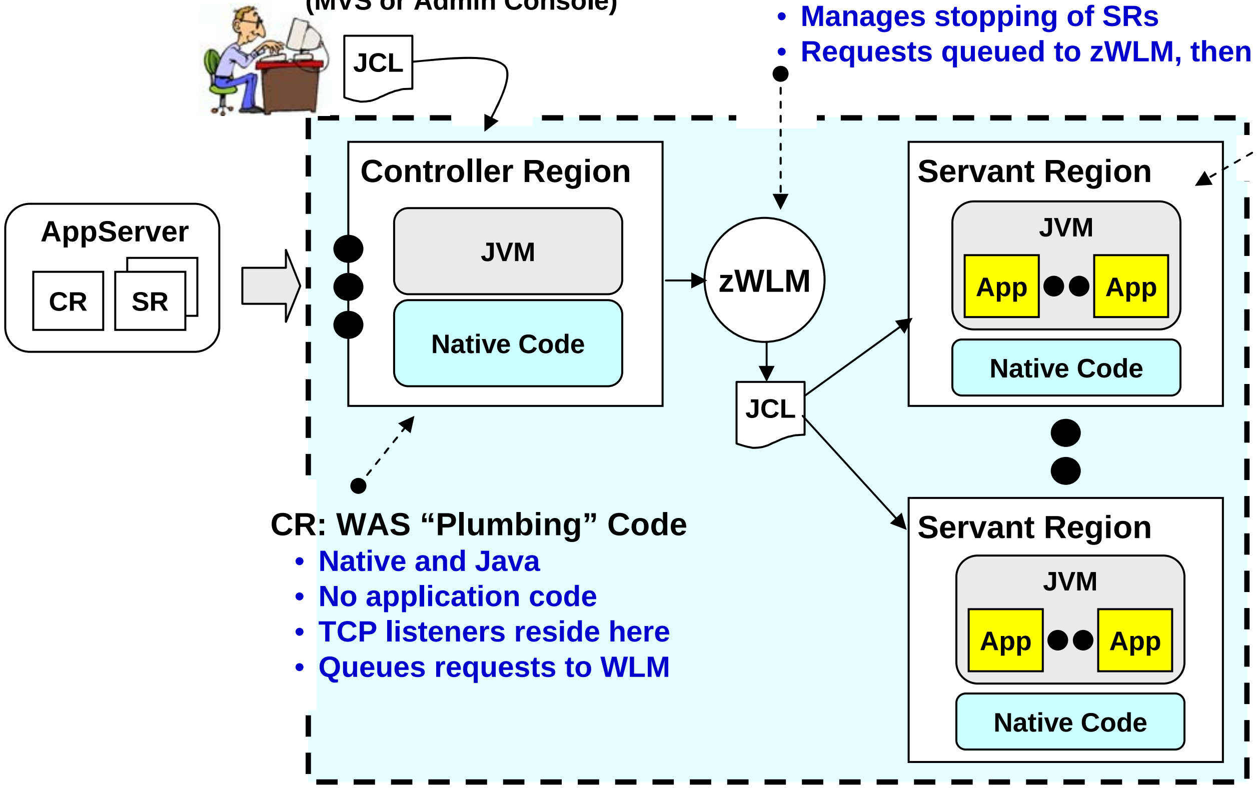
WTOs for Automation

- Besides responses to the display commands
- WAS on z/OS issues WTOs to hardcopy and glass to interact with your operators (real and automated)
- For example:

```
BBOO0001I WEBSPHERE FOR Z/OS CONTROL PROCESS %s/%s/%s/%s IS STARTING.
BBOO0002I WEBSPHERE FOR Z/OS CONTROL PROCESS %s ENDED NORMALLY.
BBOO0003E WEBSPHERE FOR Z/OS CONTROL PROCESS %s ENDED ABNORMALLY, REASON=%X.
BBOO0019I INITIALIZATION COMPLETE FOR WEBSPHERE FOR Z/OS CONTROL PROCESS %s.
BBOO0299I SERVER %s/%s/%s HAS NO SERVANTS. WORK IS BEING REJECTED.
```

The Controller / Servant Architecture

This is a unique architectural element to the WAS z/OS design. No other platform has this design because no other platform has WLM**:



zWLM

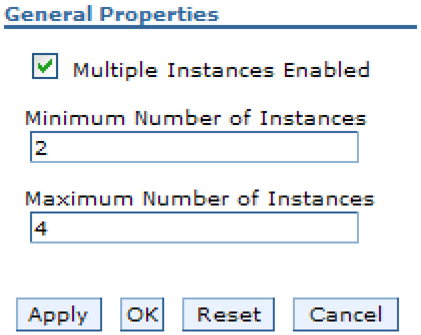
- Manages starting of SRs
- Manages stopping of SRs
- Requests queued to zWLM, then to SR

SR: Application Infrastructure

- Maintains app JVM runtime
- May support one or more applications
- Connectivity to data resources from SR
- Min/Max controllable by admin

CR: WAS “Plumbing” Code

- Native and Java
- No application code
- TCP listeners reside here
- Queues requests to WLM



General Properties

☑ Multiple Instances Enabled

Minimum Number of Instances
2

Maximum Number of Instances
4

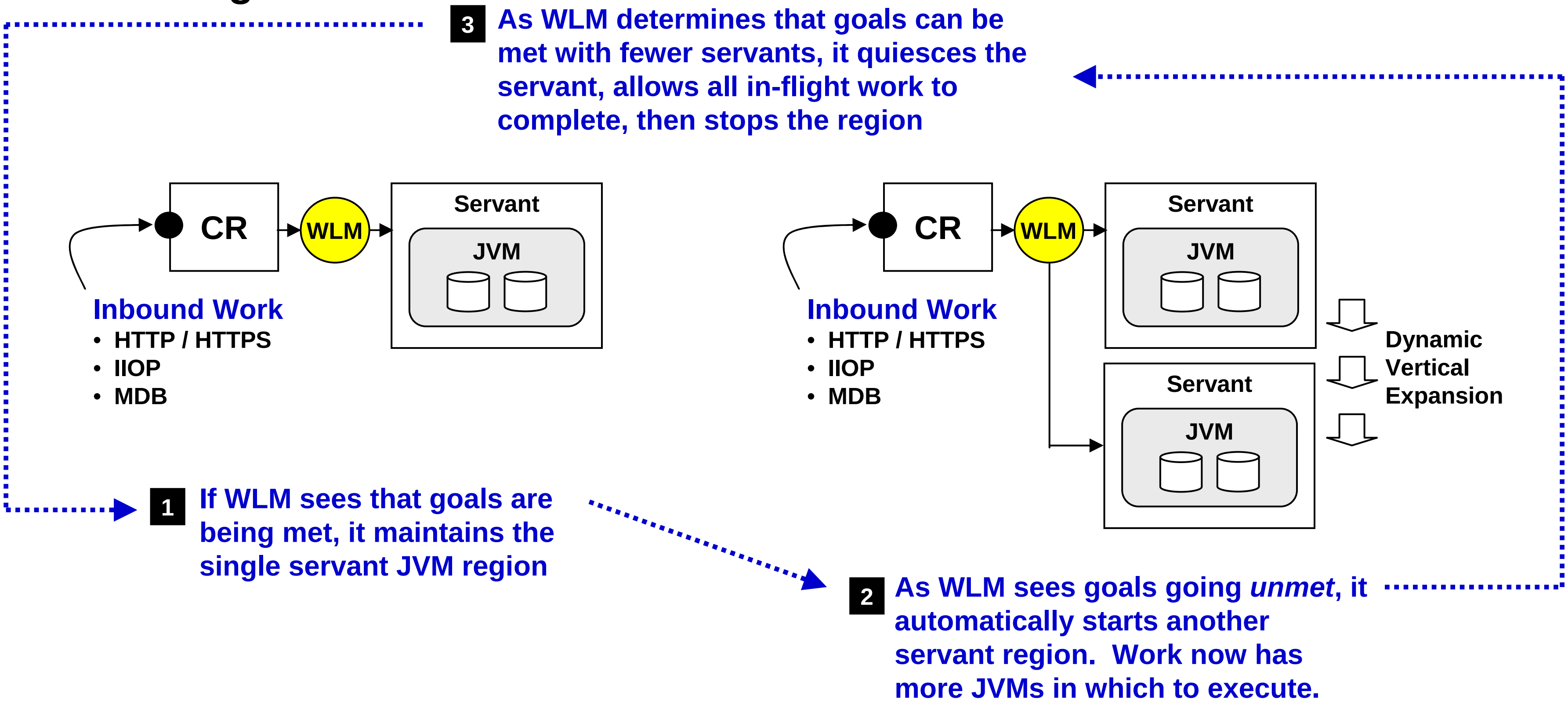
Default: min=1, max=1

Let’s now explore how this is accomplished …

** WebSphere on distributed uses the phrase “Workload Management” but it’s not the same as zWLM

Intelligent Dynamic Capacity Expansion

This is the “vertical scaling” capability of the multi-Servant structure. If allowed, WLM will start additional servant regions if it sees unmet goals:



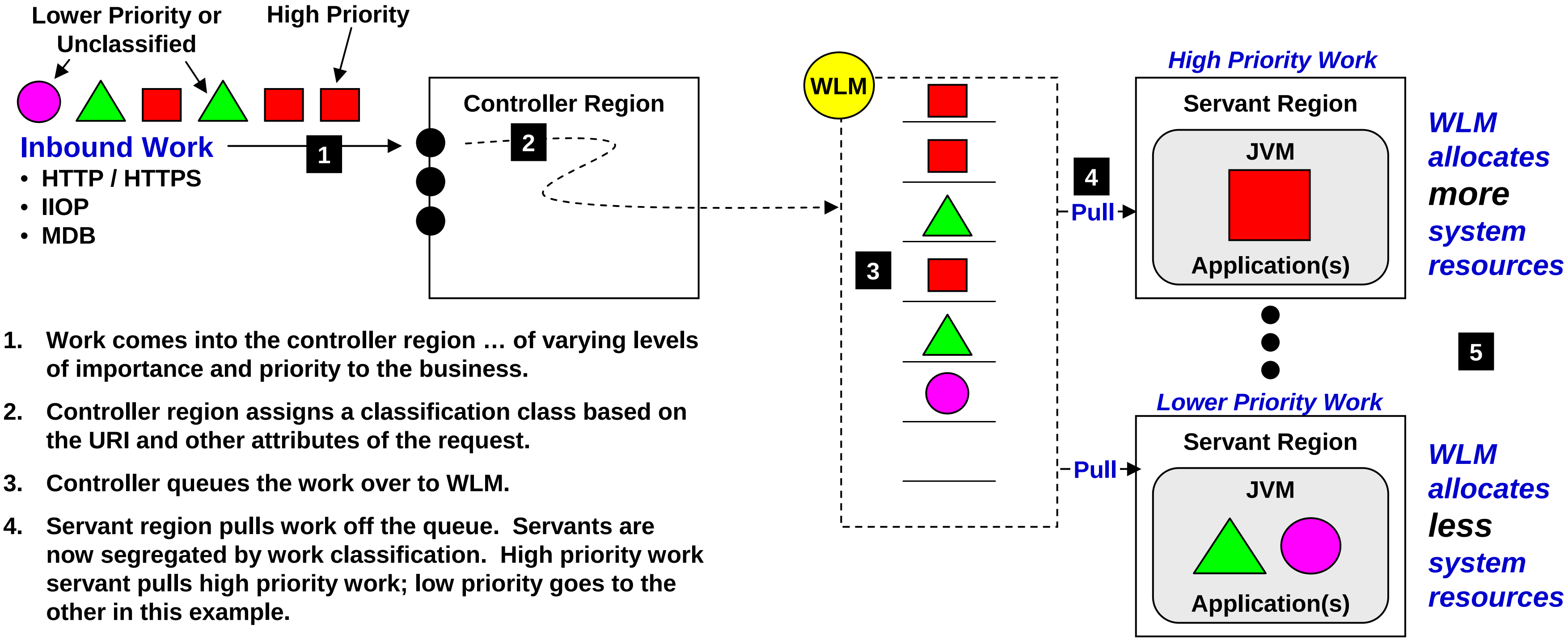
3 As WLM determines that goals can be met with fewer servants, it quiesces the servant, allows all in-flight work to complete, then stops the region

1 If WLM sees that goals are being met, it maintains the single servant JVM region

2 As WLM sees goals going *unmet*, it automatically starts another servant region. Work now has more JVMs in which to execute.

Intelligent Management of Mixed Work in Server

This involves inbound work being given a “Transaction Classification.” With that, the CR can direct work to servants and WLM can manage:



Lower Priority or Unclassified

High Priority

Inbound Work

- HTTP / HTTPS
- IIOP
- MDB

1

Controller Region

2

WLM

3

4

Pull

Pull

High Priority Work

Servant Region

JVM

Application(s)

WLM allocates **more** *system resources*

5

Lower Priority Work

Servant Region

JVM

Application(s)

WLM allocates **less** *system resources*

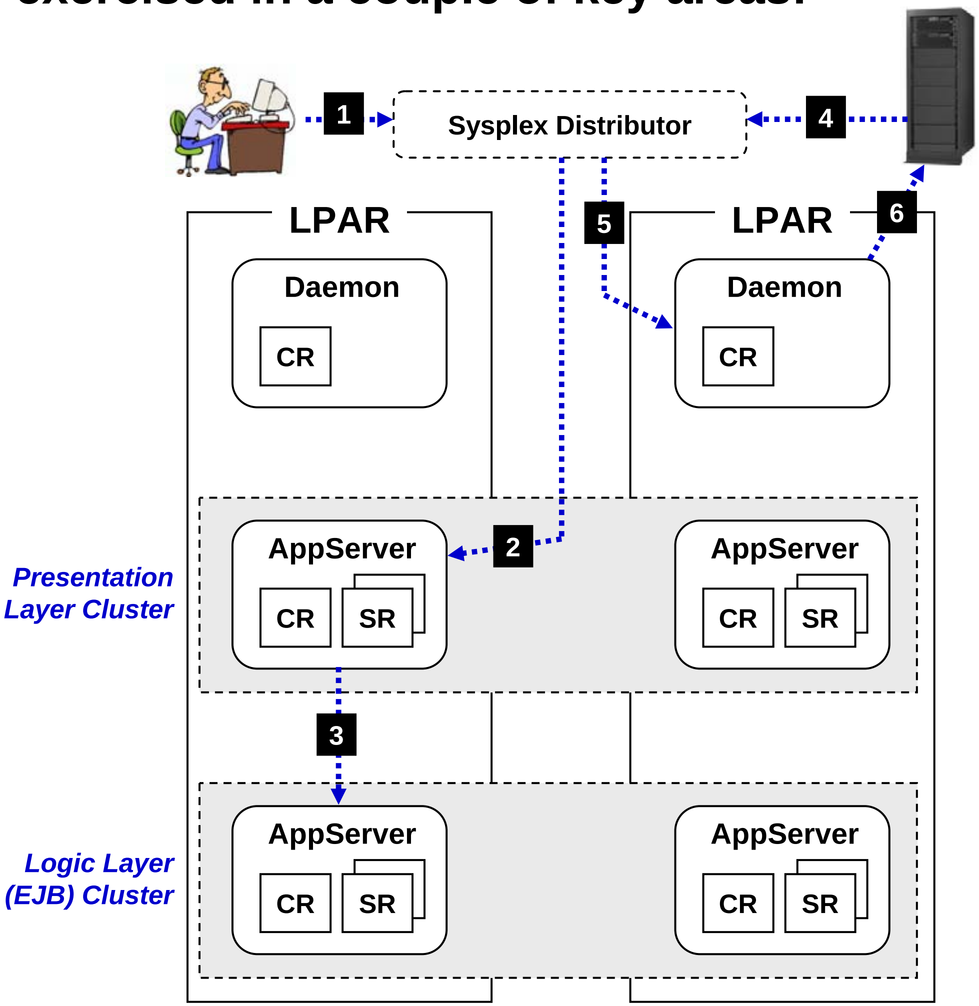
1. Work comes into the controller region … of varying levels of importance and priority to the business.
2. Controller region assigns a classification class based on the URI and other attributes of the request.
3. Controller queues the work over to WLM.
4. Servant region pulls work off the queue. Servants are now segregated by work classification. High priority work servant pulls high priority work; low priority goes to the other in this example.
5. With work segregated by servant region, WLM can now manage the system resources given to each servant. High priority work gets more, lower priority less, all according to defined WLM goals.

Sophisticated Work Prioritization

On other platforms this can only be done by allocating work to separate servers. No WLM there to manage at this level.

Intelligent Workload Routing Advice

WAS z/OS relies on WLM to make routing decisions. We see this exercised in a couple of key areas:



Inbound HTTP or IIOP Work

1. Work comes in over network aimed at DVIPA and Sysplex Distributor
2. WLM advises Sysplex Distributor of presentation cluster member to place TCP request to
3. For servlet-to-EJB flow (IIOP) WLM advises server, which then places the IIOP request to one of the members in the EJB cluster

IOR Resolution

4. External client seeks to use EJB deployed in WAS. It addresses itself to DVIPA/Sysplex Distributor for bootstrap (not shown on picture), then gets DVIPA host for Daemon location service.
5. External client then come back to DVIPA/SD for Daemon location service. WLM advises Sysplex Distributor for best Daemon to place request.
6. Daemon consults WLM for object location best able to service external client at that time. External client provided with host:port for EJB in the cluster.

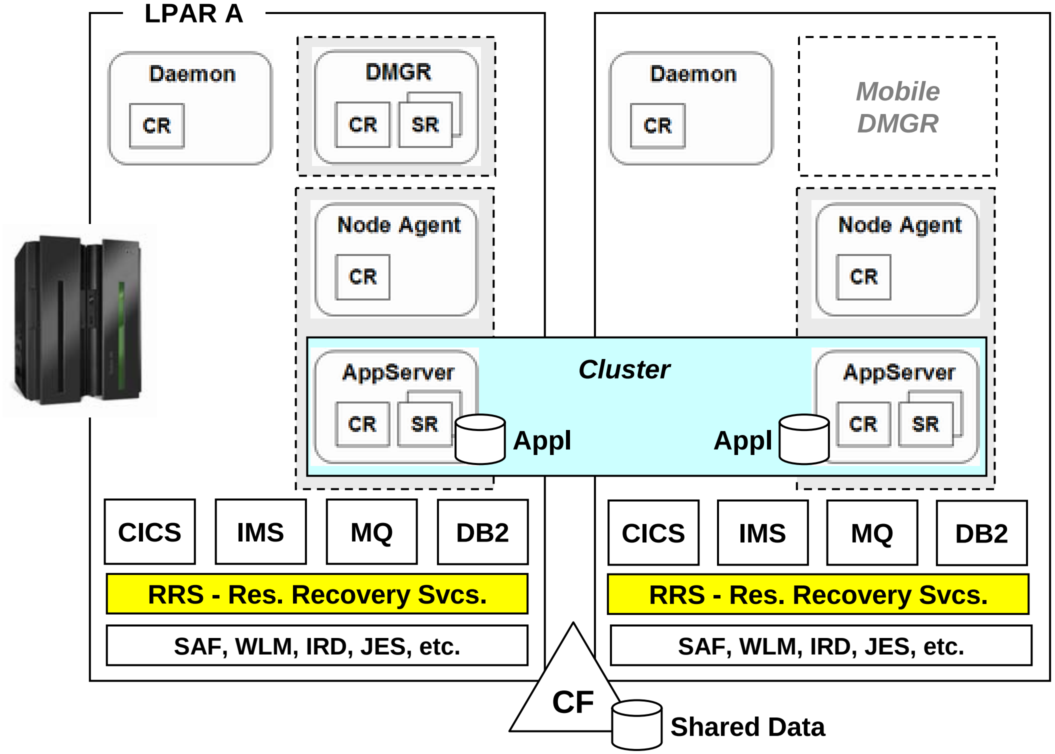
The point here is that WLM plays a key role in the routing decisions made by clients and WAS itself for real-time routing

More Exploitation of WLM

- WLM Enclaves are created for every request dispatched in the server
- This allows WLM to manage system resources used by the work to meet the goals defined for work
- It also allows RMF to report how z/OS is doing trying to meet the goals
- Enclaves are created for directly dispatched requests
  - HTTP, IIOP, etc.
- Enclaves are created for async work
  - Async Beans, ‘Batch’
- Enclaves are created for internal work
  - MBeans, etc.

Exploitation of Resource Recovery Services (RRS)

Two-phase commit processing involves coordination of participants to make sure all are ready to commit. RRS plays that role in Parallel Sysplex:



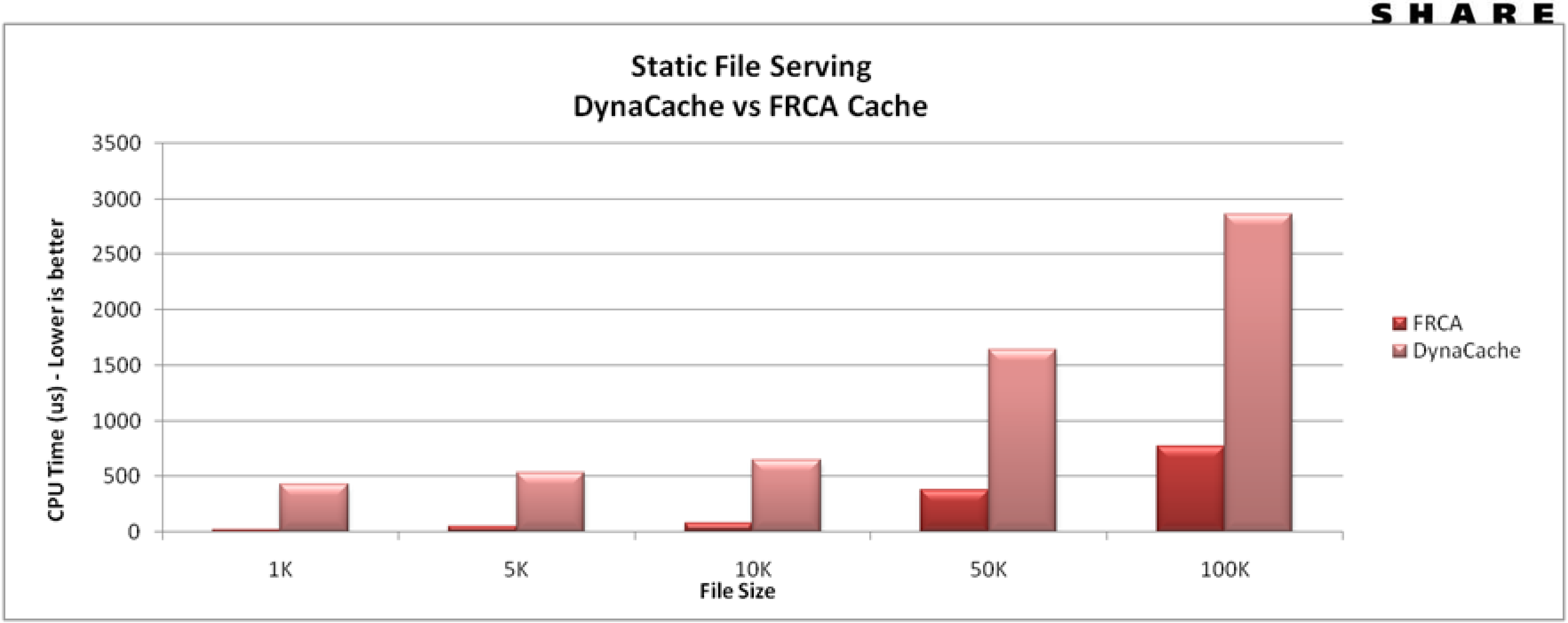
- We'll see this picture later when we discuss high availability
- WebSphere Application Server is a transaction manager … it is able to initiate a transaction and have other resource managers (DB2, CICS, IMS) participate in the unit of work
- For two phase commit processing, *someone* has to play the role of syncpoint coordinator
- On z/OS and Parallel Sysplex that someone is RRS, which uses Coupling Facility data structures and patented recovery algorithms to provide very efficient failed transaction recovery
- WAS z/OS registers with RRS, as do resource managers. RRS handles the two-phase commit coordination

Another case of "below the specification line" exploitation of existing z/OS and Parallel Sysplex technology to perform a task in an optimized manner for the platform

Exploitation of TCP/IP

- Controller Region uses z/OS TCP/IP stack's Asynchronous I/O Capability
  - TCP/IP schedules an SRB into the Controller's address space when data is available
  - The SRB queues work to a pool of WebSphere threads
  - This avoids the need to have one thread per connected client
  - In 2007 we ran a test with **120,000** clients connected and actively using one controller with five servant regions.
- Exploit the Fast Response Cache Accelerator (FRCA) capability of the z/OS TCP/IP stack
  - vastly improves response time for both static and cacheable dynamic content

z/OS: FRCA Performance with Simple File Serving Application



- The amount of CPU time needed to process a request is dramatically reduced using FRCA as compared to Dyna Caching
  - Dyna Cache is 14 times more costly than FRCA caching for 1k file sizes
  - Larger file sizes, 5k to 100k, Dyna Cache is 11x to 4x more costly

System Configuration
Workload: Simple File Server App
SUT: IBM z10 Processor (model 2097 – 720) 4 x 4.4 GHz, 32 GB Real
Driver:

Exploitation of GRS

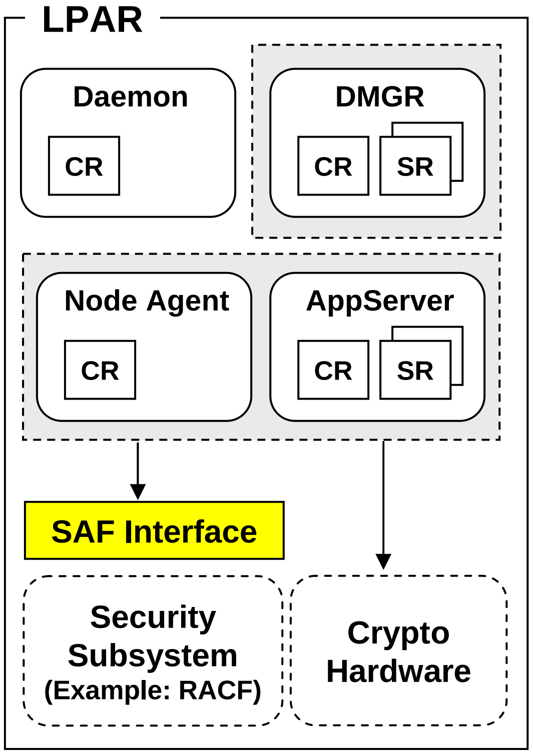
- GRS ENQs are used to manage session affinity to a servant region
    - Controller uses GQSCAN to find the right servant
- ENQs are also used to manage transaction affinity to a server in a cluster
    - IIOP requests may be forwarded to keep a transaction in one server or just on the same LPAR
- ENQs are also managed to interact with the Location Service Agent in the Daemon

Exploitation of XCF

- XCF Group Services are used by the High Availability Manager (HA Manager)
- Group notification exit informs WAS when another member of the group starts or terminates
- This enables HA Manager to react to server failures
- Use of XCF eliminates the need for TCP/IP pinging to determine the state of other servers
- XCF also avoids the ‘partition’ problem possible with a network solution

Exploitation of SAF and Crypto

SAF is a security interface; Crypto is a hardware-assist processor for encryption and key storage on the System z and z/OS platform



SAF Security Subsystem

- Sysplex-wide integrated security repository
- Single location for security artifacts rather than scattered model
- IDs, groups, keyrings, certificates, EJB role enforcement
- Local access … unlike LDAP, do not need to traverse network
- Extremely robust security model

Crypto Hardware

- Hardware-assisted cryptographic encryption and de-encryption
- Extremely secure private key store management

Properly configured, z/OS provides an extremely secure environment … many say the most secure available

Exploitation of z/OS Logstreams

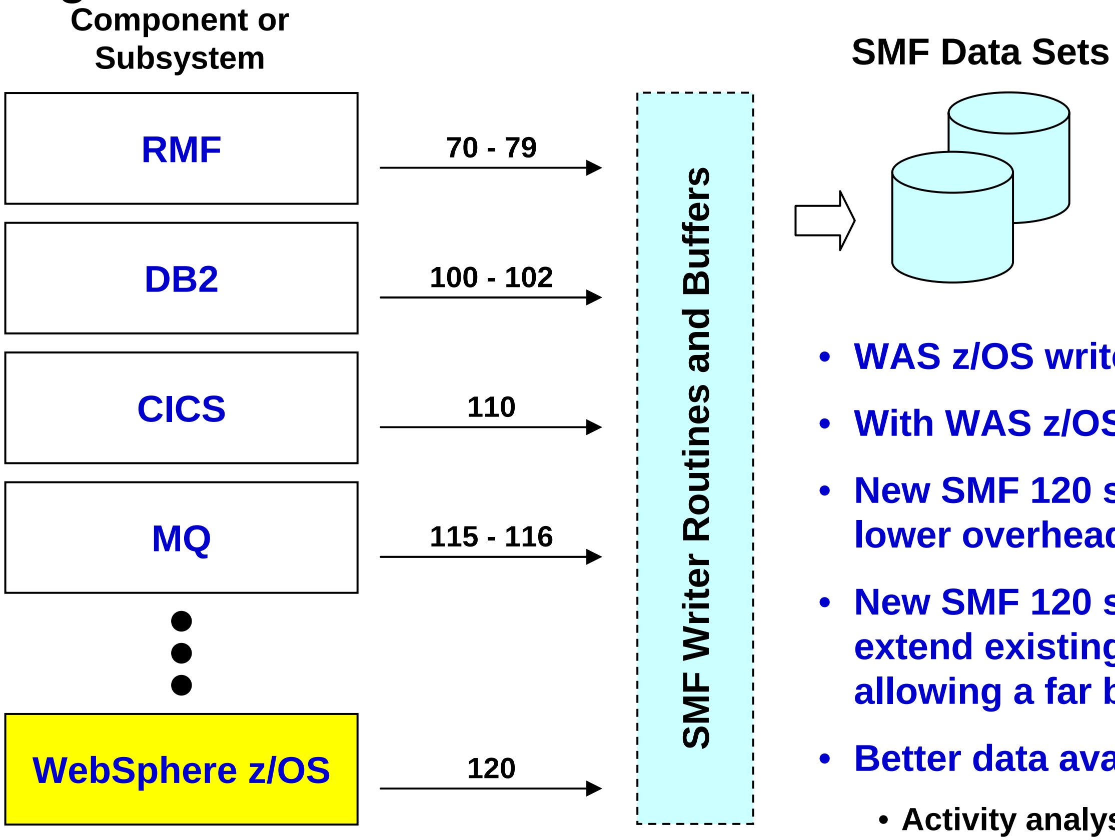
- A logstream is essentially a file, but
  - Shareable across the sysplex
  - Archiving and retention are policy driven
- WebSphere uses logstreams for:
  - A log of messages not suitable as WTOs
  - Provides a hardened log of in-flight transactions

Exploitation of z/OS Dataspaces

- Before 64 bit, Dataspaces provided extra room
  - Each one up to an extra 2GB of space
  - Can be shared by more than one process
- WebSphere uses dataspaces for:
  - Used to hold local-comm messages for cross process communication
  - Used to park CRA<->SR JFAP messages
  - Used to hold intermediate data for SMF 120 records
  - Used by dynacache for cross servant session data recovery
  - Used by XA transaction logging

Exploitation of SMF

SMF is an activity recording facility of z/OS that allows subsystems to record key activity for analysis, management and accounting chargeback



- WAS z/OS writes SMF 120 records
- With WAS z/OS V7, a new subtype was created: 9
- New SMF 120 subtype 9 provides better data with lower overhead cost
- New SMF 120 subtype 9 records complement and extend existing SMF from other subsystems, allowing a far better picture of what's going on
- Better data available for …
  - Activity analysis
  - Usage statistics
  - Accounting chargeback

`http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101342`

Exploitation of ENF

- ENF is the z/OS Event Notification Facility
- WebSphere generates ENF signals to allow the Daemons in the same cell and sysplex to share information about server status
- WebSphere also listens for z/OS generated events:
  - LPAR down indicates all servers on that LPAR are gone
  - USS terminating causes a shutdown of all the WAS servers on the LPAR

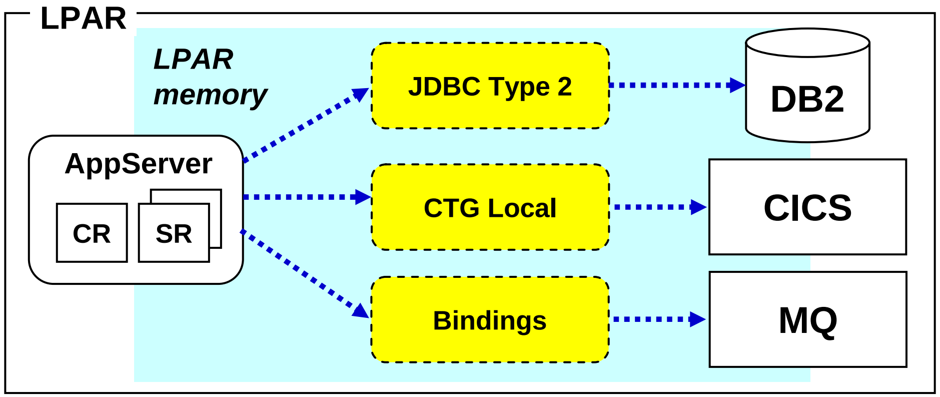
Exploitation of USS

- USS is Unix System Services
- Obviously WebSphere uses the USS File System to hold our configuration and .jar and .dll files
- Assembler services are also used to load modules from the file system into Common Storage (LPA).
- USS Services are also used to access what would normally be ‘C’ or Java APIs from assembler
    - gethostbyname, TCP/IP Read/Write, getenv, etc.

Exploitation of Cross-Memory Communications

Any time client and target are in the same LPAR, there's an opportunity for cross-memory exploitation. Let's look at a few examples:

Data Access

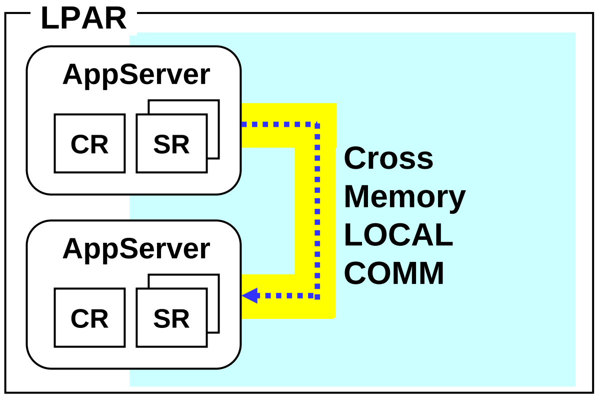


Benefits:

- Cross memory speed
- Security ID propagation (no alias)
- Exploitation of RRS
- Avoid serialization of parameters
- Avoids SSL overhead
- Single thread of execution

LOCAL COMM

Used for IIOP flows between servers on the same LPAR.



Benefits:

- Avoids IP stack entirely
- Avoids SSL overhead
- Very fast, very secure

Extension to Local Comm: new Optimized Local Adapters …

SHARE in Anaheim 2011

Exploitation of z/OS Storage Management

- Authorized key storage used in the servant region to prevent application code from damaging WebSphere control structures
- Shared above-the-bar storage used by WOLA
- Storage cell pool services (especially in common storage) to improve performance

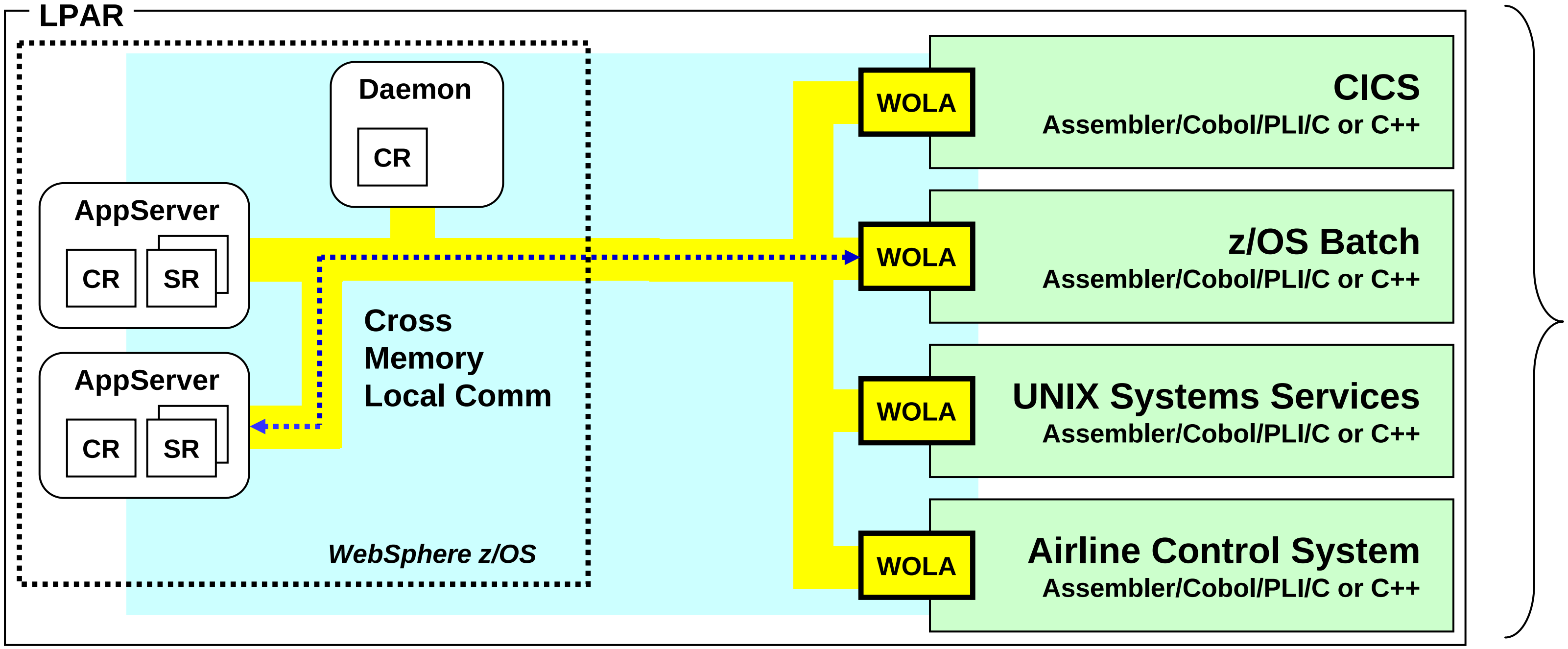
z/OS Servicability

- SVCDumps for failures and diagnosing problems
- IPCS formatters to analyze and format data
- Tracing to CTRACE external writer or in buffers in the dump
    - So trace doesn’t occupy file space or spool and is present with other data structures captured in the dump

WebSphere Application Server on z/OS Exclusive Features

Cross-Memory: Optimized Local Adapters

This function allows external address spaces to participate in a cell's Local Comm communications.



External address spaces supported in 7.0.0.4 release.

IMS added in 7.0.0.12

Benefits:

- Based on Local Comm (z/OS exclusive)
- Bi-directional … WAS outbound or inbound to WAS (WOLA exclusive)
- CICS Security and Transaction propagation (some restrictions apply)
- Faster than other local solutions

WP101490 on
`ibm.com/support/techdocs`
for more

Timeouts

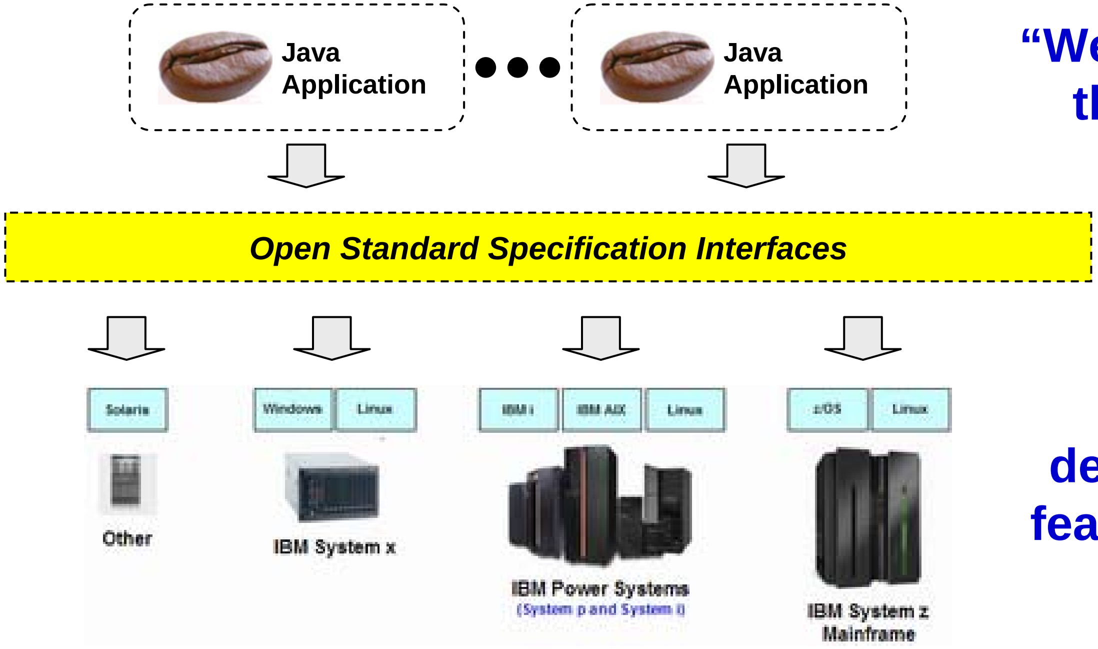
- Hung or deadlocked application requests can be shaken loose or abended
- Callstack or other debug documentation automatically collected
- CPU timeout to break out of endless loops
- “Grace period” before the abend to let innocent requests in the same servant complete
- Timeout on the queue work that has no chance of completing in time
- ‘Save Last Servant’ to not abend the last servant if a recurring problem is killing them all
- Auto-pause the server (close ports) if all/only servant(s) die

Pause/Resume Listeners

- Modify command causes the server to stop taking work
  - HTTP listener ports are closed
  - MQ Message Listener Ports stop taking messages
  - IIOP Location Service Agent stops directing locate requests to this server
- Server is still visible to admin
- In-flight requests/transactions complete
- Caches remain intact, JIT-ed code stays
- Resume Listeners Modify command reverses the pause
  - Much faster than starting a fresh server

WebSphere is WebSphere…but

This point can't be stressed enough -- the differentiation is *not* in the open standard specification support offered. *That is common across platforms!*



Java Application ●●● Java Application

Open Standard Specification Interfaces

Java EE, Java SE, EJB, Servlet, JSP, JDBC, JCA, JMS, Web Services, etc.

"WebSphere is WebSphere" above the specification interface line

How it's *implemented* is dependent on the platform … its features, functions, attributes and qualities of service