WAS z/OS Update

Steve Foley

IBM Software Group

steve_foley@uk.ibm.com

IBM Software Group

Important Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

2

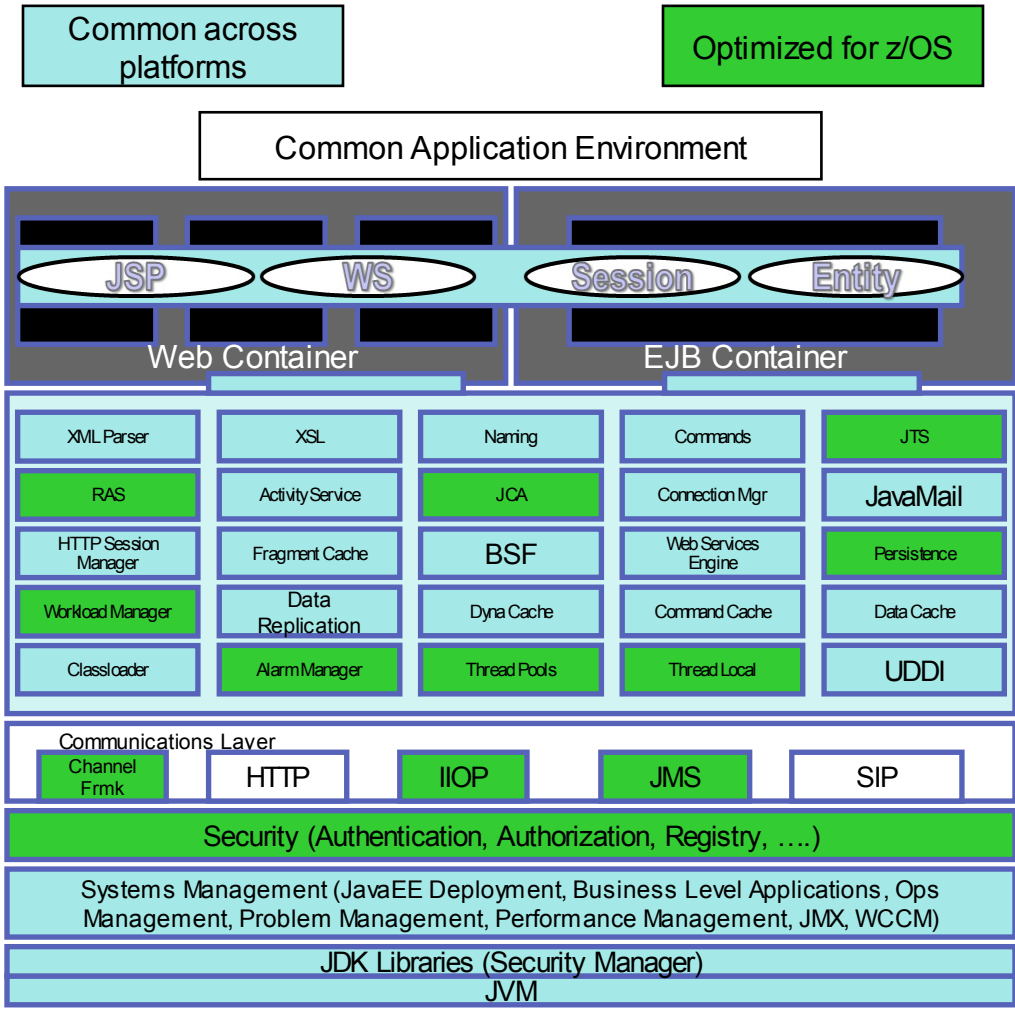
Agenda

- **WebSphere Application Server for z/OS V7.0**
  - WAS z/OS V7.0 exclusives
  - Performance
- **Value of Co-Location**
  - Enhanced JDBC Type 2 driver
  - Benchmark result

WebSphere Application Server for z/OS: Unique Elements

Integration with z/OS that maintains application transparency

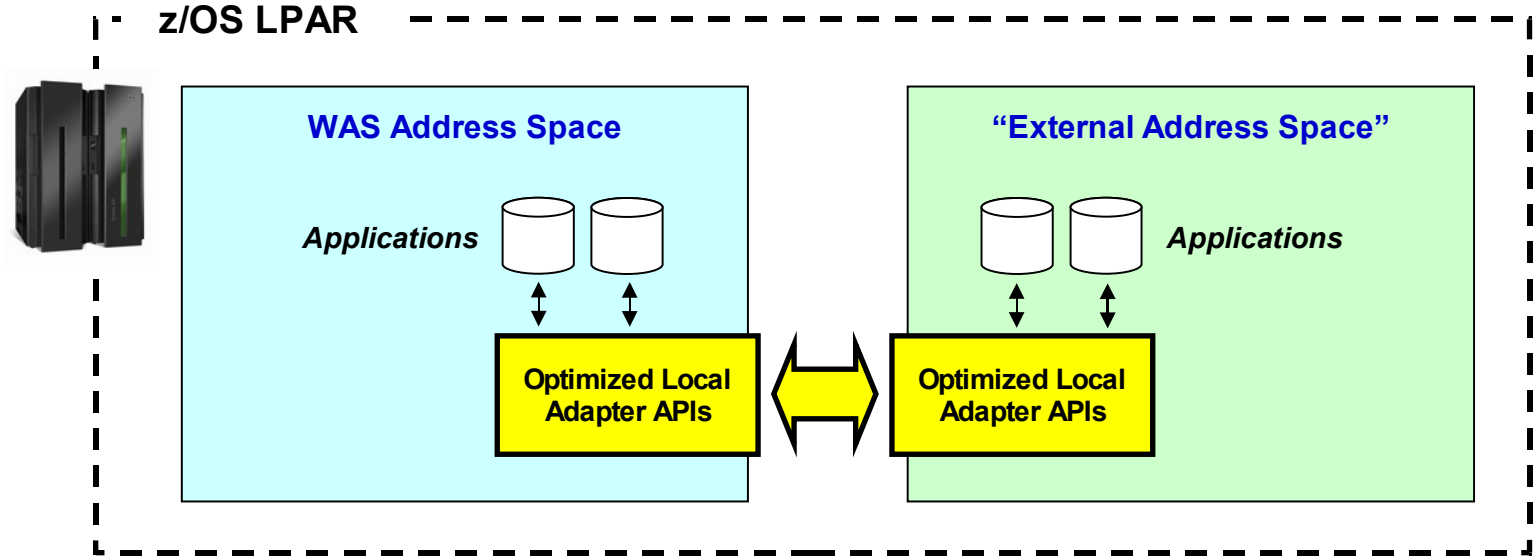
- **Server Architecture**
  - Control/Servant Region Split
- **Workload Management**
  - Leverages Workload Manager
- **Security**
  - Use of the Security Authorization Facility
- **Transaction Management**
  - Leverages Resource Recovery Services
- **Connectors**
  - Leverages available local (Type 2) connectors
- **Thread Management**
  - OS level threads for monitoring and control
- **Scalability**
  - Multiple Servant Region
- **Communications layer**
  - True Asynch I/O model
- **Recovery**
  - Leverages Automatic Restart Manager
- **Reporting**
  - System Management Facility



WOLA

WebSphere Optimized Local Adapters

WOLA is a high-speed cross-memory bi-directional communications mechanism made available in WAS z/OS 7.0.0.4



High-Speed -- because it is cross-memory

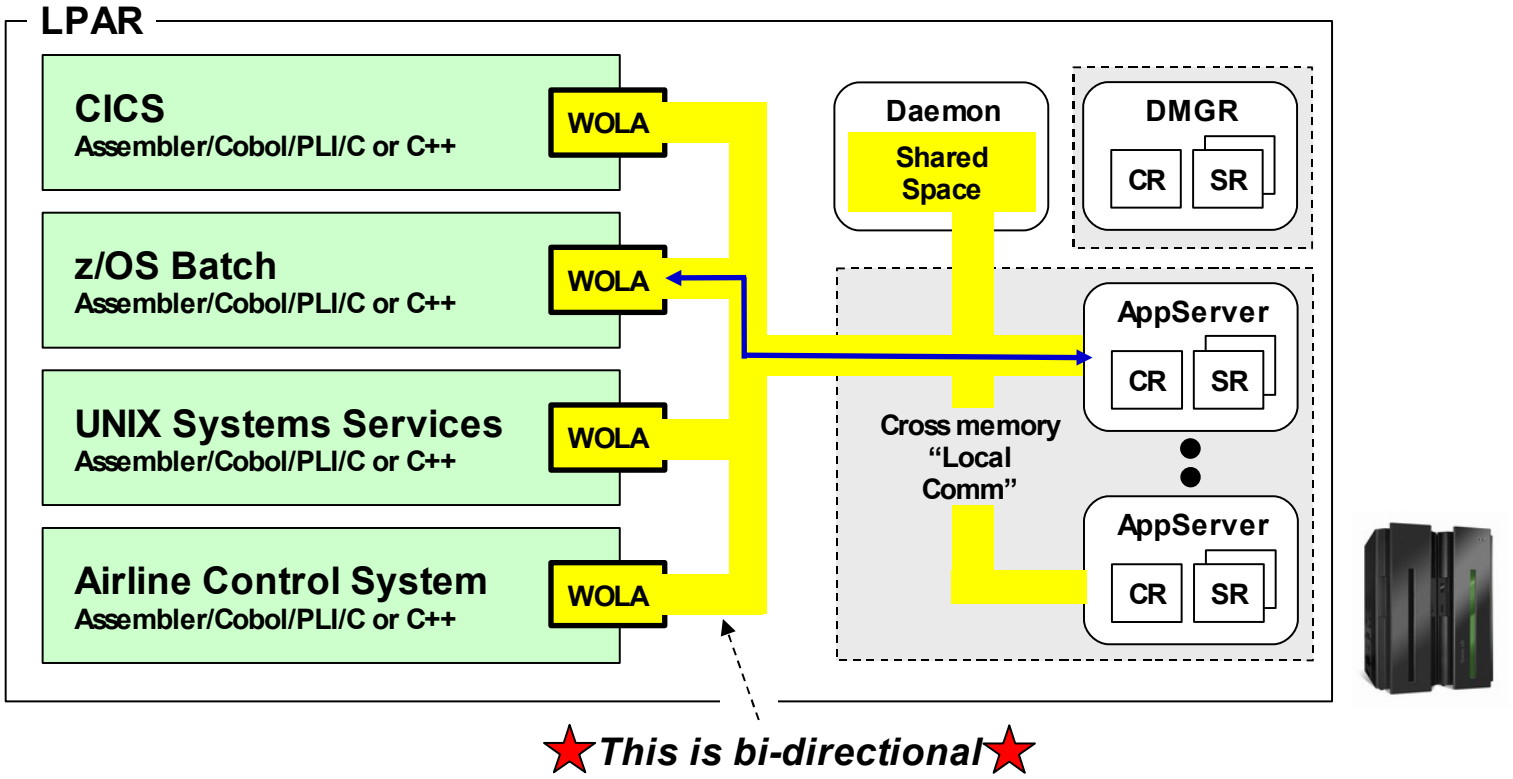
Cross-Memory -- avoids network latency and serialization latency

Bi-Directional -- from WAS outbound and inbound to WAS

Exploit WAS EJB assets from batch programs or CICS

So … What Is WOLA?

WOLA is a new cross-memory communication structure for WAS V7. It's an *extension* of a WAS "Local Comm" used inside of WAS:

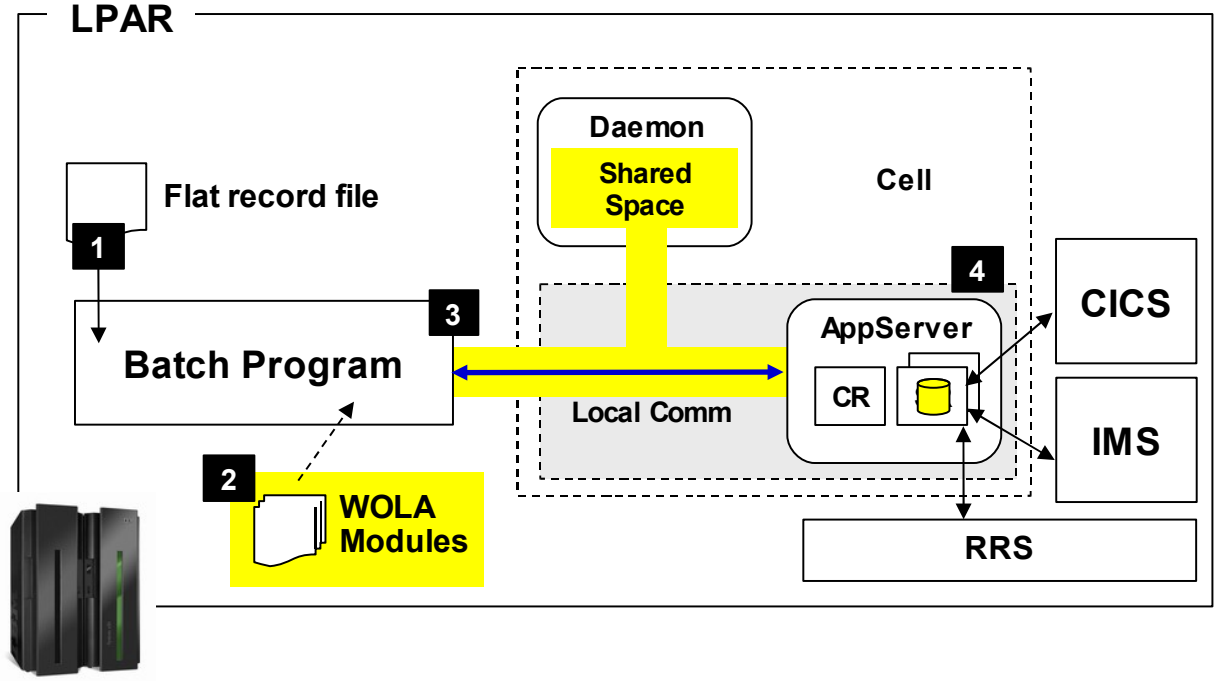


This extension is implemented with a new set of modules that provide an API for external address to access servers using this Daemon shared space mechanism

High Level Picture …

A Simple Use-Case Scenario

Here's a picture … just to help us get our minds around this thing:



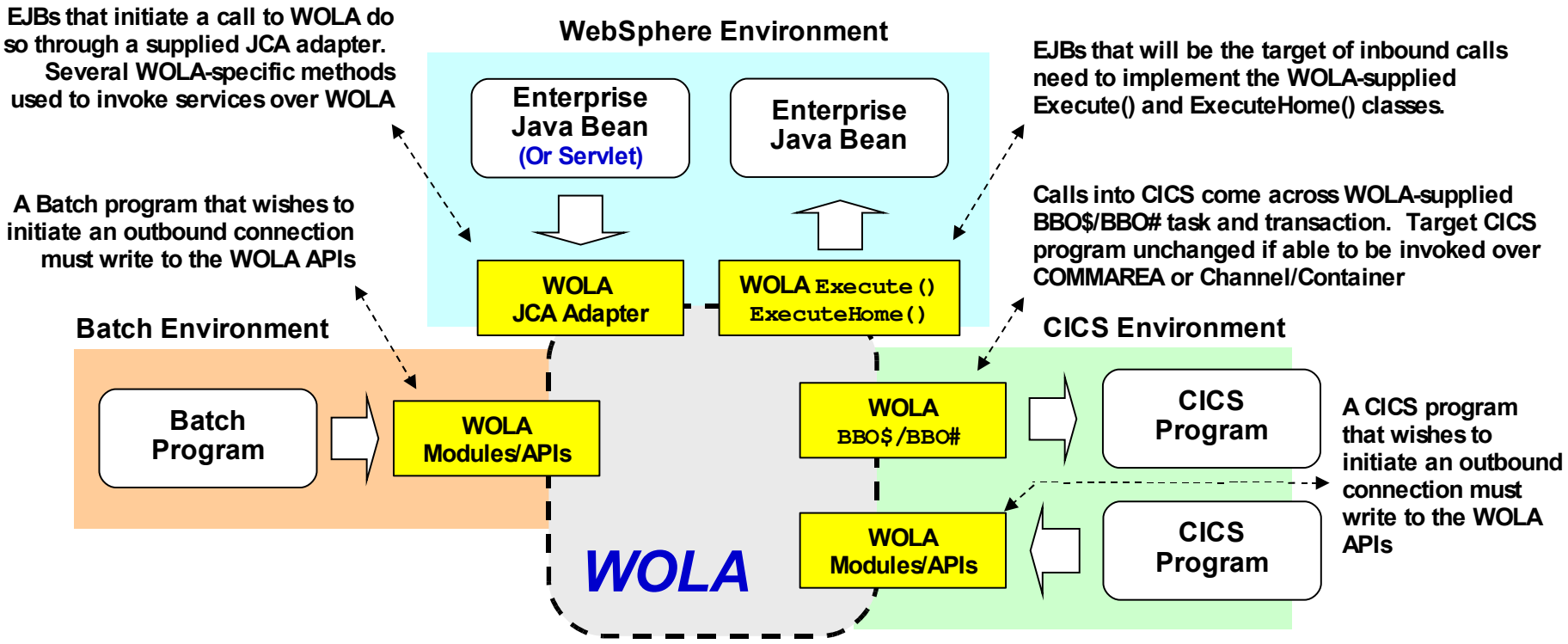
1. Flat record file serves as input to batch program
2. WAS 7.0.0.4 WOLA modules STEPLIBed to from batch JCL
3. Batch program uses WOLA APIs to access WAS and invoke EJB
4. EJB initiates transaction and updates CICS, IMS with two-phase commit using standard WAS data connector architectures

This illustrates a relatively simple -- but likely common -- usage: batch file using an existing transactional EJB to update data.

Five perspectives …

WOLA Interface -- Perspective from Five Angles

This conceptual picture offers insight into the interface requirements:



WAS z/OS 7.0.0.4 supplies the WOLA artifacts. We'll see how to enable the environment later.

You make modules/classes available: `STEPLIB`, `DFHRPL`, `ola.rar` **and** `ola_apis.jar`

Batch CICS WAS Development Tool

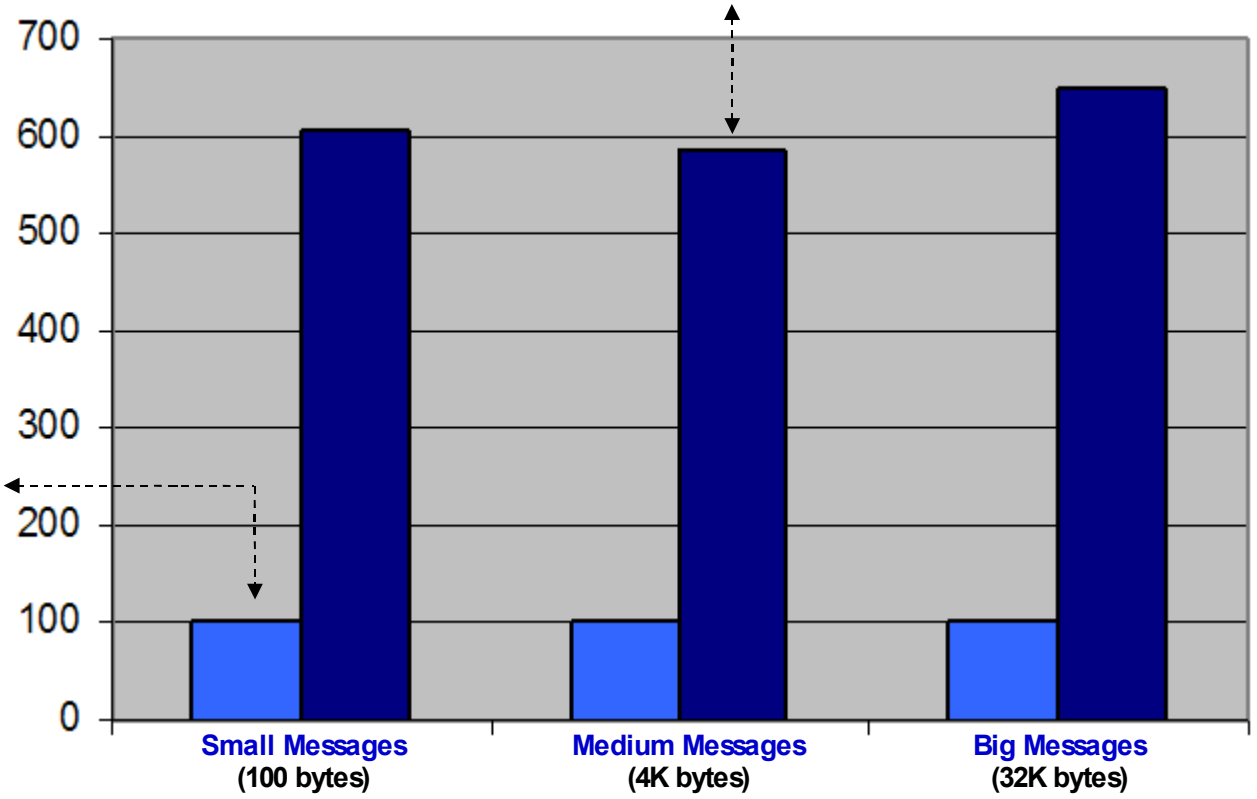
Reducing impact …

Larger Messages

We see WOLA outperforming web services across a range of message sizes:

WOLA

Relative throughput … relative to the normalized web services baseline



Web Services

Normalized so it represents the baseline "100"

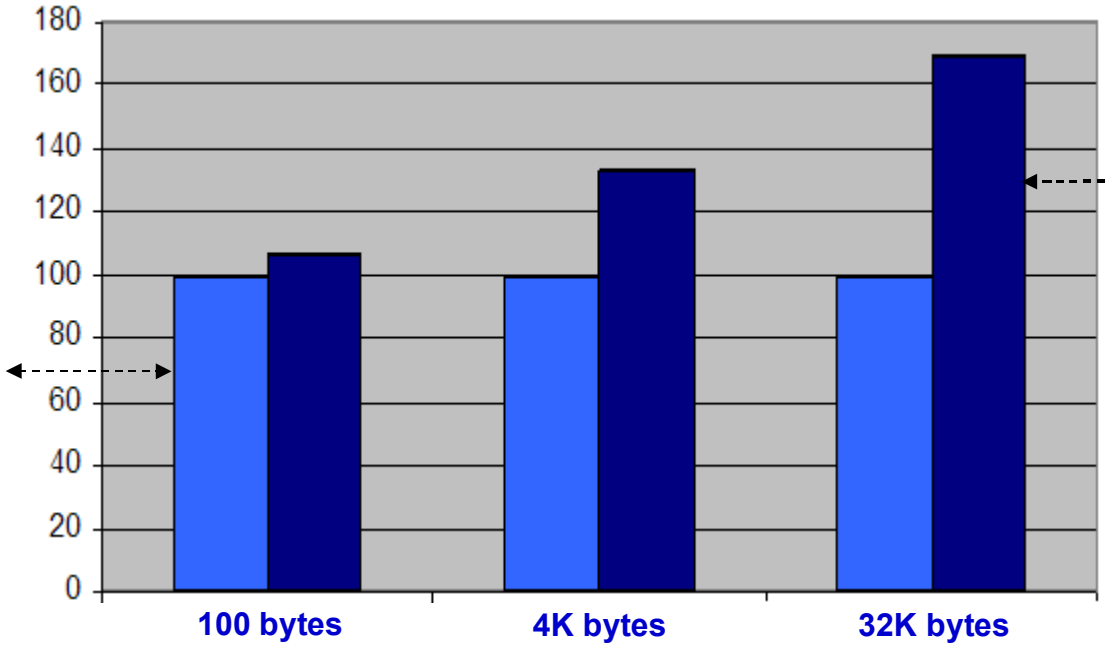
EXCI …

WAS ⇨ CICS Transaction Gateway - Local EXCI

CTG EXCI is itself a cross-memory technology … yet we still see a delta in favor of OLA. Not as much as with web services, but it's still there:

Again … relative throughput, normalized

In this case, CTG EXCI is the constant reference; OLA is adjusted proportionally



CTG EXCI

Normalized so it represents the baseline "100"

WOLA

Relative throughput … relative to the normalized CTG EXCI baseline

We see WOLA improving its advantage as the message size gets bigger and reaches the limit of EXCI, with is 32K

We have some careful positioning information coming!

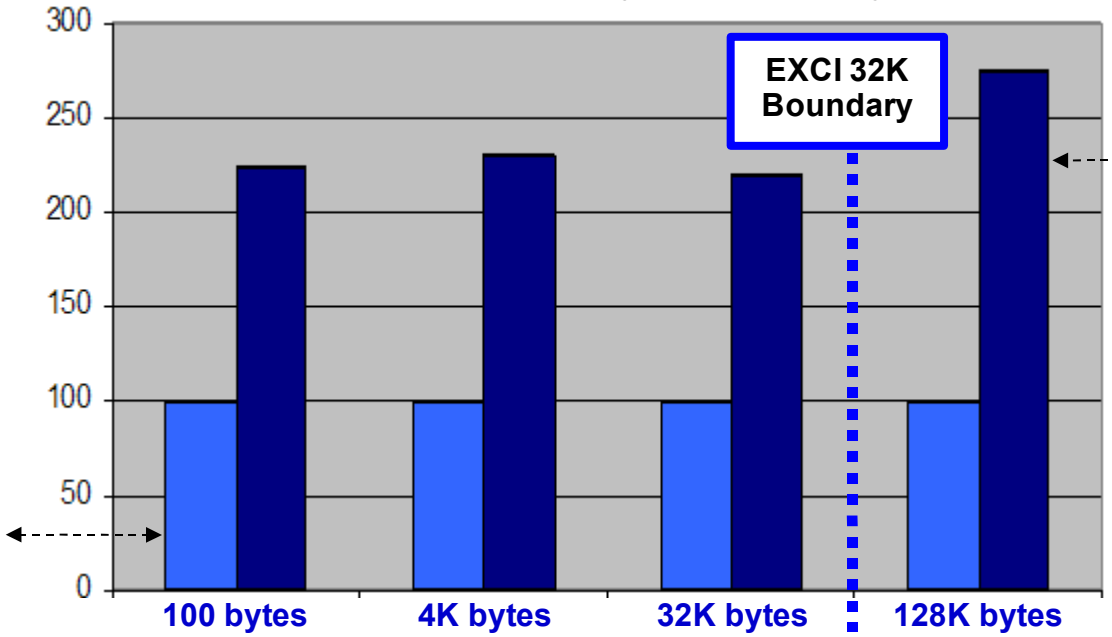
IPIC …

WAS ⇨ CICS Using IPIC of CICS TS3.2

IPIC supports channels and containers and allows us to get back the 32K limit of EXCI. How does WOLA compare to that?

Relative throughput, normalized

CTG IPIC is the constant reference; OLA is adjusted proportionally



IPIC

Normalized so it represents the baseline "100"

WOLA

Relative throughput … relative to the normalized IPIC baseline

WOLA is a very good large message local transfer mechanism

We have some careful positioning information coming!

Positioning …

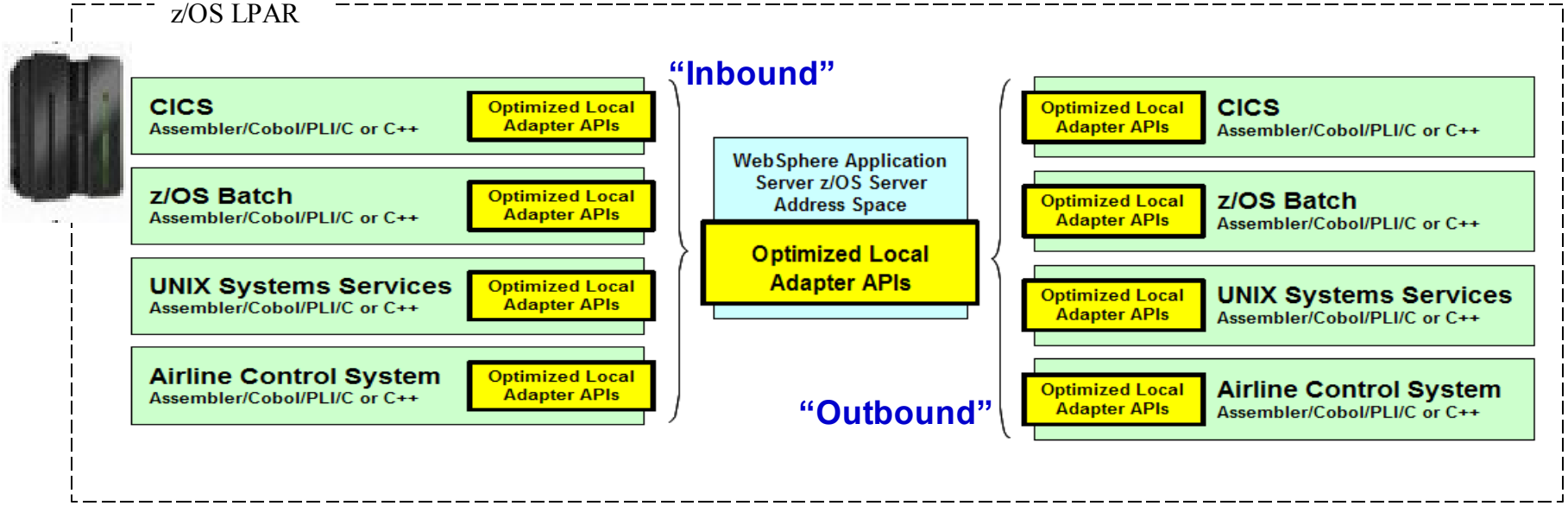
Relative Advantages

Of WOLA and CICS Transaction Gateway

	Relative Advantage Favors … WOLA	CTG
Bi-directional … WAS➔CICS and CICS➔WAS WOLA is bi-directional, CTG is only WAS➔CICS	●	
Part of the WebSphere Application Server z/OS Product WOLA shipped with 7.0.0.4, CTG is a separate FMID	●	
Able to be used for local *or* remote access to CICS WOLA is a local technology only, CTG supports both local EXCI as well as TCP-based remote access		●
Two-Phase Commit WAS➔CICS WOLA is limited at present ot synconreturn only. In future WOLA maintenance that support will be upgraded to 1-phase. 2-phase limited by design of CICS TRUE, which is what WOLA's support in CICS uses		●
Two-Phase Commit CICS➔WAS CTG can not be used for CICS➔WAS. WOLA able to propagate TX CICS➔WAS with full 2-phase commit support using RRS for syncpoint coordination.	●	
Flexible use of CICS channels and containers WOLA restricts container usage to one named channel only: IBM-WAS-ADAPTER. CTG supports multiple channels. WOLA uses indexedrecord while CTG uses mappedrecords. That means CTG supports the passing of multiple named containers on a channel while WOLA can not.		●

WOLA is a *complementary* technology with CTG. Both will have their place within an enterprise architecture.

WOLA - Summary

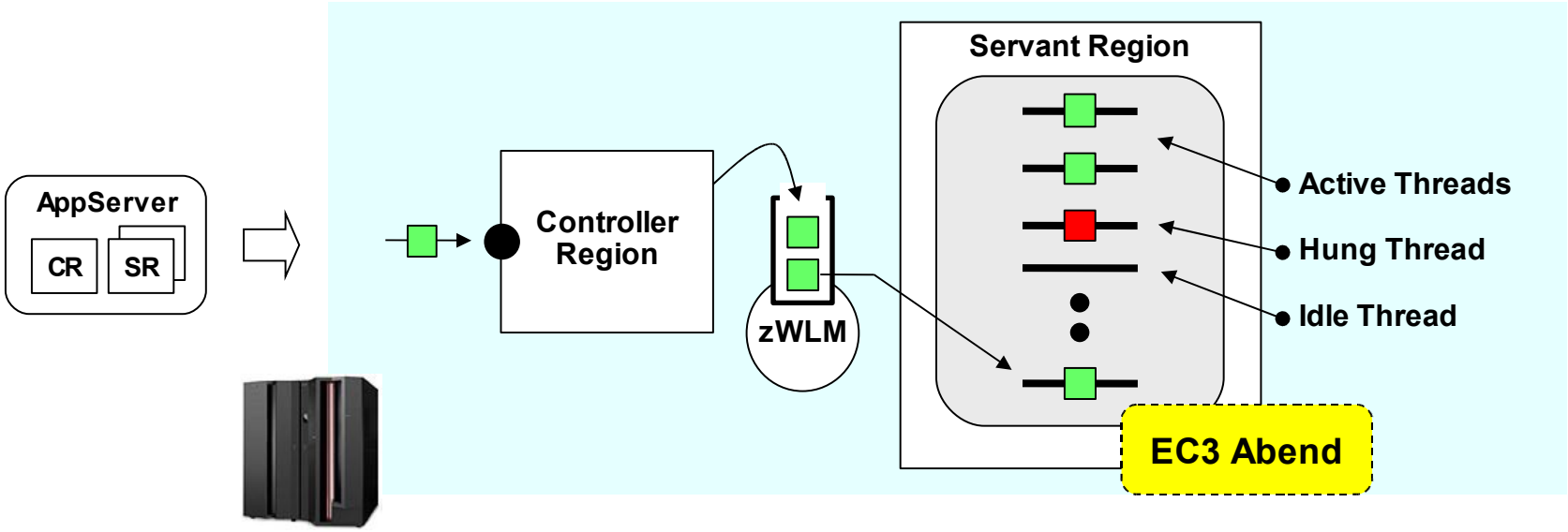


- "WebSphere Optimized Local Adapters" is a new method of cross-memory local communications between applications running on WebSphere Application Server for z/OS and those in external address spaces on the same system such as Customer Information and Control System (CICS), batch programs, Unix Systems Services (USS) programs, and Airlines Line Control System (ALCS) programs.
- Advantages of OLA:
  - Bi-directional with optimal performance for application patterns that involve fine-grained calling to/from WAS.
  - Security propagation.
  - Global Transactions support.
  - Workload Context propagation.

Thread Hang Recovery

Thread Hang Recovery

Before V7, when a thread was marked as "hung" (usually due to a timer having expired), WebSphere had no choice but to abend the servant region:



WebSphere would abend the servant. WLM would restart.

All other in-flight work in that servant region is lost

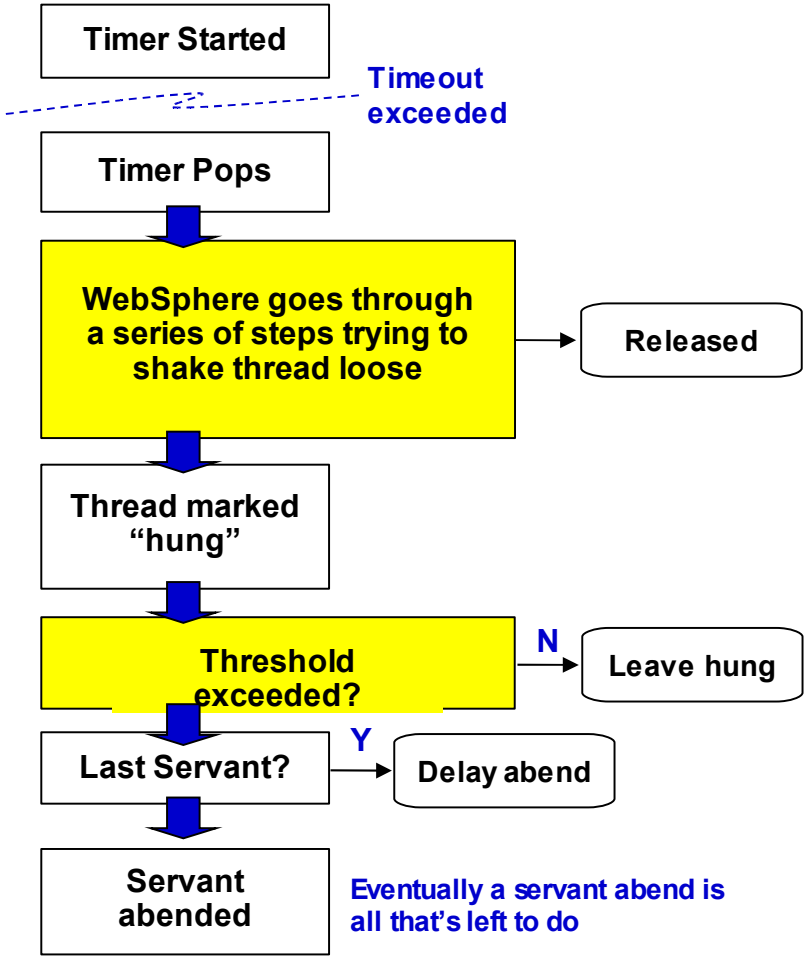
Multi-servant architecture protects against *total* outage*, but you recycle of servant still takes time, cycles and the work in-flight is lost

* A WebSphere z/OS exclusive

Simplified illustration of processing …

Simplified Flow of Processing

A feature of WebSphere z/OS Version 7.0 that provides a way to try to interrupt a thread ("shake it loose") … and if not, then it may abend the servant:



Algorithm is a bit more complex than this, but this shows the essense of it

It makes use of the Java 5 Interruptible Thread APIs as well as a new internal registery of interruption objects

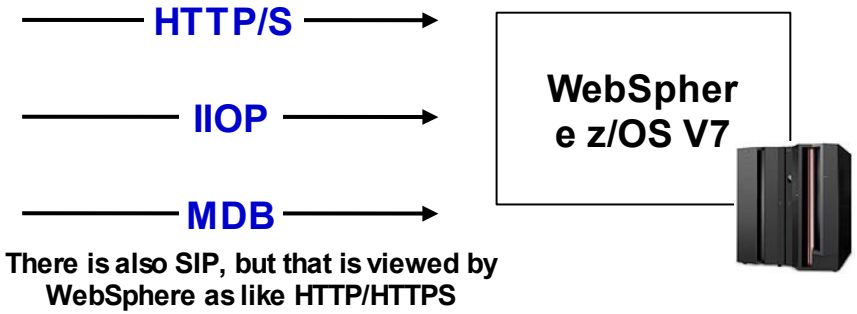
When attempts to unblock the thread are exhausted, WebSphere checks new "threshold" value to see if the that has been exceeded

If it runs out of options, it gathers information and moves to abend servant. You decide what output:

`none`, `svcdump`, `javacore`, `heapdump`, `traceback`

"Hang" is Determined By Timer Value, Based on Request Access

A "hung" thread is determined by a timer value "popping." Those timer values are based on the method of access the request came into WebSphere z/OS



There is also SIP, but that is viewed by WebSphere as like HTTP/HTTPS

Understanding how the requests are viewed by WebSphere will help you code your timeouts and Thread Hang Recovery variables more accurately

V7 InfoCenter Links for Timeout Value Settings

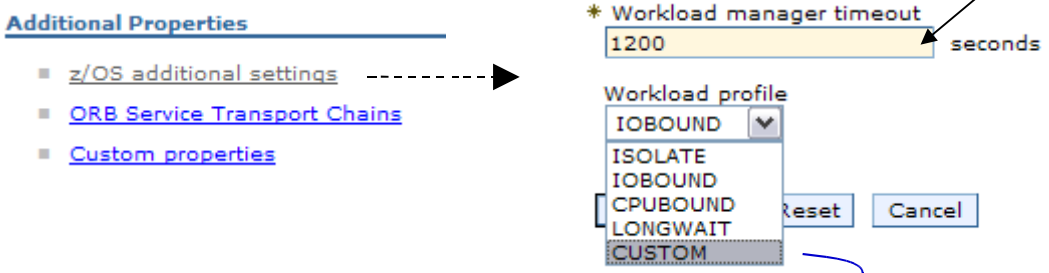
HTTP/S	ConnectionResponseTimeout Default: 300 seconds http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic /com.ibm.websphere.zseries.doc/info/zseries/ae/xrun_transport.html
IIOP	Workload Manager Timeout Default: 1200 seconds http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic /com.ibm.websphere.zseries.doc/info/zseries/ae/rtrb_controllingtimeout.html
MDB	control_region_mdb_request_timeout Default: 120 seconds http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic /com.ibm.websphere.zseries.doc/info/zseries/ae/rtrb_controllingtimeout.html

Threads and threshold …

Threads and Threshold -- Delaying Servant Recycle

In V7 the number of threads in a servant region is customizable. You may also set a *percentange of threads* that are marked "hung" before servant recycle:

Application servers ⇨ smsr01a ⇨ ORB service



FYI …This is the IIOP timeout setting

ISOLATE	1
IOBOUND	based on CPUs
CPUBOUND	based on CPUs
LONGWAIT	40
CUSTOM	*you set number*

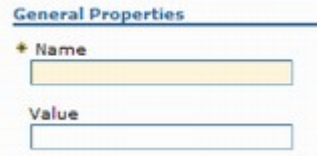
New in V7

Search on "uorb_rorb_service.html" in InfoCenter

`servant_region_custom_thread_count`

`server_region_stalled_thread_threshold_percent`

Both are normal WebSphere variables. You set at the "scope" you desire (cell, node, server or cluster)



Verify in *controller's* output (JESMSGLG):

```
BBOO0234I SERVANT PROCESS THREAD COUNT IS xx
```

10	Idle
9	Working
8	Working
7	Working
6	Working
5	Working
4	Working
3	Stalled
2	Stalled
1	Stalled

Threshold=50

In this example the servant would stay up, even though three of its threads where marked hung

Objective is to delay servant recycle. Threads marked hung may recover on their own, so delaying avoids servant recycle, outage and CPU consumed restarting

New DISPLAY Command

In V7 they added a new modify command so you can see the status of the *active* threads in a server (idle threads not displayed)

```
F <server>,DISPLAY,THREADS,ALL
F <server>,DISPLAY,THREADS,TIMEDOUT
F <server>,DISPLAY,THREADS,REQUEST=value
F <server>,DISPLAY,THREADS,ASID=value
F <server>,DISPLAY,THREADS,AGE=value
```

- Show all threads
- Show only timed out threads
- Display on a specific request
- Display on a specific ASID (useful when multiple servants)
- Display threads over a certain time since request

Server short name … command issued against controller, not servant

,SUMMARY or ,DETAILS may be added to any of these; default is ,SUMMARY

```
F <server>,DISPLAY,THREADS,ALL
```

One servant example:

```
BBOJ0111I:  REQUEST ASID JW TO RE DISPATCH TIME
BBOJ0112I: ffffb35f 0176  N  N  N 2008/03/24 22:39:58.423648
BBOJ0112I: ffffb360 0176  Y  N  N 2008/03/24 22:39:56.569834
BBOJ0112I: ffffb361 0176  Y  Y  N 2008/03/24 22:39:53.790695
BBOJ0112I: ffffb362 0176  Y  Y  Y 2008/03/24 22:39:50.720461
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,THREADS,ALL
```

- Healthy thread
- In a Java Wait but not yet Timed Out
- Java Wait and Timed Out, but ITI has not yet marked it “hung”
- JW, TO and retried -- Hung

In this example Threshold > 0 otherwise servant would be recycled because one thread is hung

New variables …

Changing the number of servants

You can define the minimum and maximum number of servants in the admin console

To change it, you had to recycle the server

If you find you need more, shutting down the whole server is probably the last thing you wanted to do

In V7 we introduced two new Modify commands

```
MODIFY server,WLM_MIN_MAX=(min,max)
MODIFY server,DISPLAY,WLM
```

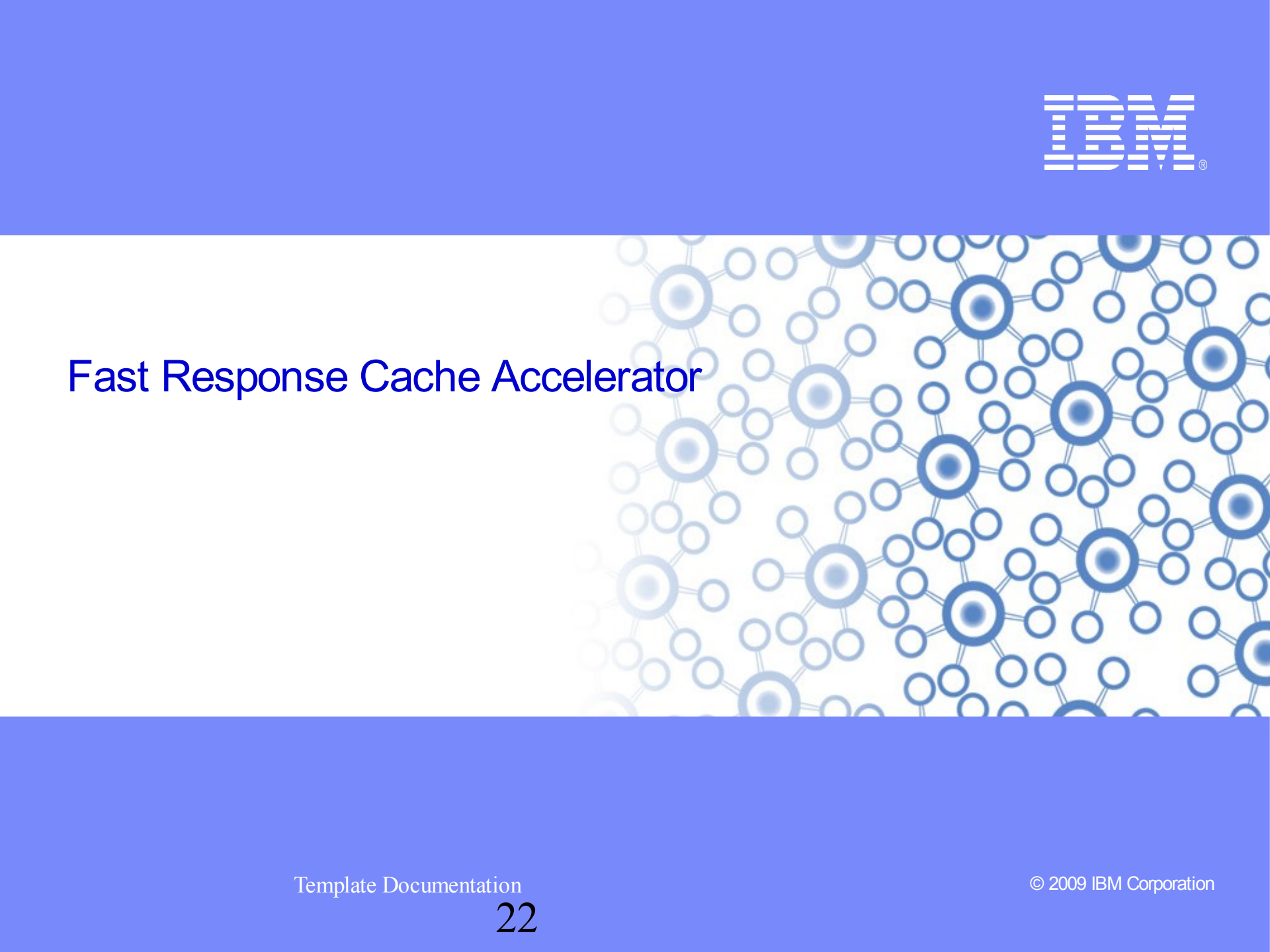
This temporarily (until the server restarts) changes the min and max values

If the current number of servants is lower than the new MIN value, WLM will **usually** start more to catch up

If the current number of servants is greater than the new MIN value, WLM will **probably eventually** shut down the extra servants… **maybe**.

If the current number of servants is at the old MAX value and you've increased it, WLM **might** start more

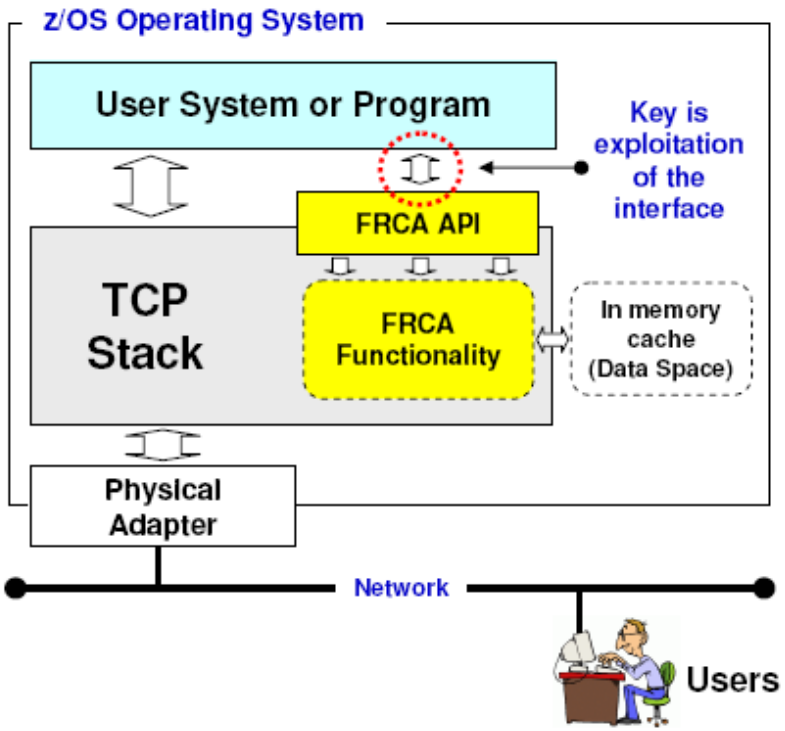
If the current number of servants is more than the new MAX value, WLM **may eventually** shut down the extras to get below the new MAX. **Maybe**.



Fast Response Cache Accelerator

FRCA - Fast Response Cache Accelerator

FRCA is a caching mechanism provided by TCP that allows for caching of content "down low" in the stack

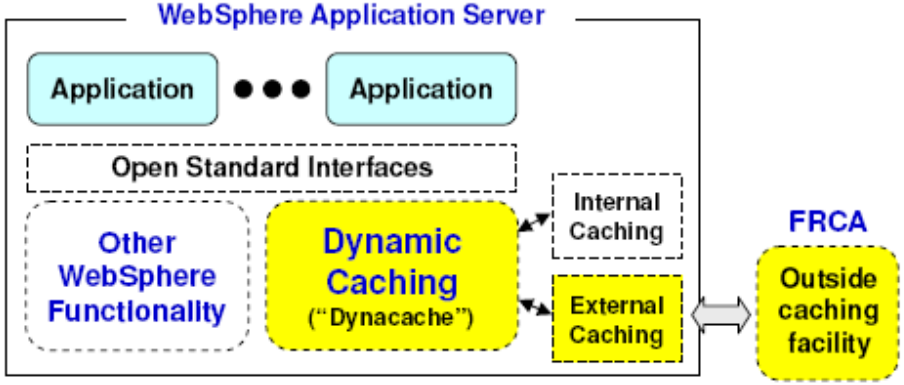


Important Points:

- FRCA is *not* new. It's been around for many years.
- FRCA itself is *not* a z/OS exclusive … it is present on other platforms.
- It is *not* "automatic" to everyone; it must be intentionally exploited. Not everything does. Old z/OS HTTP Server (Domino Go) did. New z/OS Apache does *not*. Before V7, WebSphere itself did *not*.
- Value of FRCA is that it provides very efficient low-level caching, which reduces overall cycles and provides increased scaling
  It's remarkably efficient … really, it is ☺
- *Be careful* -- the degree of exploitation is very much application dependent
  The greater the static content, or cachable dynamic content, the greater the value seen

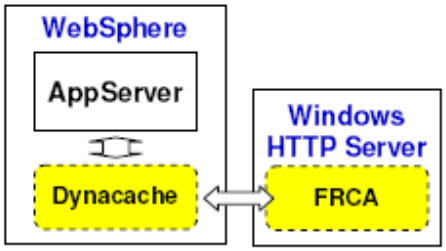
Version 7.0 FRCA Support

In V7.0, WebSphere z/OS is capable of using FRCA as an "external cache group" for it's "Dynacache" function *without* an HTTP Server involvement



- This picture is a *generic* WebSphere picture … applies to all platforms
- Key to understanding FRCA is that it's an "*external* cache group"

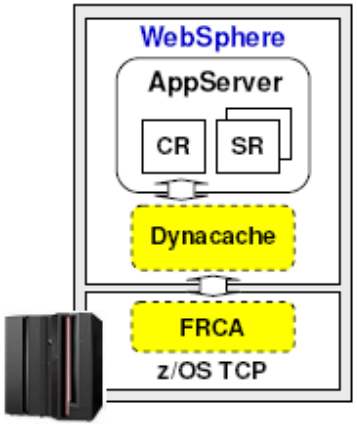
Distributed Platforms



Uses FRCA, but requires Windows HTTP Server

WebSphere z/OS can push to this as well

z/OS Platform



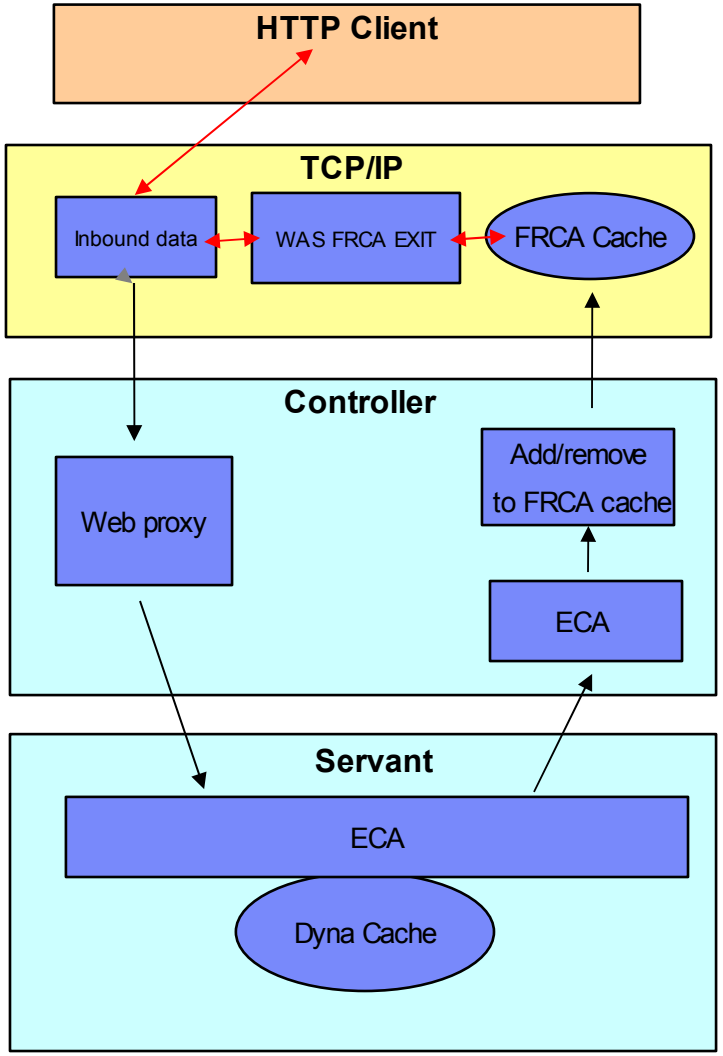
- No HTTP Server required
- HTTP Server *may* be part of larger architecture

Fast Response Cache Accelerator (FRCA)

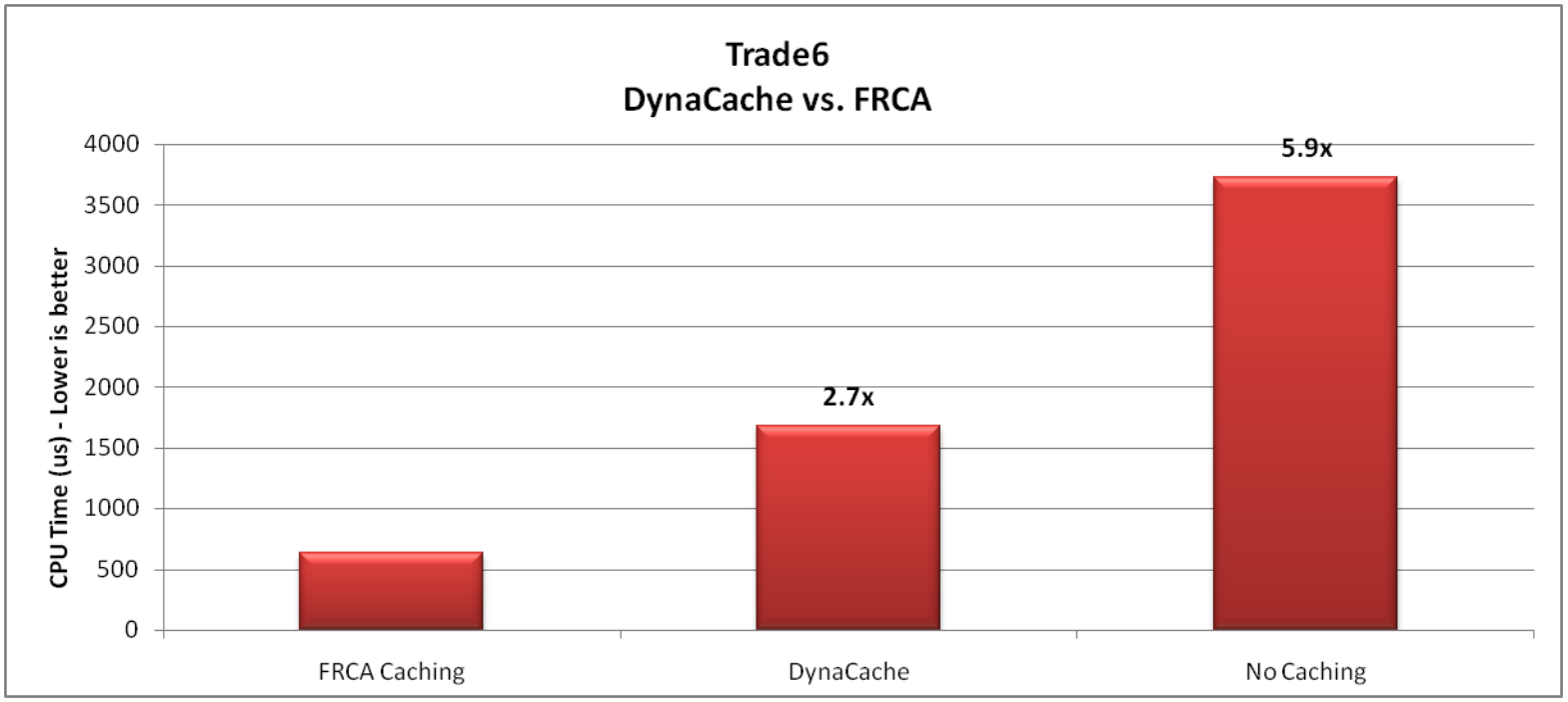
- **The FRCA cache is an HTTP cache that is maintained by TCP/IP**
- **Cached responses can be served with high performance using a minimal amount of CPU cycles**
- **Serve static requests from the FRCA cache**

  Provide equivalent performance on WAS as is possible with the FRCA cache on the web server
- **Serve dynamic content from the FRCA cache**

  Serve the same content that the Dynamic Cache serves but serve it from the FRCA cache
- **Record HTTP Access Log entries for requests served from the FRCA cache**



FRCA Performance with Trade6 EJB



- **Trade6 contains dynamic caching capabilities, for servlets and JSPs and is more customer-like than the simple file serving application.**
- Compared to FRCA, Dyna caching consumed 2.7x more CPU cycles
- Compared to no caching, FRCA was almost 6x more efficient

System Configuration
Workload: Trade6 EJB
SUT: IBM z10 Processor (model 2097 – 720) 4 x 4.4 GHz, 32 GB Real
Driver:

Summary of FRCA Support

Before:

- No ability to access FRCA from WebSphere z/OS
- Older HTTP Server (Domino Go) supported FRCA, but no way to push Dynacache to it

V7.0:

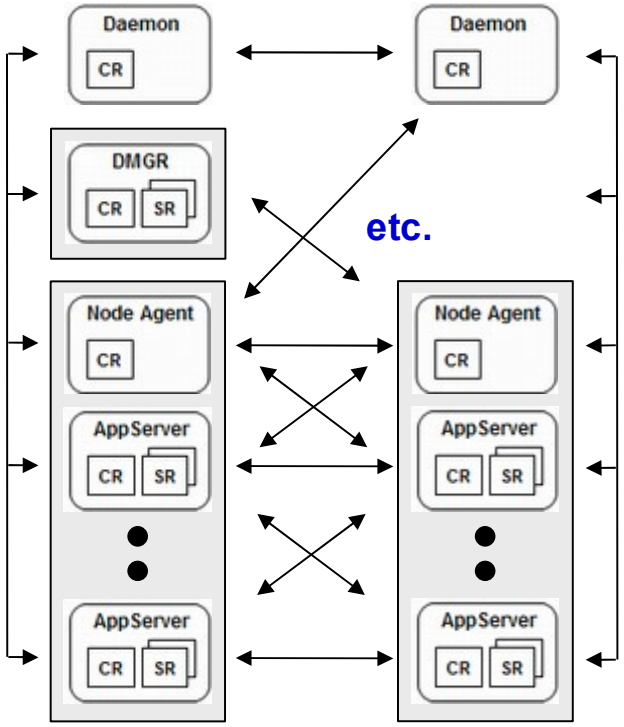
- WebSphere z/OS V7 Dynacache extended to support use of supplied adapter bean that works with z/OS TCP FRCA
- z/OS 1.9 or higher with, with 1.9 requiring ++APAR for TCP
  Function support rolled into z/OS 1.10
- FRCA is seen by WebSphere as an "external cache group," so all the other facilities of Dynacache are present
  Cache content timing out and being removed; making copies of cached content so all members of a cluster have access, etc. Complex topic beyond this workshop.
- You decide what is cached; defined in `cachespec.xml`
- Degree of utilization/exploitation is really a function of the application
  Applications with high static content, *or dynamic content where caching makes sense,* make good candidates for this new functionality

DCS signalling over XCF

XCF exploitation by High Availability Manager

Part of HAM and DCS, it involves heartbeats and communications between servers. It's a *mesh* architecture.

- HAM = "High Availability Manager" … DCS = "Distribution and Consistency Services"
- HAM and DCS are part of WebSphere Application Server on all platforms, not just z/OS



- **This is an N^2 issue -- the more servers, the more chatter, the more overhead**
- **The High Availability Manager (HAM) can be turned off, but you lose certain functionality**
  - For example, default messaging engine failover in a cluster
- **The default transport is TCP … but a Parallel Sysplex and z/OS has a more efficient signalling mechanism: XCF**

Therefore, this new function provides a way to define XCF as the transport rather than TCP

Summary of DCS/XCF Support

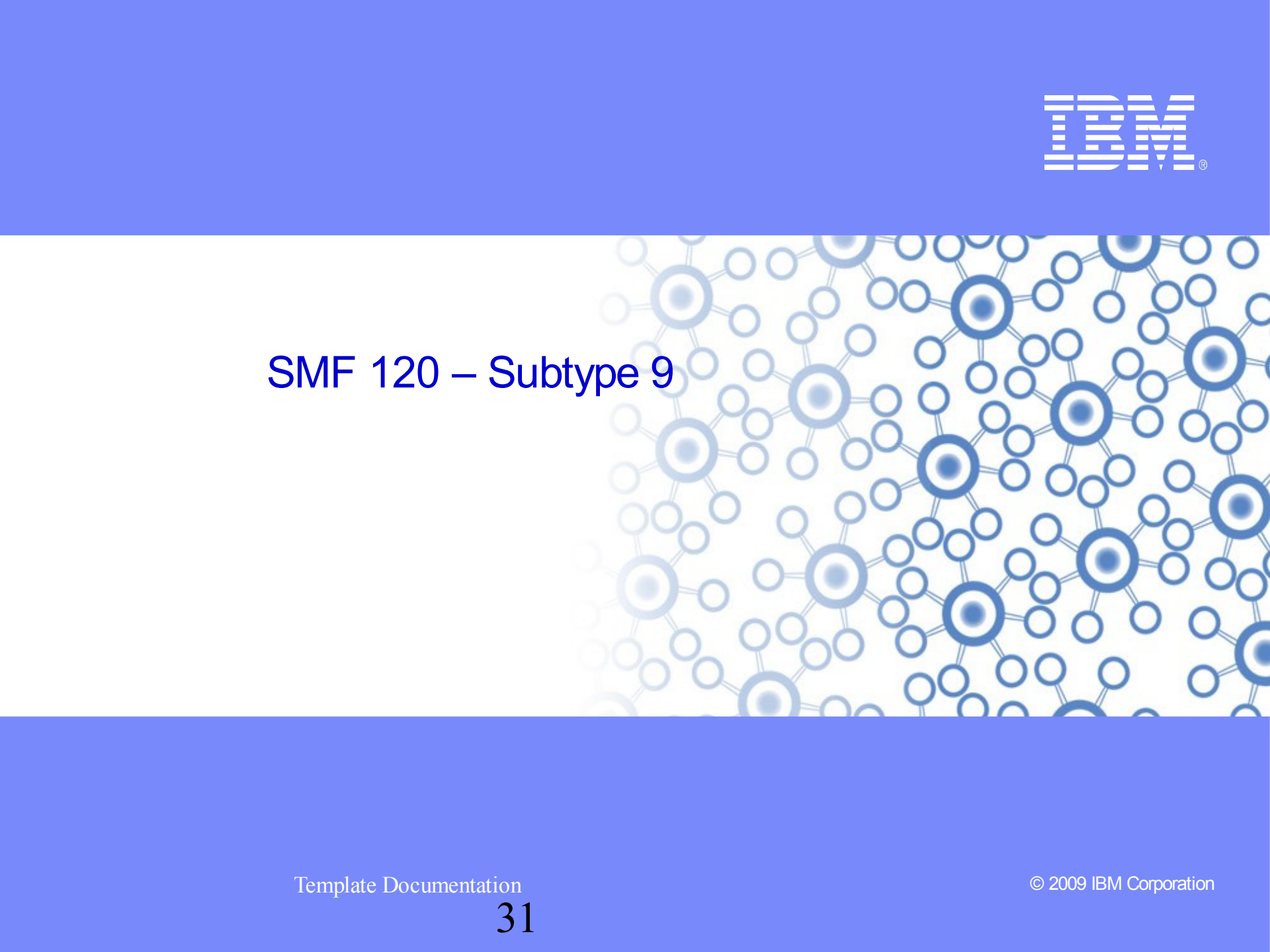
Before:

- DCS signalling over TCP -- only option
- As core group gets larger, overhead of TCP signalling can become prohibitive

V7.0:

- New "alternative protocol provider" -- XCF
- Lower overhead using XCF
  Percent reduction still be measured -- stay tuned
- Configured at the core group level
  Either DefaultCoreGroup or at other-defined core group
- The entire core group must be on z/OS
  Makes sense -- other platforms don't have access to Coupling Facility and XCF
- Entire core group must be at WebSphere V7.0
  Also makes sense -- if some part of the core group is at V6.1 or lower, the new function isn't there in the node.

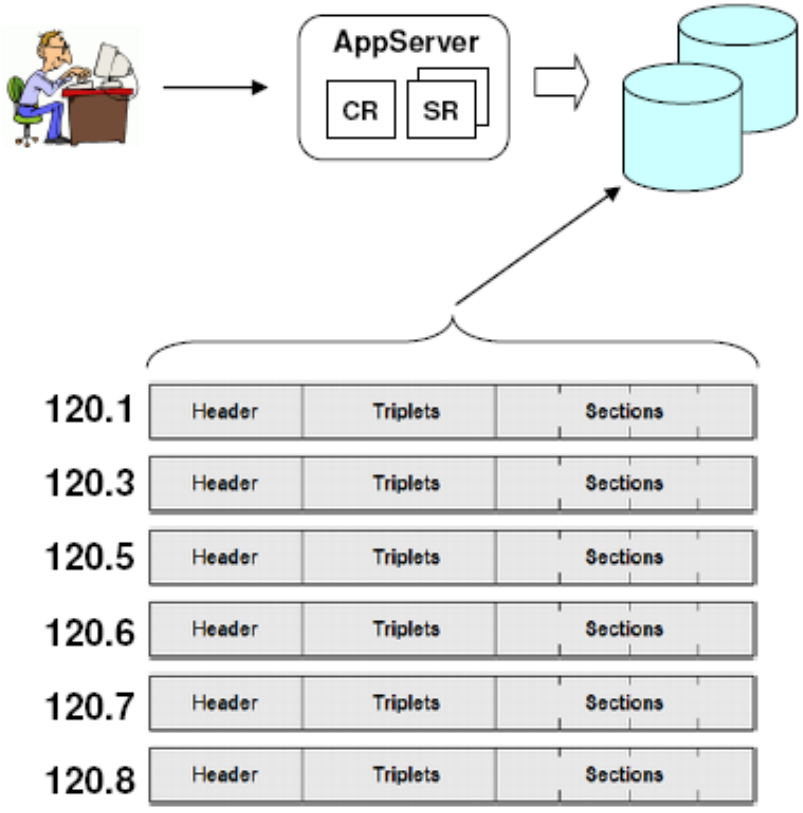
This is really "lower level plumbing" stuff … but if you have a very large topology, need HAM and have noticed high CPU, this may well be for you



SMF 120 – Subtype 9

Pre-V7 Behavior -- The Reason We Have a New Record Type

Before V7, WebSphere SMF information was spread across multiple subytpes. The fragmented view required careful coordination. Also heavy overhead.



Issues:

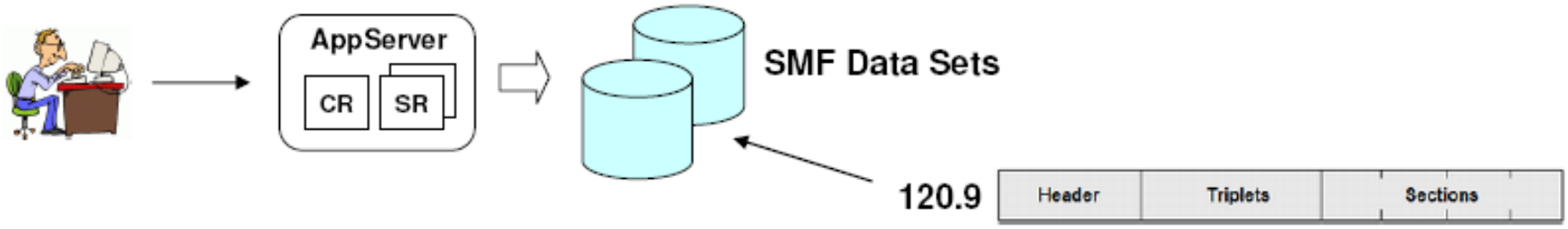
- Each request resulted in multiple records
- The overhead to capture and write was seen as excessive
- Many people turned off WAS z/OS SMF recording, which meant they lost visibility into usage
  - Which meant for many the accounting and chargeback was at the server level, which drove some to "one app per server" configurations. That's not always the most efficient thing to do, but if SMF was seen as too expensive, it's what they settled for.
- On top of that, the information people needed wasn't all there

In summary -- expensive and incomplete

Hence the birth of a new WAS z/OS SMF record

Basic Philosophy Behind SMF 120 Subtype 9 Design

The architects of this new SMF record had three essential objectives:



One Record

- All the information is in one record, not six like before
- Each record is initiated by a "request" to the WebSphere runtime

Information Customers Actually Wanted and Could Use

- Record content was based on customer interviews

Low System Overhead

- Minimize the CPU overhead of capturing and writing the records
- Provide optimized "basic request" information -- less than 1% overhead
- Provide "pay as you go" optional additional information -- single digit overhead

SMF 120 – New Subtype 9

A review of the new SMF record …

Provides more information about WebSphere transactions to help customers with chargeback information, such as:

What ran (Application, Servlet, EJB method, MDBean)

When it ran, how long it took

Who ran it (Calling host:Port, Security ID: Origin, Received, Invocation)

Resources used (CPU – CPs, zAAPs, zIIPs, Bytes transferred)

Other (Classification Names)

Overhead significantly less compared with current SMF Type 120 (subtypes 1-8) records.

A browser to display the contents of the new SMF records is provided.

SMF 120-9 Operational Considerations

Here are the commands and variables used to control the SMF 120-9 recording

Option to turn subtype 9 records on/off

(Environment Variable or Controller Property)

- `server_SMF_request_activity_enabled`
- Defaults to 0. Set to '1' (or 'true') to turn it on.

Control of optional data

- `server_SMF_request_activity_CPU_detail`
- `server_SMF_request_activity_timestamps`
- `server_SMF_request_activity_security`

This is the "pay as you go" optional information mentioned earlier

Modify commands for all options

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

These are z/OS "Modify" commands against the server controller region

Display command to report # records written and last error writing a record

```
F <server>,DISPLAY,SMF
BBOO0344I SMF 120-9: ON, CPU USAGE: ON, TIMESTAMPS: OFF, SECURITY INFO: OFF
BBOO0345I SMF 120-9: TIME OF LAST WRITE: 2008/05/31 15:32:51.112,
SUCCESSFUL WRITES: 2366157, FAILED WRITES: 0
BBOO0346I SMF 120-9: LAST FAILED WRITE TIME: NEVER, RC: 0
```

Summary of SMF 120.9

Before:

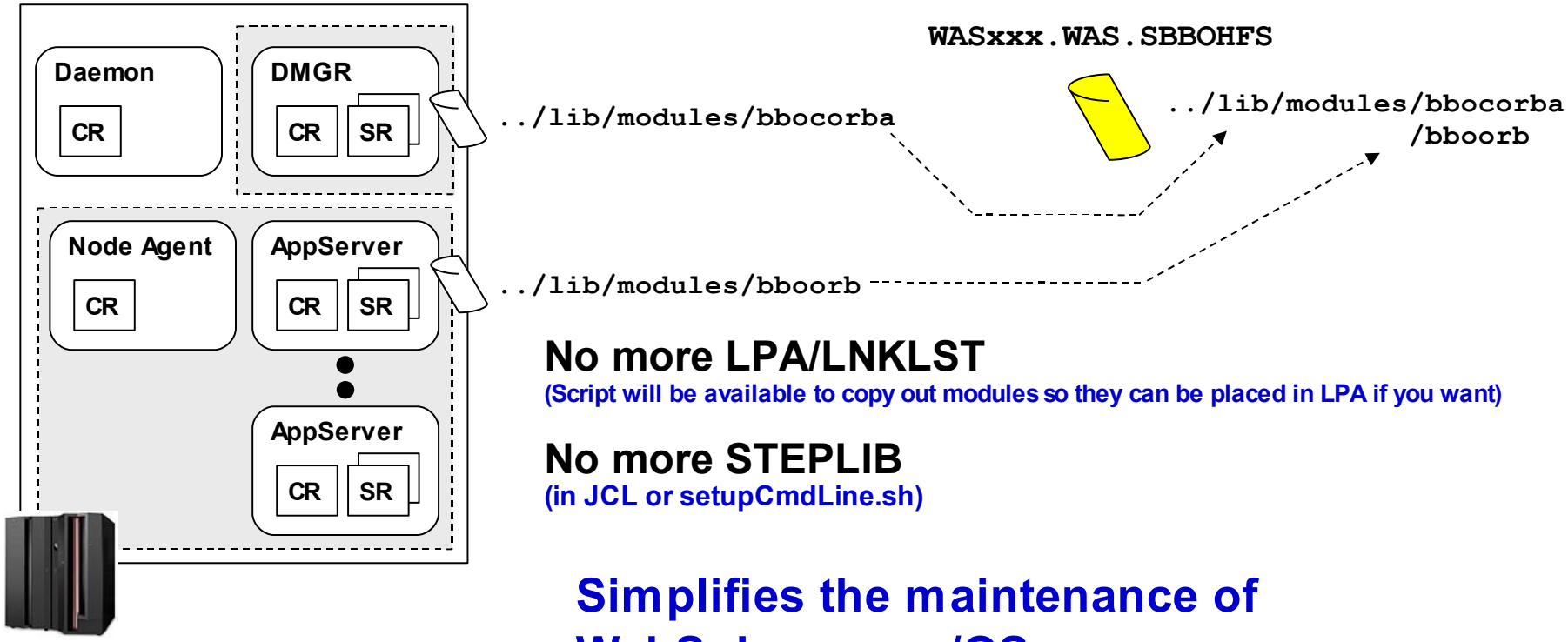
- Information in SMF 120 records unsatisfying
  - Fragmented across multiple subtypes; some information not present
- Overhead of turning them on was high

V7.0:

- New 120 record type -- Subtype 9
- Based on customer input of what information was desired
- Much more usable data for usage tracking and accounting
- Four increments: Request, CPU, Timestamps, Security
  - We believe the first will satisfy *most* requirements
- Settings may be static (variables) or dynamic (modify)
- Overhead of Request is very small

No More Load Module Libraries

The load modules will now be included in the HFS under `/lib/modules`.



No more LPA/LNKLST
(Script will be available to copy out modules so they can be placed in LPA if you want)

No more STEPLIB
(in JCL or setupCmdLine.sh)

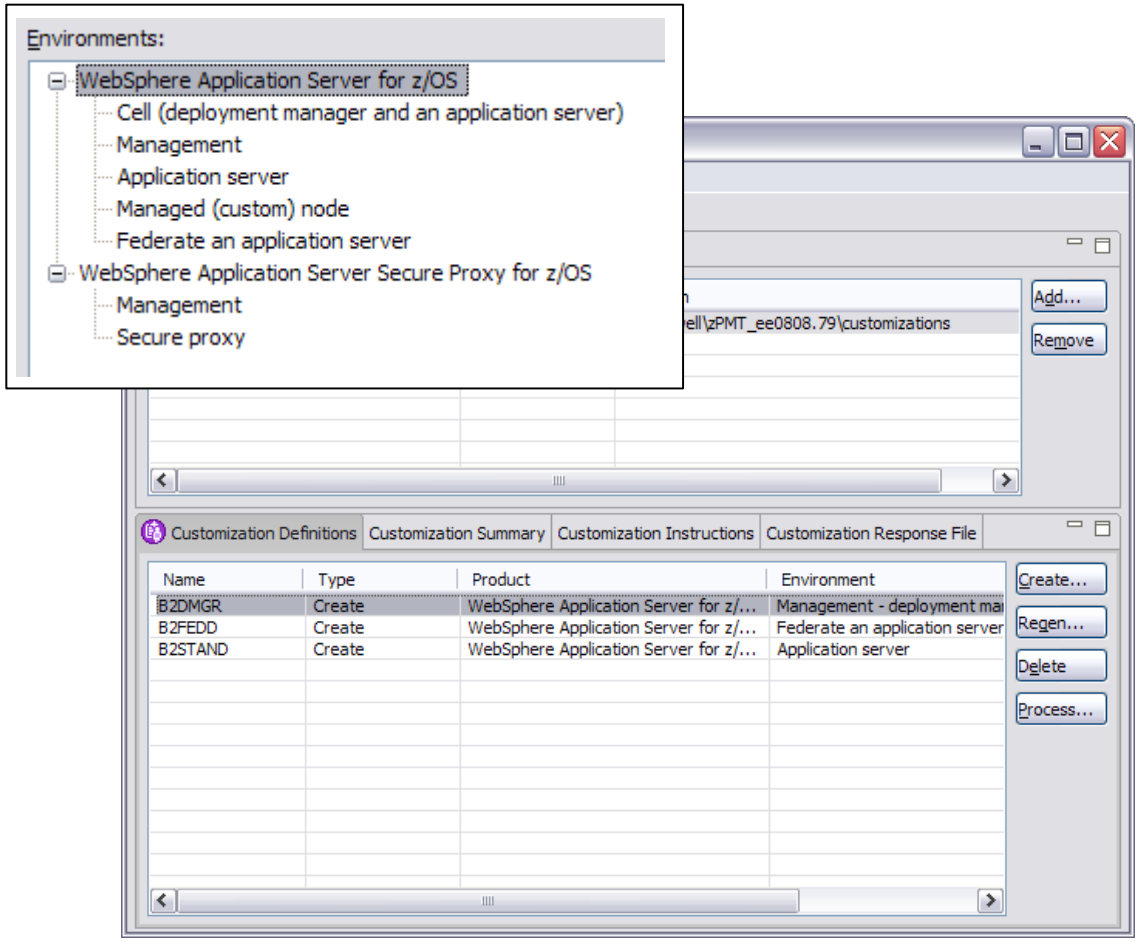
Simplifies the maintenance of WebSphere on z/OS.

Eliminates possibility of accidental mis-match between PDSE and HFS

Needs APAR OA25489

Configuration Tool

The ISPF panels are gone … the old AST-based zPMT gives way to a new smaller footprint "WebSphere Configuration Tools" PMT



- Similar in look and feel to the V6.1 zPMT
- Now called WCT - WebSphere Customization Tools
- No longer part of the AST … that means a smaller footprint and better responsiveness
- New customizations or migration of existing
- V7.0 as well as V6.1
- Uploaded jobs are similar to before, with a few differences

Performance

WAS v7 64-bit Provides New Performance Gains

WAS z/OS 64-bit support

- Shipped in 6.1 PTF 4
- Allows for 64 bit JVM
  - >1GB heap…
- 64 bit selected on a per server basis
  - Allows for incremental migration
- **Is the default mode in V7 (still configurable)**

Large Page Support

- New hardware feature introduced with IBM System z10
- Allocates and manages memory in 1M chunks

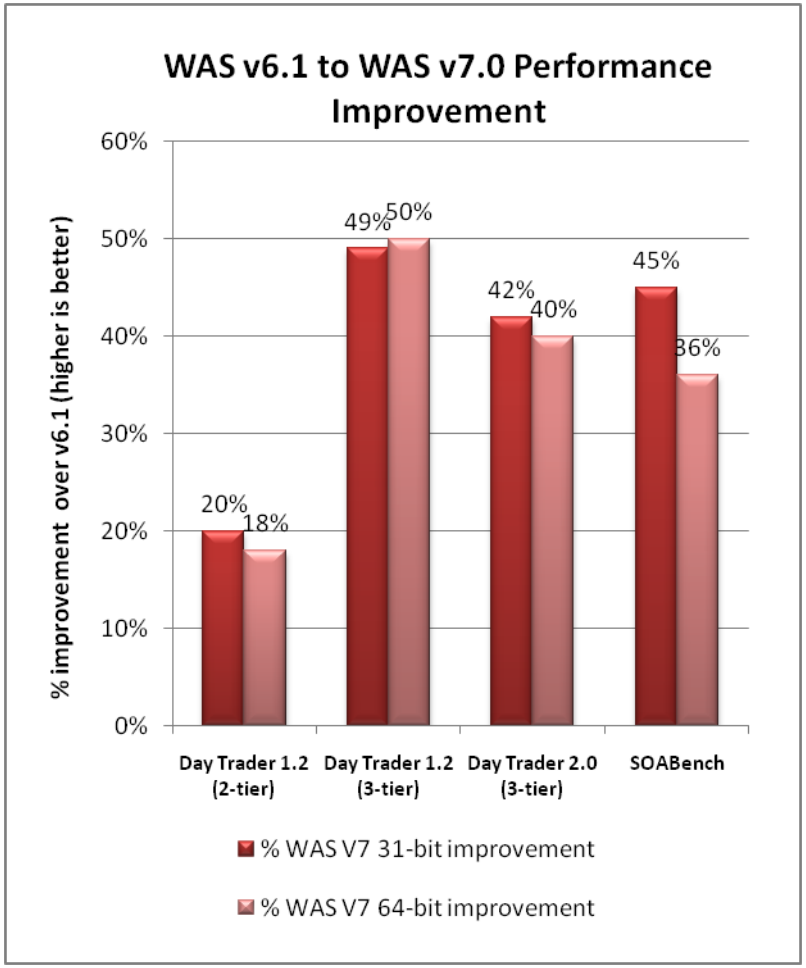
Compressed references

- New method for managing object pointers with the JVM
- Reduces the size of the 64-bit object pointer to 4 bytes

Pointer size	Space	Max heap	Efficiency
31 bits	2 GB	1.3 GB (z/OS)	100%
32 bits	4 GB	1.7 GB (Win) 3.2 GB (AIX)	100%
64 bits compressed (V7)	4 to 32 GB	4 to 32 GB	~70-100%
64 bits	16 EB	16 EB	50-70%

WAS z/OS V7 Performance Improvement

- **DayTrader 1.2 Performance Improvements from WAS v6.1 to V7**
  - Clear performance improvement by moving to WAS V7 for Legacy JEE applications
  - zWAS V7 performance is up 20% from v6.1 for 2-tier configuration (WAS/DB2 colocated)
  - JDK improvements
  - Servant/Controller communication optimizations
  - Codepath improvements throughout WebSphere V7
  - zWAS 3-tier performance is up 50% in 3-tier configuration
  - 2-tier improvements above plus RRS optimizations for single resource manager
- **DayTrader 2.0 EJB3 Performance Improvements from WAS v6.1 FeP to V7**
  - Large performance improvements by moving to V7 from v6.1 FeP for next generation JEE applications
  - WAS V7 is 40% faster than v6.1 EJB3 FeP
  - Runtime improvements above compounded with additional improvements in EJB3/OpenJPA code base
- **SOABench Performance Improvements from WAS v6.1+ WebServices Feature pack to V7**
  - Performance improved due to parser improvements for JAXB databinding
  - zWAS V7 improved 25-50% for payload sizes ranging from 3kin3kout to 100kin100kout.
  - Common payload 10kin10kout improved 45%

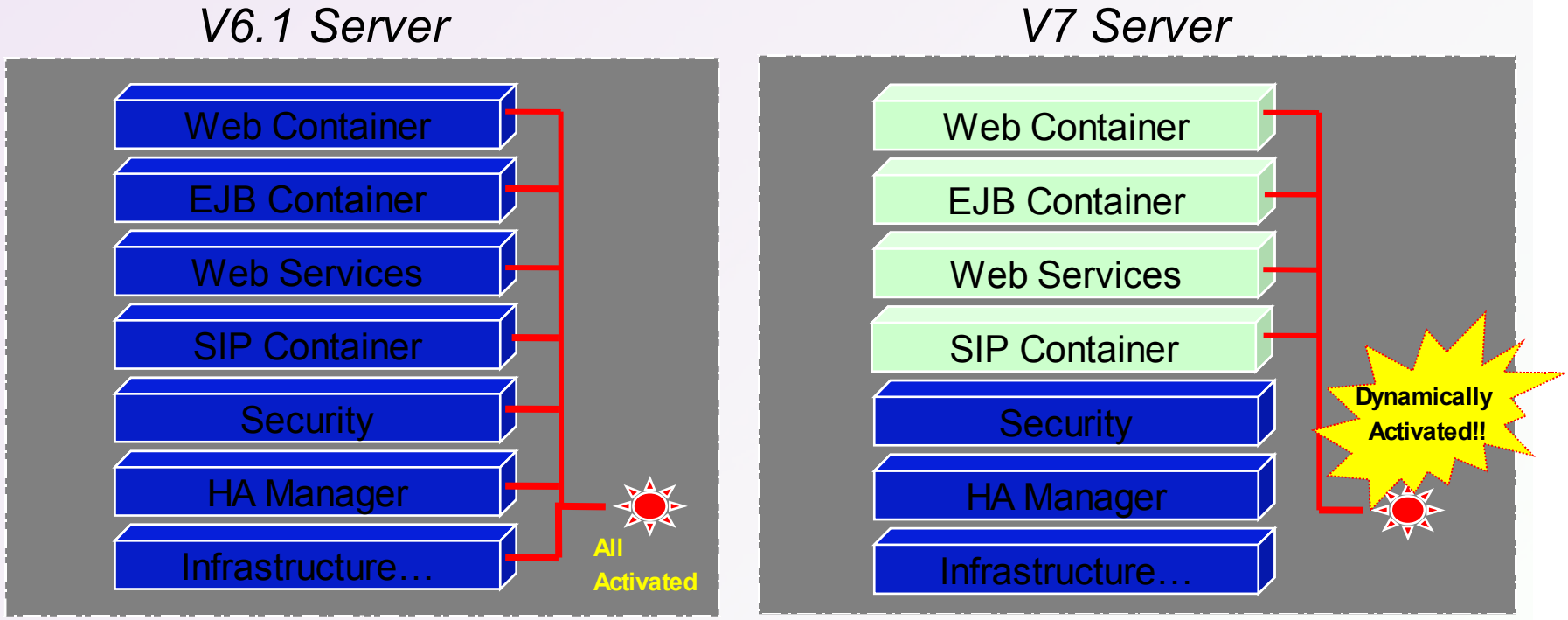


System Configuration
Workload: DayTrader, SOABench
SUT: IBM z10 Processor (model 2097 – 720) 4 x 4.4 GHz, 32 GB Real
Driver:

Performance

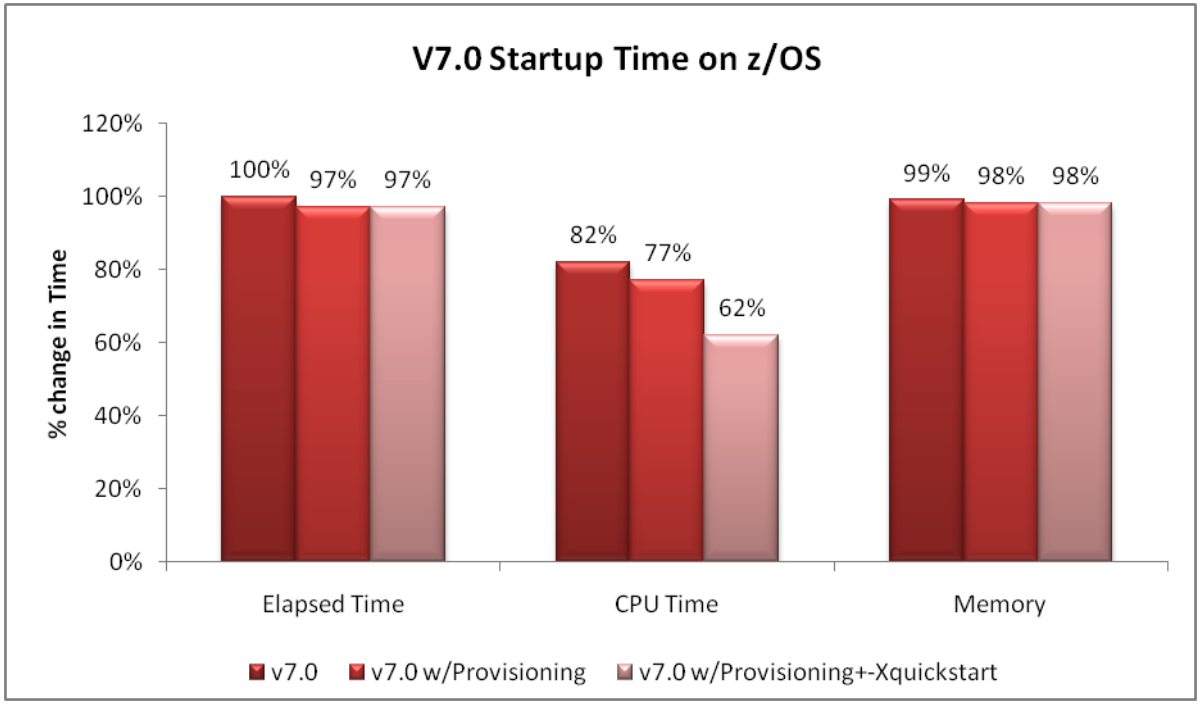
Runtime Provisioning

• Selects only the needed functions for memory and space efficiencies significantly reducing the application server footprint and start-up times



WAS V7 Startup time and memory footprint on z/OS (out of the box)

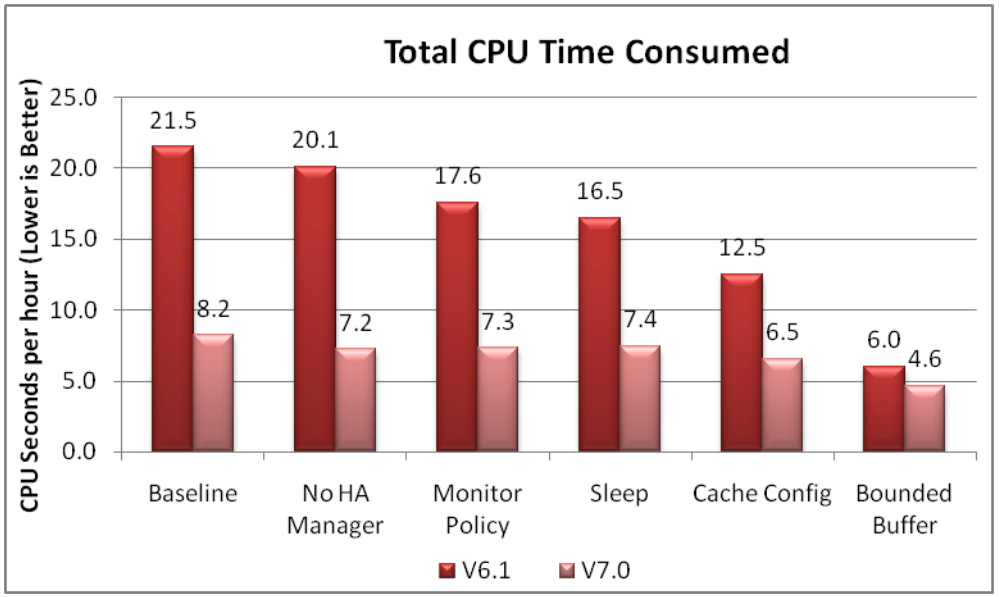
- Startup time and Footprint reduction from WAS v6.1 to V7 z/OS
- Both elapsed time and cpu time reduced by moving from WAS v6.1 to WAS V7 and memory footprint remained within 1-2% in 'Out of the box' scenario.
- WAS V7 is 3% faster in Elapsed time and 18% in CPU time compared to WAS v6.1.
- With provisioning enabled, CPU time is reduced by 23% and further reduced by total of 38% with jvm option – Xquickstart.
- Results achieved due to V7 enhancements and new features - provisioning, JDK improvements and parallel startup of the applications.



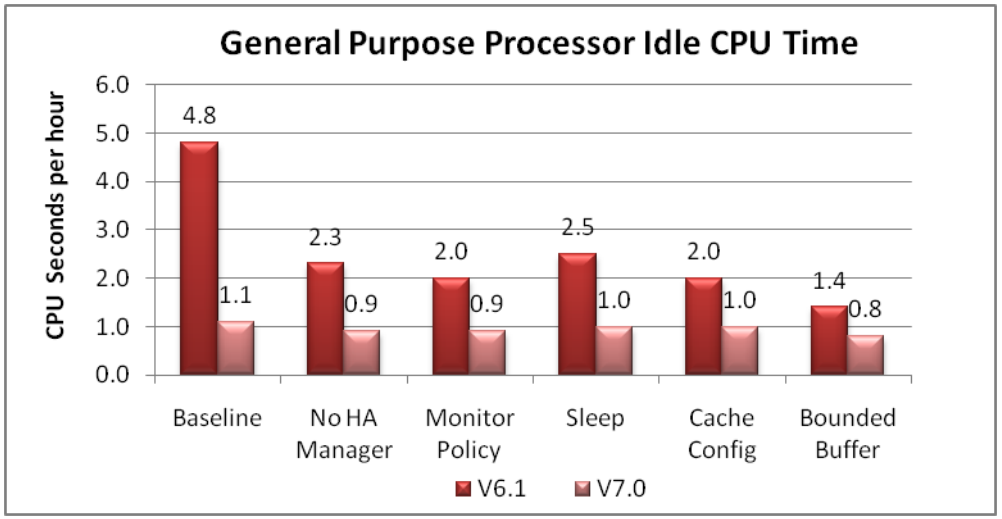
System Configuration
SUT: Z10 – 4.4Ghz, 4 processors 4 GB of real

z/OS: CPU consumption of an idle server

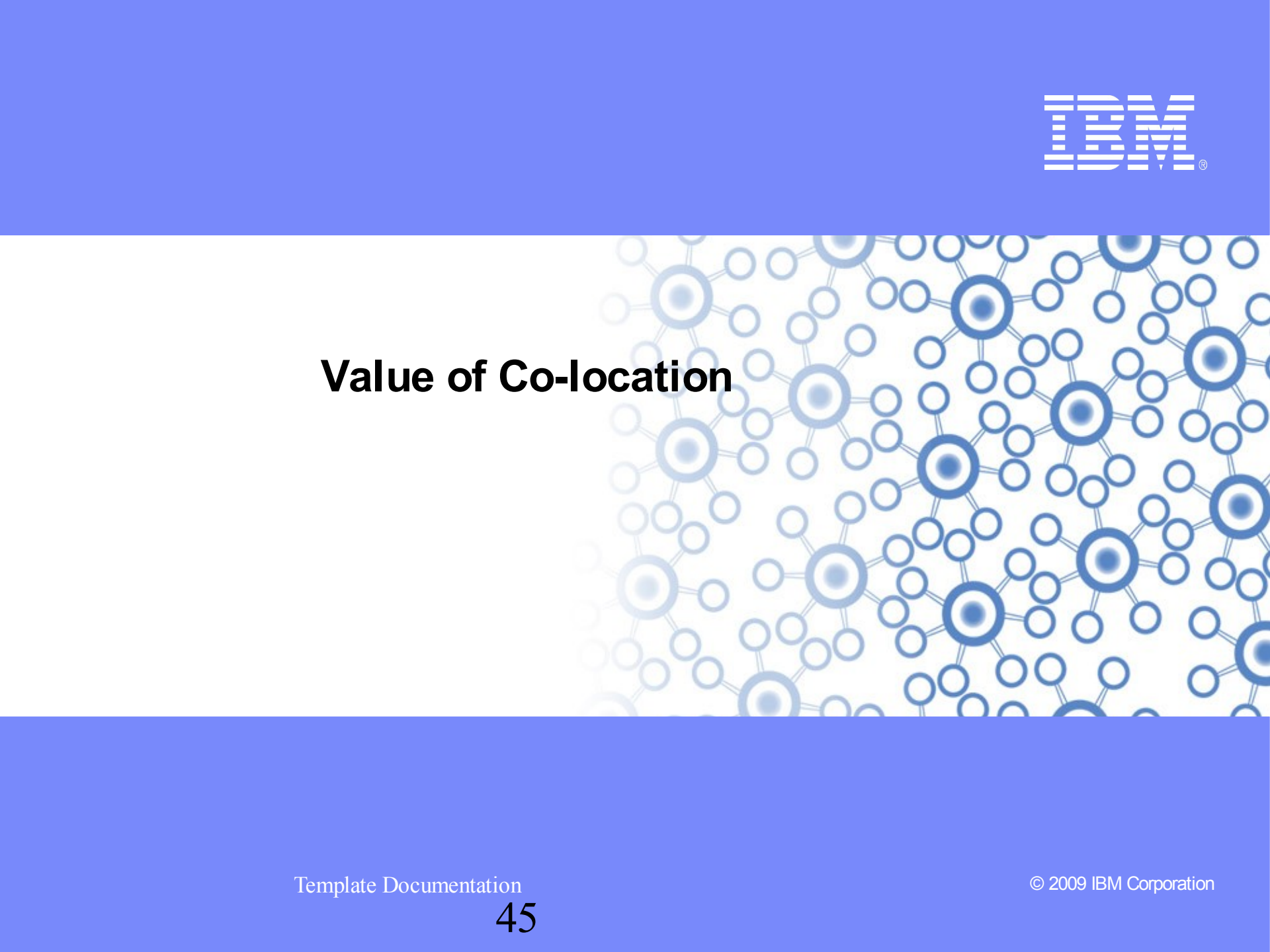
- **The amount of CPU time consumed by an idle server is significantly improved in V7 as compared to V6.1**
- **In V6.1, there are several properties that had been used to reduce the amount of CPU time consumed. V7 provides approximately the same level of consumption without the need to limit or disable these functions.**
  - High Availability Manager
  - The Monitor Policy Ping Interval
  - MaxActive and MaxPassive sleep intervals
  - JVM Cache Configuration settings
  - The Bounded Buffer Wait Interval



- **For most z/OS customers the cost of time on general purpose processors is much higher than the cost of time on zAAPs.**
- **In V7, the amount of idle server CPU time spent on general purpose processor time is reduced to only about 1 second per hour (or 24 seconds per day), or 0.03% !!**

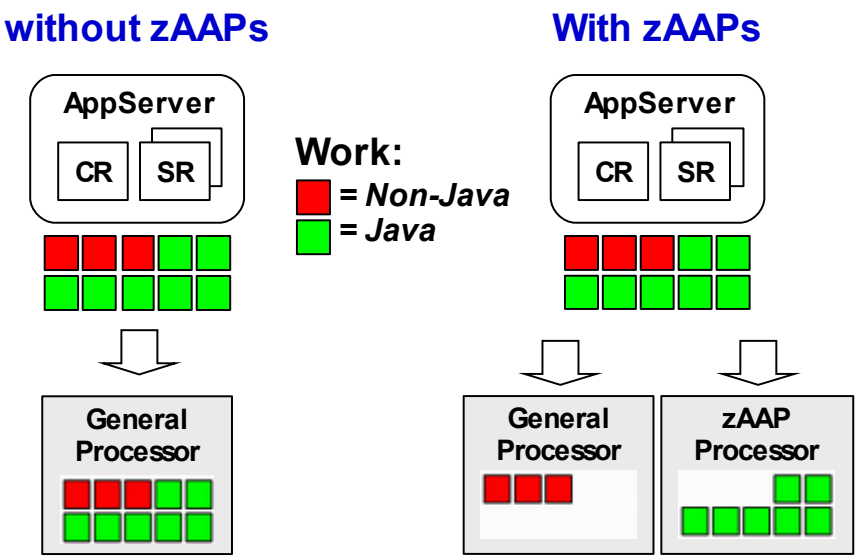


System Configuration
Workload: n/a
SUT: IBM z9 Processor (model 2094 – 715) 4 x 1.7 GHz, 10 GB Real
Driver: n/a



Value of Co-location

zAAP on zIIP Annoucement



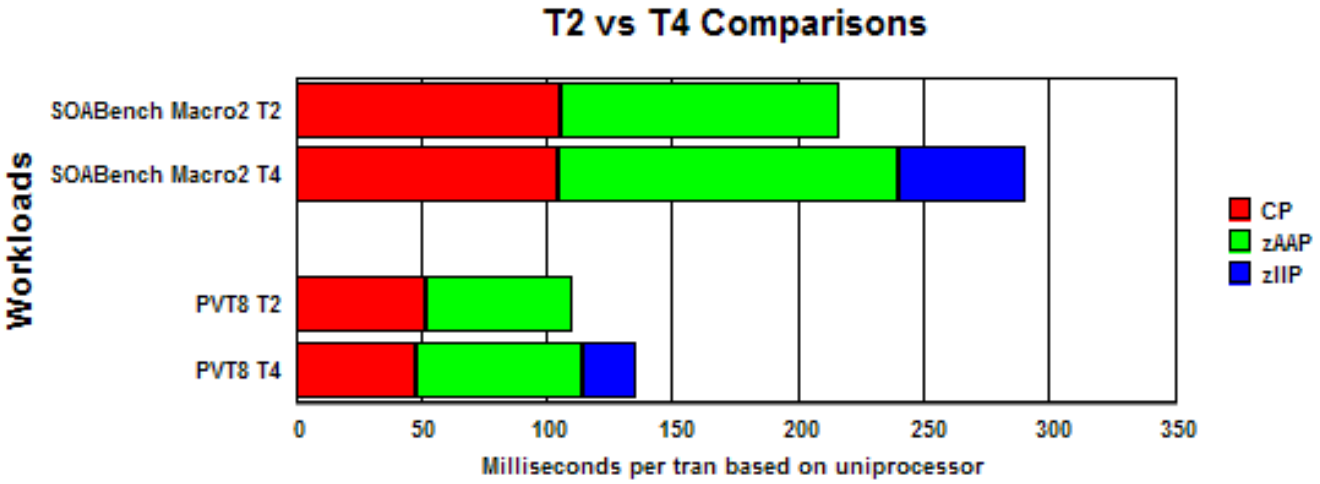
Prerequisits:

- zAAP eligible workload runs on zIIPs
- z9 or z10 Hardware
- z/OS V1.11 or z/OS V1.9 or V1.10 with the PTFs for APAR OA27495 installed
- This function is not available to z/OS LPARs if zAAPs are installed on that maschine

http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=an&subtype=ca&appname=gpateam&supplier=897&letternum=ENUS209-242

Result of the WebSphere z/OS Co-location Benchmarks (2)

Compared to a database connection over TCP/IP (JDBC Typ 4) Cross-Memory connections (JDBC Typ 2) could reduce the overall CPU consumption dramatically.



These values represent the utilization of the CPs, zAAPs and zIIPs on the system in total while the workload is processed with a constant transaction rate. This figure doesn't represent neither the enclave time for the application nor just the JDBC access to DB2.

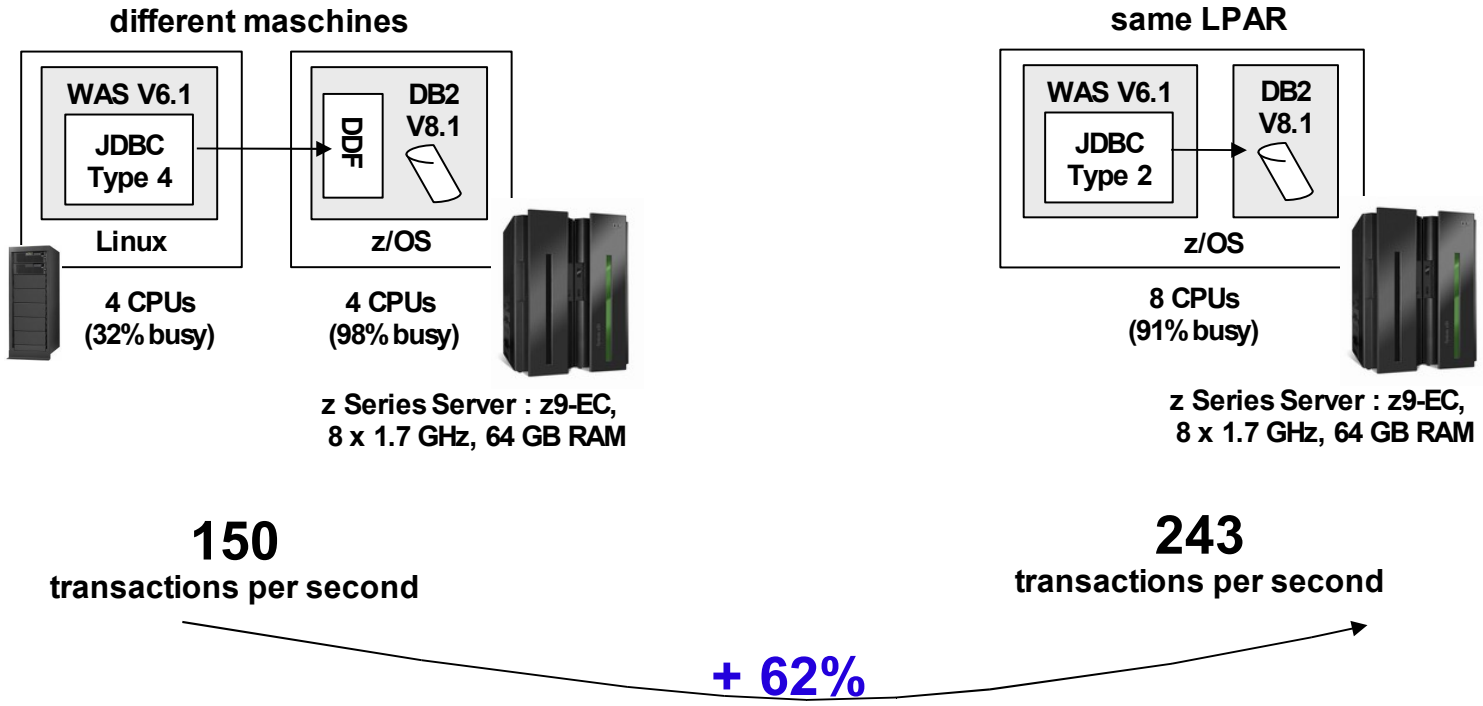
Quelle: WebSphere z/OS – The Value of Co-Location Benchmark

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101476

Result of the WebSphere z/OS Co-location Benchmarks (1)

If the data for an existing Java Enterprise applications are located in **DB2 z/OS, it makes sense** from a technical as well as financial point of view **to co-locate WAS z/OS and DB2 z/OS** in the same LPAR**.** In this **official benchmark the throughput** has been **increased by 62% by co-location.**

This throughput could be realized with **Cross-Memory Communication** (JDBC Type 2) from WAS z/OS to DB2 z/OS . The Overhead through network protocols will be eliminated. As a consequence the cpu consumption can be reduced dramatically.



Quelle: WebSphere z/OS – The Value of Co-Location Benchmark

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101476

Summary - WAS V7.0

Standards Currency

- Standards currency with Java EE 5, including EJB 3.0, enhances productivity and ease of use.
- New JDK 6 for improved performance and reduced footprint.
- Enhanced Web services standards.

Consumability, Simplicity and Performance Improvements

- Flexible systems management options.
- Enhanced diagnostic tools that help pinpoint problems.
- Security enhancements.
- New virtual image delivery option.
- Tight integration across the WebSphere family of products improves ROI .

New Application Types and Workloads

- Improved performance, scaling and reliability.
- Reduced cost of managing and administering large numbers of individual servers.

z/OS Key Differentiation

Performance

- ✓ Improvements in response time for static and dynamic content with Fast Response Cache Acceleration first availability in z/OS 1.9.
- ✓ Increased application runtime performance with focused analysis and code path improvement effort for JEE, Web Services and Connectors.
- ✓ **v7.0 performance is up 42% from v6.1**

High Availability and Reliability

- ✓ High Availability Manager based on Cross-System Coupling Facility (XCF).
- ✓ Thread Hang Recovery improves server reliability and performance.

Consumability and Usability

- ✓ Redesigned data collection facility to improve chargeback capabilities.
- ✓ More unified install and configuration tasks (load modules in HFS).

WebSphere software

WebSphere Application Server Roadshow

Are you making the most of your WebSphere Software investment?
What are other customers doing with the technology?

The WebSphere Application Server roadshow is designed to provide customers with a walk-through of all that's new in WAS V7 and the wider WAS family. There is a WAS V7 technical update, as well as discussion around extended capabilities and architectures.

The roadshow also provides details of unique function only available with WAS V7 on z/OS, and highlights the capability of the WebSphere platform to host both OLTP and Batch workloads.

The event is free of charge - it normally runs for about half a day on the customer site.

Please contact Steve Foley, IBM WebSphere Technical Specialist, to arrange the roadshow to visit your site – e-mail steve_foley@uk.ibm.com.

Alternatively, contact your local IBM account representative.

Trademarks

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

AIX*	ES/9000*	Language Environment*	S/390*
AnyNet*	FICON*	Lotus*	Sysplex Timer*
CICS*	FlashCopy*	Multiprise*	System Storage
DB2*	GDPS*	Notes*	System z
DB2 Connect	Geographically Dispersed Parallel Sysplex	OMEGAMON*	System z9
DB2 Universal Database	HiperSockets	On demand business logo	SystemPac*
developerWorks*	Hiperspace	OS/390*	Tivoli*
DFSMSdfp	HyperSwap	Parallel Sysplex*	TotalStorage*
DFSMSdss	IBM*	PR/SM	Virtualization Engine
DFSMShsm	IBM eServer	Processor Resource/ Systems Manager	VTAM*
DFSMSrmm	IBM e(logo)server*	pSeries*	WebSphere*
DFSORT	IBM logo*	RACF*	z/Architecture
Domino*	IMS	RAMAC*	z/OS*
DRDA*	Infoprint*	Redbook	z/VM*
Enterprise Storage Server*	IP PrintWay	RMF	zSeries*
ESCON*	iSeries		

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Intel is a trademark of the Intel Corporation in the United States and other countries.
Linux is a trademark of Linux Torvalds in the United States, other countries, or both.
Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
* 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.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.