How to Leverage Large Memory on z

SHARE in Orlando
August 10th, 2015
Session 17878

Elpida Tzortzatos:
elpida@us.ibm.com

Outline

- Large Memory Customer Value
- Example of Large Memory Benefits you can Leverage Today
- Large Memory and Analytic Workloads
- z/OS Potential Items for future Memory Management Enhancements

Industry Trends -- Response Time

- Response Time is Important
  - Imagine a human being waiting on a transaction that spans many data centers
    - Clouds, Multi-site Clients, Multi-Enterprise Transactions)
  - Clients configure systems to meet ever tighter response time goals
    - High Performance Servers, Disk, Networks
  - Response Time Gain in Productivity and Sales are real, measured and well documented
- In-Memory Databases gain response time in part by avoiding IO wait and CPU queing
  - IBM DB2 Local and Global Buffer Pools
  - IBM DB2 with BLU Acceleration dynamic in-memory columnar technologies.
- System z has one of the best performing Memory Nests in the industry
  - Huge Caches, High Performance Interconnects, Excellent Virtualization
  - Memory as a large, shared resource is a major technical value vs. blade form factor
- System z is positioned to provide substantial Response Time Value with Large Memory
  - z/OS, DB2, IMS, JAVA ,zVM, zLinux, Adabas, etc.
  - Analytics
    - ODM Decision Server Insights
    - Real-Time SMF Analytics

© 2014 IBM Corporation

Industry / Competitive Trends -- CPU Performance

- Technology plays a lesser role in driving CPU performance
  - Smarter Core Designs & heterogeneous Accelerators augment small CPU speed gains
  - True for x86, arm, power, z and every other complex processor (even GPGPUs)
- Clients see Value in CPU performance improvements
  - Reduce the need for application/system redesign to meet service goals
  - Improves response time and shrinks batch windows
- Clients on many platforms have historically used memory size increases to improve CPU performance, especially on database workloads
  - The z/OS stack has not fully harvested these memory related performance gains
- System z Clients can typically add memory to improve system performance without changing z/OS or Linux stack software pricing.
  - CPU Performance value prop:
    - Memory Cost vs. Software/CPU savings

© 2014 IBM Corporation

IBM z13 Large Memory – Client Value from Large Memory

Response Time

- Consistent fast transactional response time can result in an improved customer experience
- Near immediate response time can drive productivity accelerating the velocity of development
- Caching and other memory related techniques can help increase service levels to new heights

Availability

- Organizations trained to conserve memory can now relax restrictions to “enable the possible”
- Tuning knobs can be adjusted to their max to further exploit memory
- Examples:
  - Increased ability to handle workload spikes
  - Faster workload startup
  - Improved performance even given I/O disruptions

Economics

- Incentive pricing encourages customers to experiment with more memory and surface new use cases.

Innovation

- With mega memory, organizations can rethink and simplify application design for new business advantages
  - Example collocate analytics and in memory data stores for high performance data mining solutions

© 2015 IBM Corporation

z Systems Memory

Industry leading tiered memory nest design designed for speed. Designed for extreme RAS with concurrent upgrade, error recovery, security

- IBM z Systems™ is designed to scale, and memory is one component of the balanced design
- IBM z13™ (z13) offers a compute intensive design with 141 configurable cores; Up to 10 TB memory per CEC.
- z/OS® V2.2 has a 4 TB Maximum. RAIM memory.
- Along with memory we have SMT threads for higher concurrent processing and designed for improved throughput
- Tiered cache design, private and shared, instruction and data
- Designed for high transaction processing, for superior response time and CPU savings
- High availability and excellent memory failover /recovery

Consumers of large memory

- DB2® Buffer pools
- MQ
- Cognos® Cubes
- CICS® pools
- Large tables
- I/O intensive work
- Large batch sorts
- IMS™ PSBs

Large Memory Value

- Memory Related Performance Gains
  - **Substantial Latency Reduction for OLTP workloads**
    - Significant response time reductions
    - Increased transaction rates
  - **In-Memory Databases dramatic gains in response time by avoiding IO wait**
  - **Batch Window Reduction**
    - Potentially increase parallelism of batch workloads (e.g. more parallel sorts)
    - Potentially improve single thread performance for complex queries
  - **Reducing time to insight for analytic workloads**
    - Can process data more efficiently helping organizations keep pace with influx of data
    - Reduces time it takes to get from raw data to business insight
  - **CPU performance improvements**
    - Improves response time and shrinks batch windows
    - Reduce the need for application/system redesign to meet service goals
    - Reduces CPU time per transaction

Outline

- Large Memory Customer Value
- **Example of Large Memory Benefits you can Leverage Today**
- Large Memory and Analytic Workloads
- z/OS Potential Items for future Memory Management Enhancements

WebSphere MQ for z/OS V8.0.0

• MQ Memory Related Benefits

- Large memory for IBM MQ V8 can help to cost effectively manage the increasing message volumes generated from today's mobile and cloud applications
- Exploiting large memory buffer pools in IBM MQ V8 can increase the process efficiency of IT integration

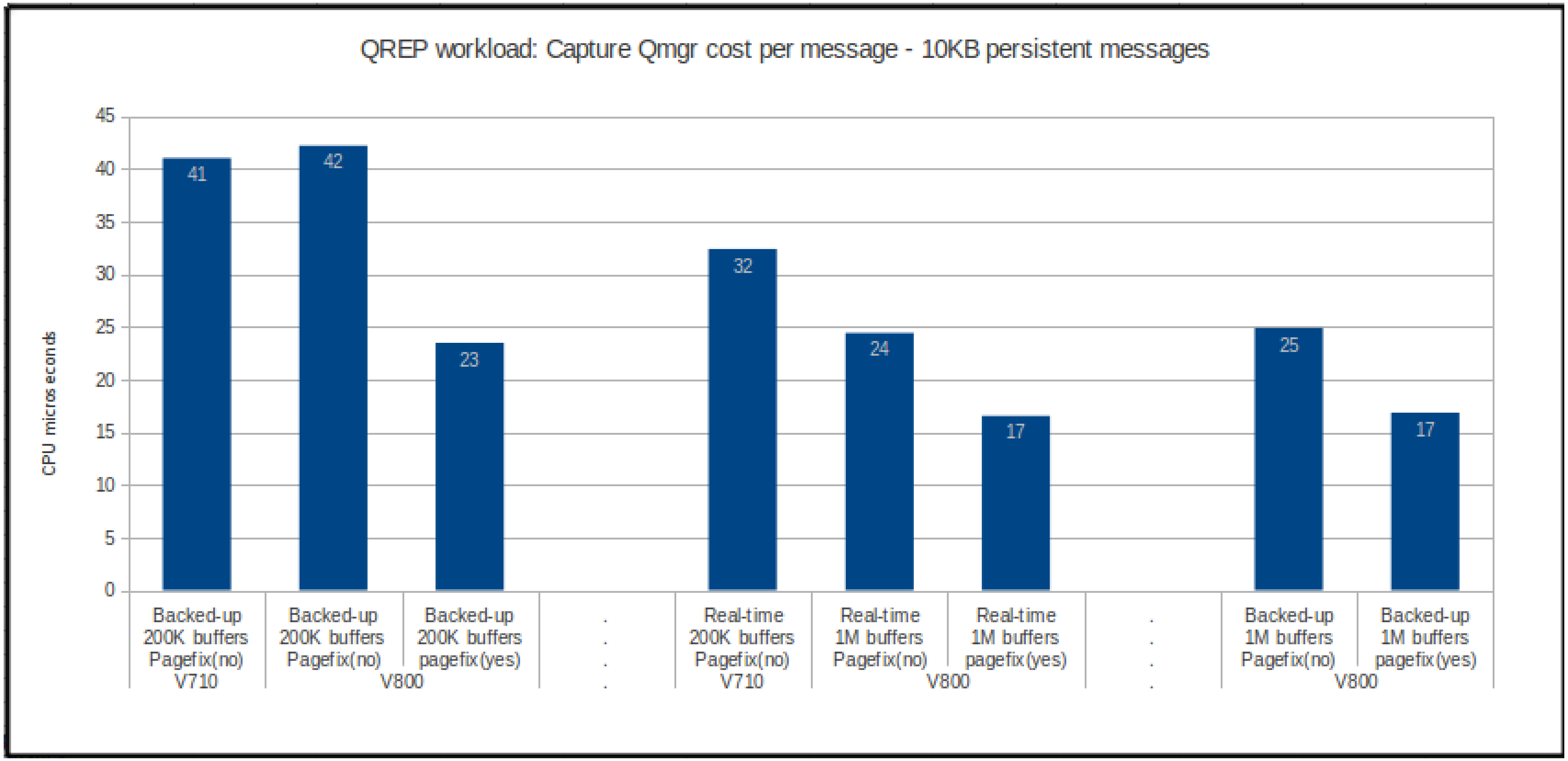
▪ ***MQv8 with above the bar memory, customer reduced batch elapsed time by 3X, with minimal CPU impact for large messages***

– Company tested large MQ messages (300KB) leveraging above-the-bar memory, reducing run times of their application from 26:76 to 7:50

© 2014 IBM Corporation

WebSphere MQ for z/OS V8.0.0

Using 10KB messages

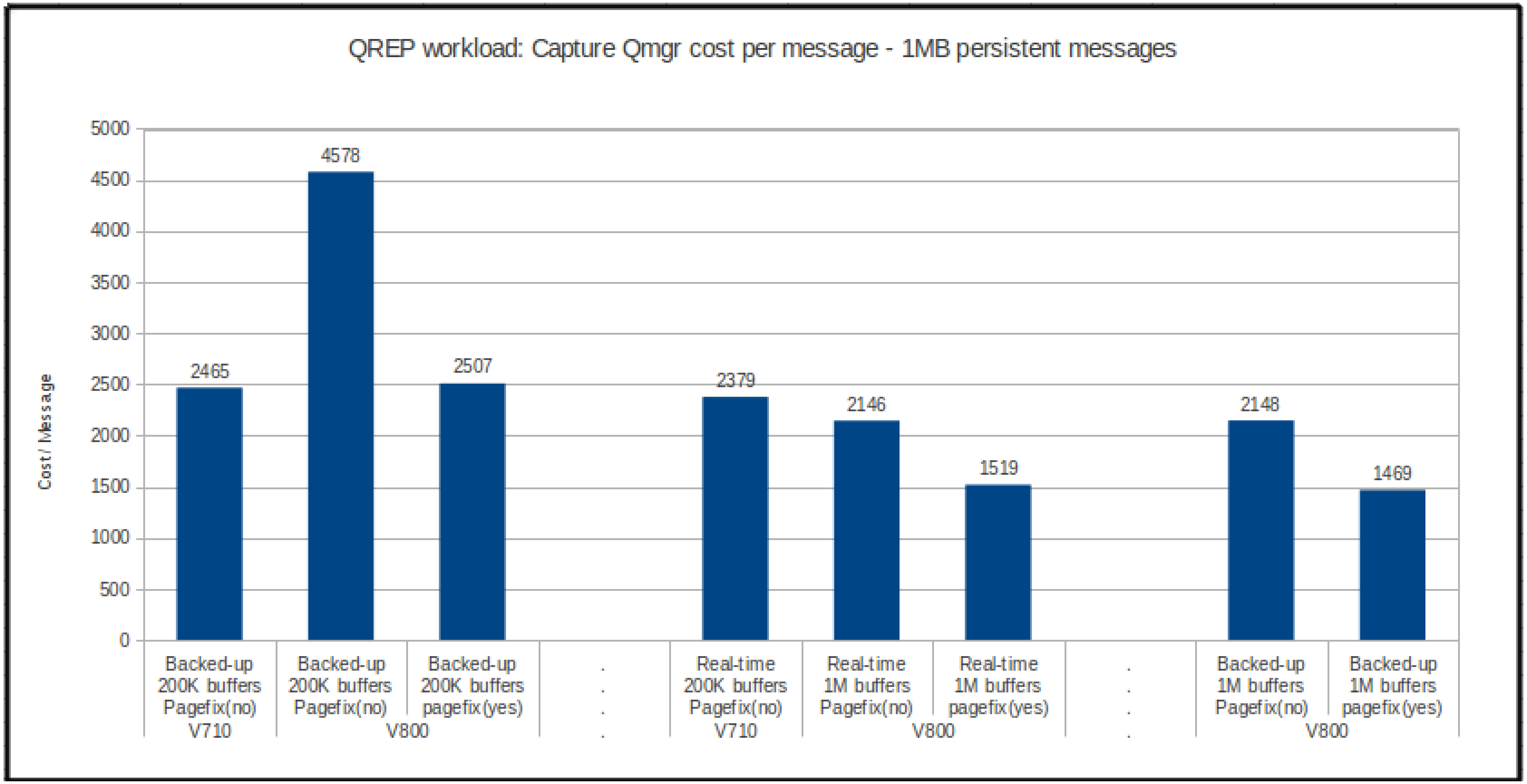


This replication workload simulates moving data from one system to another using MQ channels. As the data flows in a single direction, there is the potential for a build up of messages on the transmit queue in that the capture task puts messages more quickly than the channel initiator can get and send the messages, for example in the event of a network delay or the apply task is slow.

© 2014 IBM Corporation

WebSphere MQ for z/OS V8.0.0

Using 1MB messages



When larger buffer pools were used (version 8.0 only), PAGECLAS(FIXED4KB) reduced the cost (~20-30%) for both small and large message workloads.

© 2014 IBM Corporation

MQ and Large Memory Benefits

- MQ buffer pool (V8 64Bitconsolidated MQ w/more memory, reference account)
  - Bigger buffer pools better for performance, can be much bigger if above the bar
    - Should have sufficient memory available for buffer pools residence
    - Better to have smaller buffer pools that do not result in paging, than big ones that do
    - No point having a buffer pool bigger than the total size of pagesets that use it, including pageset expansion
    - **For a QRep workload v8 64bit, pagefix vs v7.1 31b, not fixable: up to 15-20% less CPU per processed message**
  - Aim for one buffer pool per pageset, as this provides better application isolation.
  - If sufficient memory, use page-fixed buffers
    - This can save CPU cost associated with page-fixing the buffers before the I/O, and then page-unfixing them
    - Internal tests show queue manager CPU cost per 10kB message dropped by 48% when 4GB buffers were fixed
  - There are benefits to locating buffer pools above the bar
    - 31 bit virtual storage constraint relief – for example more space for common storage.
    - If buffer pool needs to be increased while being heavily used, there is less impact by adding more buffers above the bar
  - Deep SYSTEM.* queues might benefit from being in own buffer pool, if enough memory
  - QRep: these recommendations applicable to both capture and apply side queue managers
    - important buffer pools are those for the xmitq's on capture side and the apply queues on apply side
  - For further information about tuning buffer pools, see:
    - IBM MQ SupportPac MP16 - WebSphere® MQ for z/OS Capacity planning & tuning
    - MQ Performance Supportpac, MP16 - Definition of Buffer Pool Statistics, to help monitor buffer pool usage.
    - see MQ KnowledgeCenter http://www-01.ibm.com/support/knowledgecenter/SSFKSJ_8.0.0/com.ibm.mq.pla.doc/q006005_.htm)

© 2014 IBM Corporation

Java Large Memory Benefits

- **JAVA**
  - Changing business landscapes increase demand for memory usage and parallelism in z/OS. In-transaction analytics, sub-second response times, and greater demand due to mobile all increase the need for more data and better performance.
  - Shift in application and middleware programming models, persistency systems, and application development frameworks
  - Evolution of in-memory data-bases and analytics, large scale distributed caching systems like Websphere Extreme Scale, and object-relational mapping libraries for persistency such as JPA all drive increased memory usage.
  - incremental garbage collection technology like the Balanced GC policy to address increasing heap storage to thread performance ratios.
  - Exploitation of 1MB and 2GB pages for up to 5% or more CPU benefit

© 2015 IBM Corporation

WAS benchmark: z/OS Performance for Pageable Large Pages

❖ **The WAS Day Trader benchmarks showed up to an 8% performance improvement using Flash Express.**

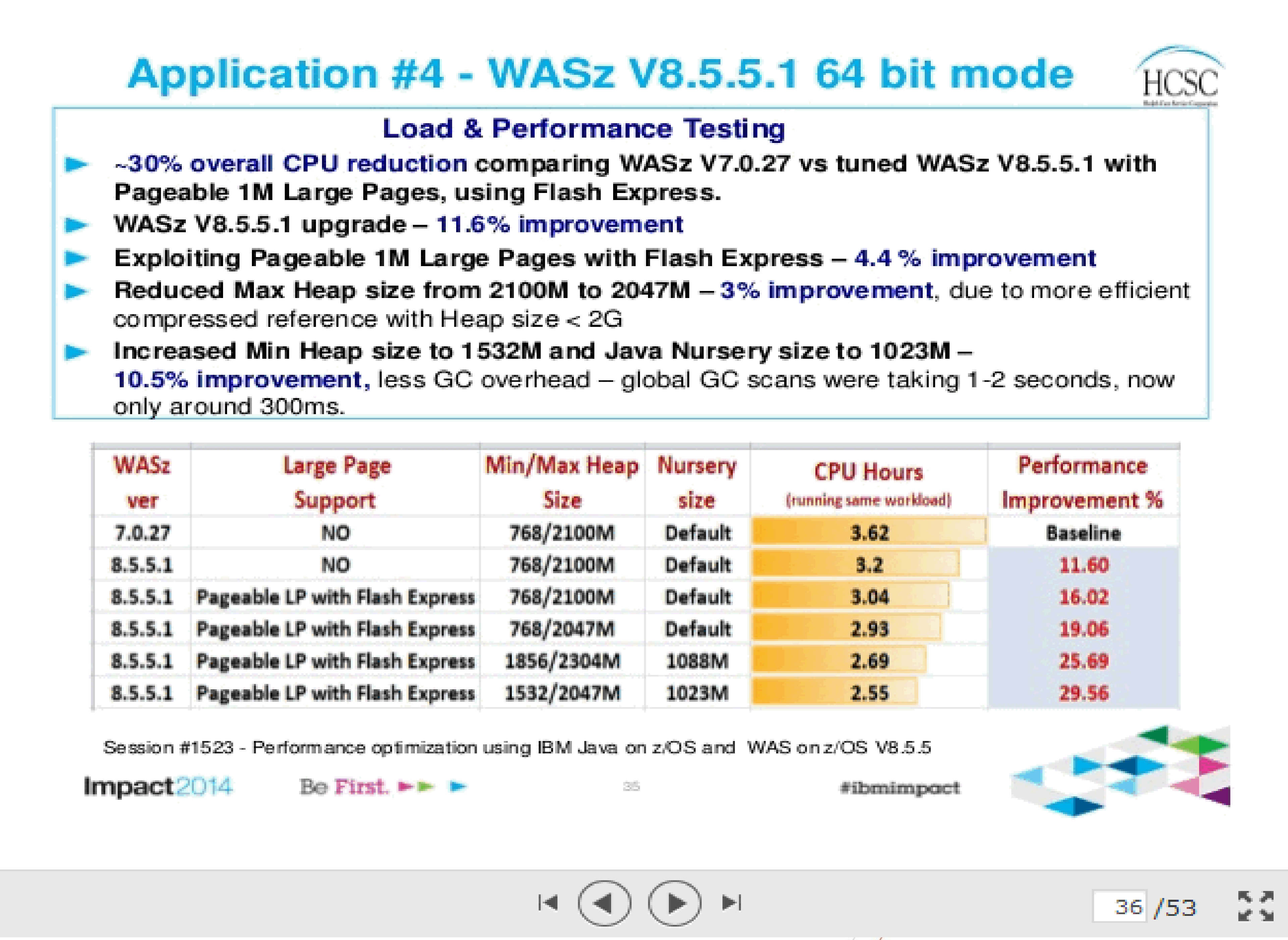
Java 7 SR3	JIT	Java Heap	Multi Threaded	WAS Day Trader 2.0
31 bit	yes	yes	4%	
64 bit	yes		1%	3%
64 bit		yes	4%	5%

* WAS Day Trader 64-bit Java 7 SR3 with JIT code cache & Java Heap

DETAILS

- 64-bit Java heap (1M fixed large pages (FLPs) or 1M Pageable (PLPs)) versus 4k pages
  Java heap 1M PLPs improve performance by about
  - 4% for Multi-Threaded workload
  - 5 % for WAS Day Trader 2.0
- 64-bit Java 7 SR3 with JIT code cache 1M PLPs vs without Flash
  - 3 % improvement for traditional WAS Day Trader 2.0*
  - 1 % improvement for Java Multi-Threaded workload
- 31-bit Java 7 SR3 with JIT code cache and Java heap 1M PLPs vs without Flash
  - 4 % improvement for Java Multi-Threaded workload

** Note: This test used 64-bit Java 7 SR3 with JIT code cache & Java Heap leveraging Flash and pageable large pages.*
Also , tests used WAS Day Trader app that supports PLP; earlier version of 31-bit Java did not allocate 1M large pages



Application #4 - WASz V8.5.5.1 64 bit mode

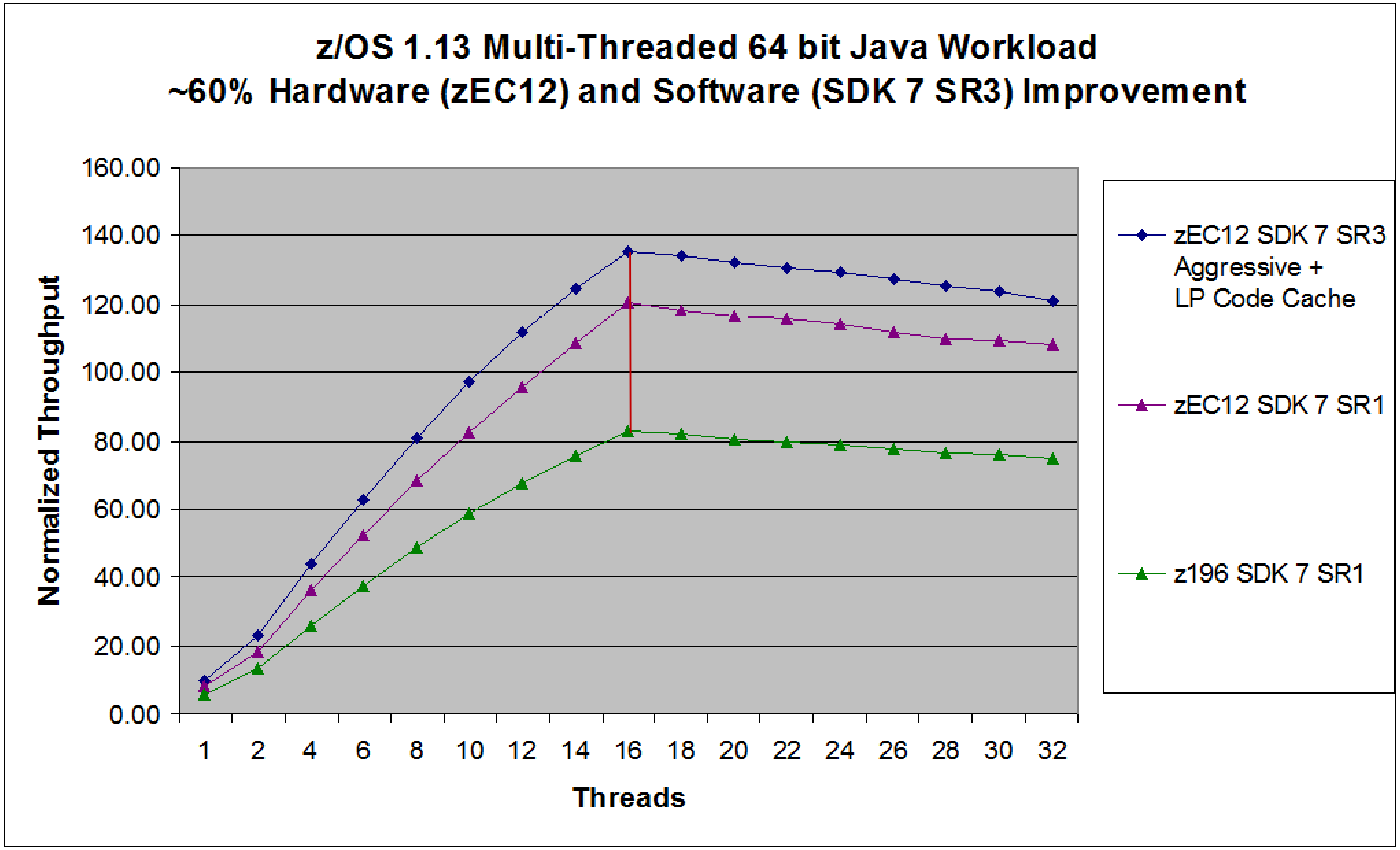
Load & Performance Testing

- **~30% overall CPU reduction comparing WASz V7.0.27 vs tuned WASz V8.5.5.1 with Pageable 1M Large Pages, using Flash Express.**
- **WASz V8.5.5.1 upgrade – 11.6% improvement**
- **Exploiting Pageable 1M Large Pages with Flash Express – 4.4 % improvement**
- **Reduced Max Heap size from 2100M to 2047M – 3% improvement**, due to more efficient compressed reference with Heap size < 2G
- **Increased Min Heap size to 1532M and Java Nursery size to 1023M – 10.5% improvement,** less GC overhead – global GC scans were taking 1-2 seconds, now only around 300ms.

WASz ver	Large Page Support	Min/Max Heap Size	Nursery size	CPU Hours (running same workload)	Performance Improvement %
7.0.27	NO	768/2100M	Default	3.62	Baseline
8.5.5.1	NO	768/2100M	Default	3.2	11.60
8.5.5.1	Pageable LP with Flash Express	768/2100M	Default	3.04	16.02
8.5.5.1	Pageable LP with Flash Express	768/2047M	Default	2.93	19.06
8.5.5.1	Pageable LP with Flash Express	1856/2304M	1088M	2.69	25.69
8.5.5.1	Pageable LP with Flash Express	1532/2047M	1023M	2.55	29.56

Session #1523 - Performance optimization using IBM Java on z/OS and WAS on z/OS V8.5.5

z/OS Java SDK 7:16-Way Performance Shows up to 60% Improvement 64-bit Java Multi-threaded Benchmark on 16-Way

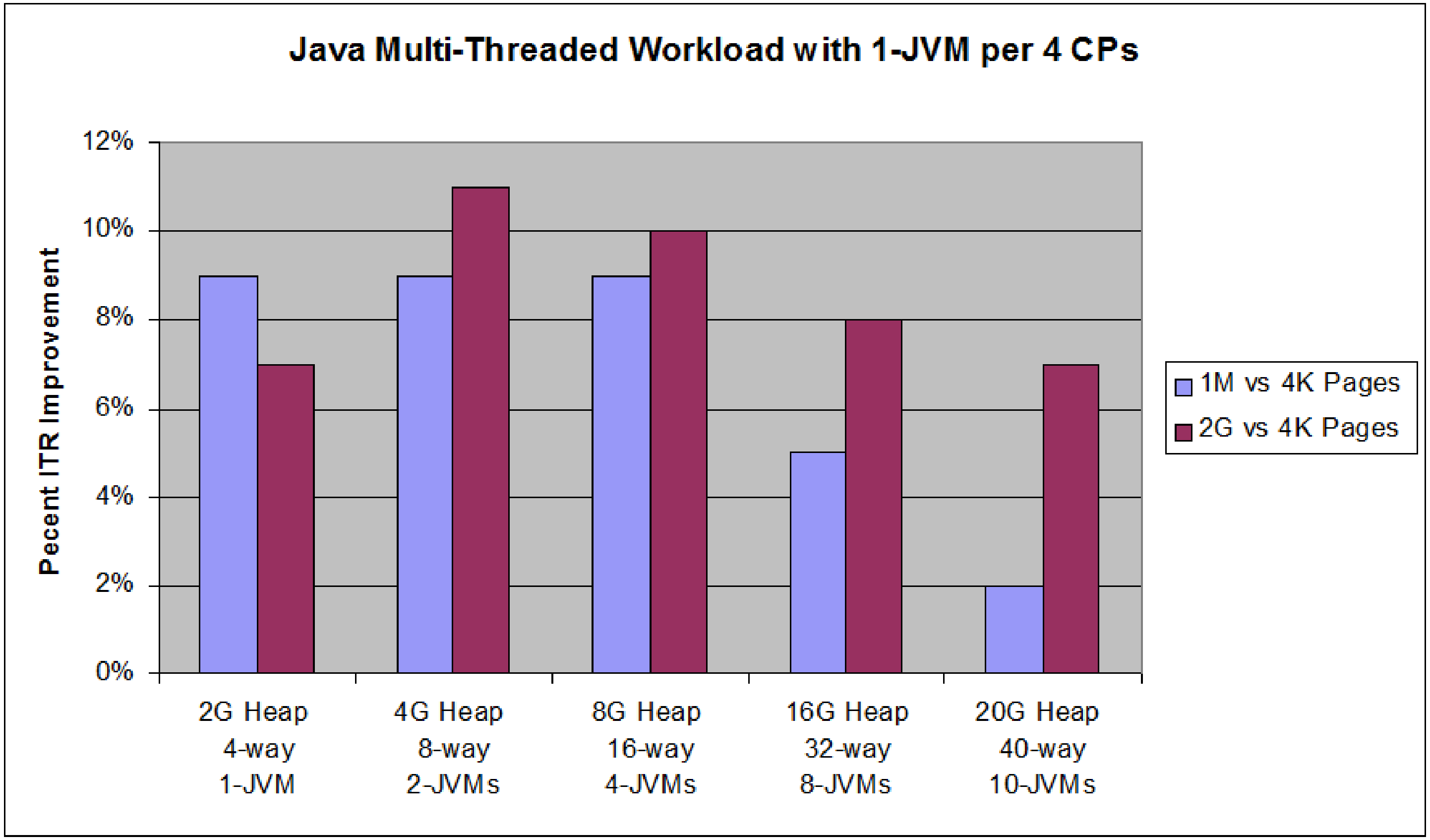


Aggregate 60% improvement from zEC12 and Java7SR3

- **zEC12 offers a ~45% improvement over z196 running the Java Multi-Threaded Benchmark**
- **Java7SR3 offers an additional ~13% improvement (-Xaggressive + Flash Express pageable 1Meg large pages)**

z/OS Java SDK 7: 2G Page Performance

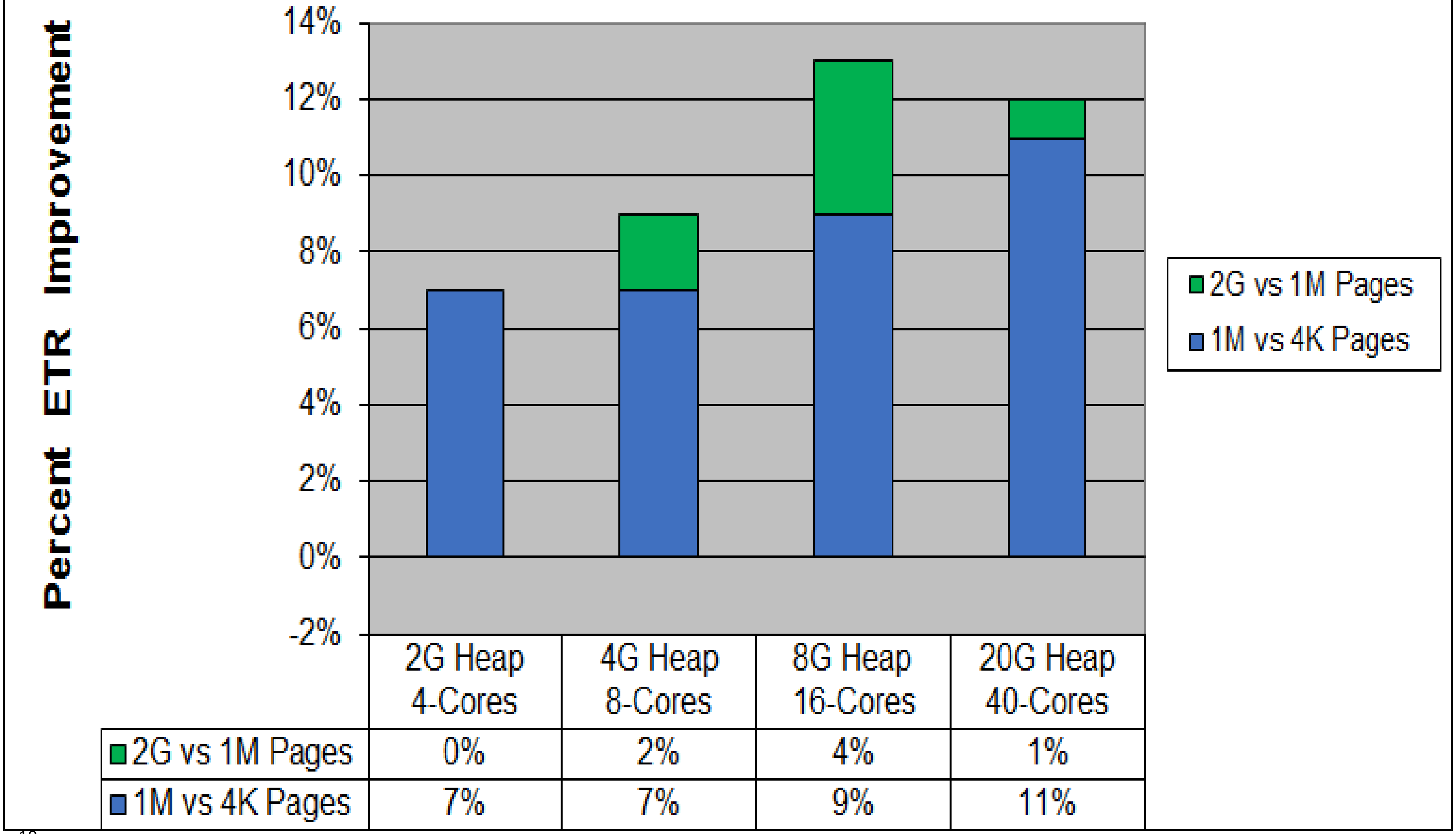
Multiple 4-way JVM Environment



2G large pages improve performance of multi-JVM environments with large aggregate footprint

(Controlled measurement environment, results may vary)

2G, 1M, and 4K Real Memory Pages on z13 no-SMT
z/OS Multi-Threaded 64 bit Java Workload
with 4 Cores per JVM

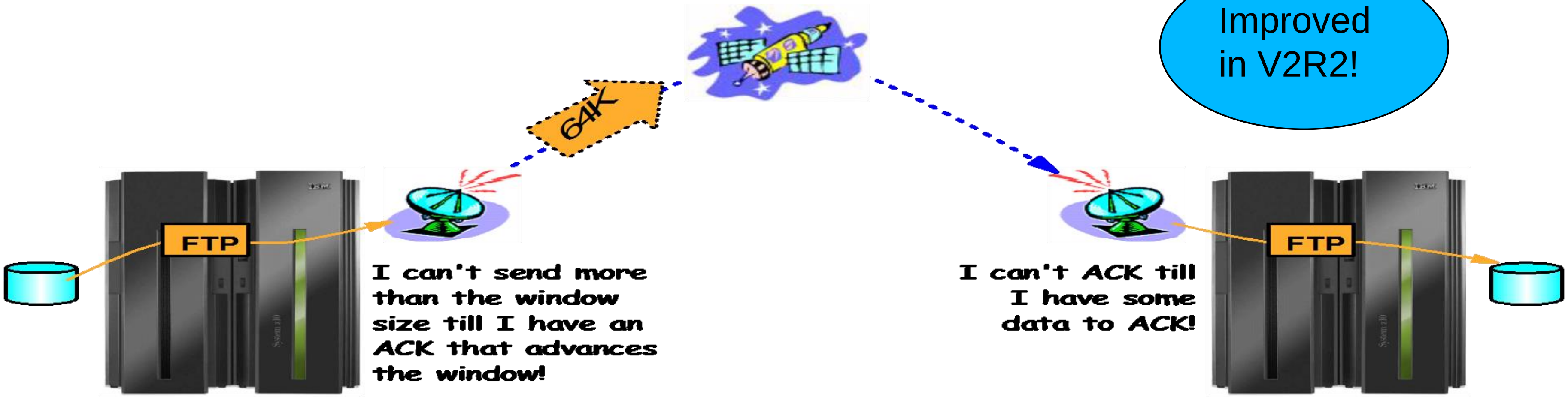


	2G Heap 4-Cores	4G Heap 8-Cores	8G Heap 16-Cores	20G Heap 40-Cores
🟩 2G vs 1M Pages	0%	2%	4%	1%
🟦 1M vs 4K Pages	7%	7%	9%	11%

(Controlled measurement environment, results may vary)

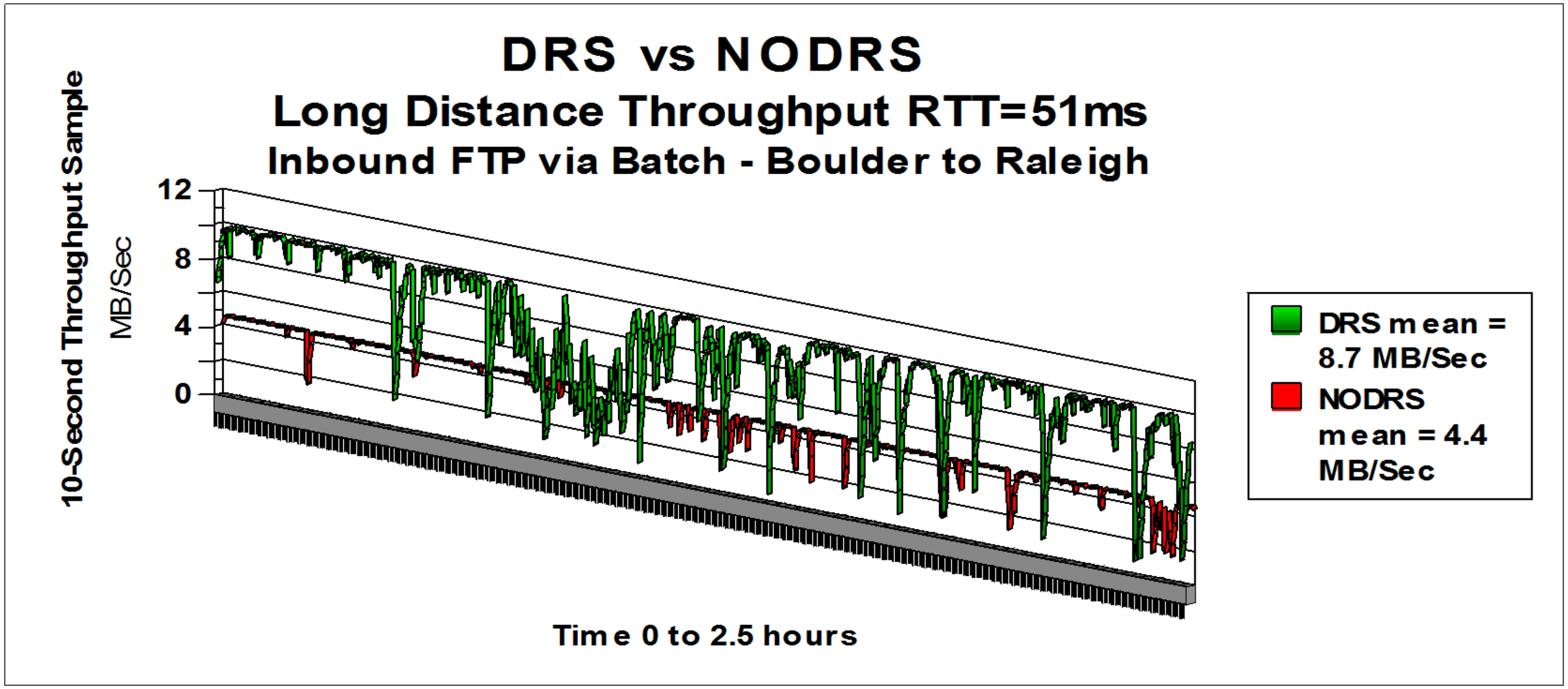
DRS - TCP throughput improvements for high-latency networks

Improved in V2R2!


64K
FTP
I can't send more than the window size till I have an ACK that advances the window!
I can't ACK till I have some data to ACK!
FTP

- ***TCP/IP in z/OS has implemented an enhancement known as Dynamic Right Sizing***
- ***Helps improve performance for streaming TCP connections over networks with large bandwidth and high latency when z/OS is the receiver***
  - ***By automatically tuning the ideal window size beyond the current window receive size setting for connections that can benefit from it***
    - ***May exceed current maximum window size of 512K for such TCP connections (up to 2MB)***
  - ***This function does not take effect for applications which use a TCP receive buffer size smaller than 64K***
    - ***TCP/IP will automatically revert back to normal TCP window size if it detects that the receiving application can not keep up with the incoming data***

Dynamic Right Sizing



Over an extended 2.5 hour experiment, the DRS enabled receiver averaged double the throughput compared to no DRS.
This experiment repeatedly transferred a 2.8 GB file, and DRS never disabled over the 2.5 hour period.

.

© 2014 IBM Corporation

Large Memory Deployment Recommendations

- **Very Rough "rule of thumb" performance expectations**

5%

- **Step 1 Convert pagable DB2 buffer to Page Fixed buffers at current BP size**
  - – Gain 0-6%, most Clients see 2-3% CPU benefit for BPs with IO activity
  - – Use Flash and/or additional real memory to mitigate any real memory concerns that are currently preventing you from page fixing DB2 buffers.
  - – IBM performance testing for very large memory will assume Page Fixed buffers

- **Step 2 Deploy 1MB or 2GB Large Pages for Page Fixed DB2 Buffers**
  - – Gain up to another 1-2% CPU benefit

- **Step 3 Deploy Pageable 1MB pages**
  - – (requires Flash Express, skip step 3 if you don't have Flash)
  - – Gain up to 1% with 1MB pages for DB2 11 executable code with z/OS 2.1
  - – Expect to gain additional CPU benefit when z/OS 2.2 delivers Shared 64bit 1MB Pageable pages exploited by DB2.

5%

- **Step 4 Increase size of DB2 local buffer pools to up to 100GB, in data sharing increase size of Global Buffers Pools enough to support local buffer pool size.**
  - – Gain up to 5% depending workload profile and tuning
  - – Note 100GB per DB2 means up to 1TB per z/OS, and >> 1TB

Example OLTP-SAP Benchmark Illustrates benefits

Significant performance improvements when more memory was used for DB2 buffer pools using the SAP Banking Services (SBS) Day Posting workload

Tests showed:

- Reduced response time of up to **70%**
- Increased transaction rates of up to **37%**
- Savings in relative CPU time per transaction of up to **25% (ITR)**
- Up to a **97%** reduction in DB2 synchronous reads
- Caching data in buffer pools helps improve response time, increase throughput, and deliver CPU savings.
- Reading data from in memory pools vs disk I/O helps improve DB2 request time for superior service levels 62

SAP Day Posting workload.

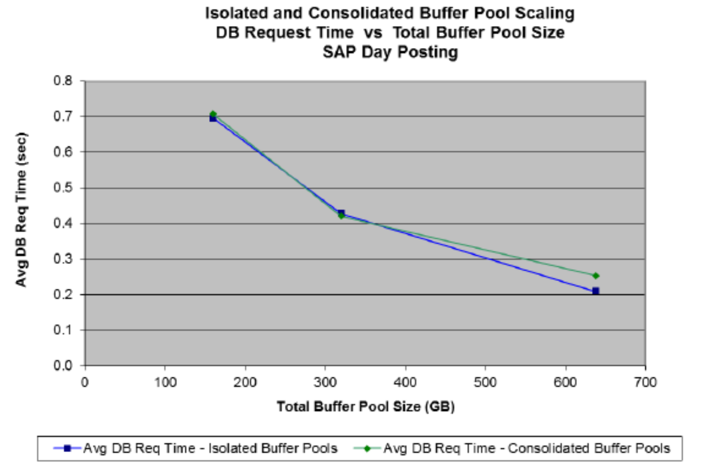


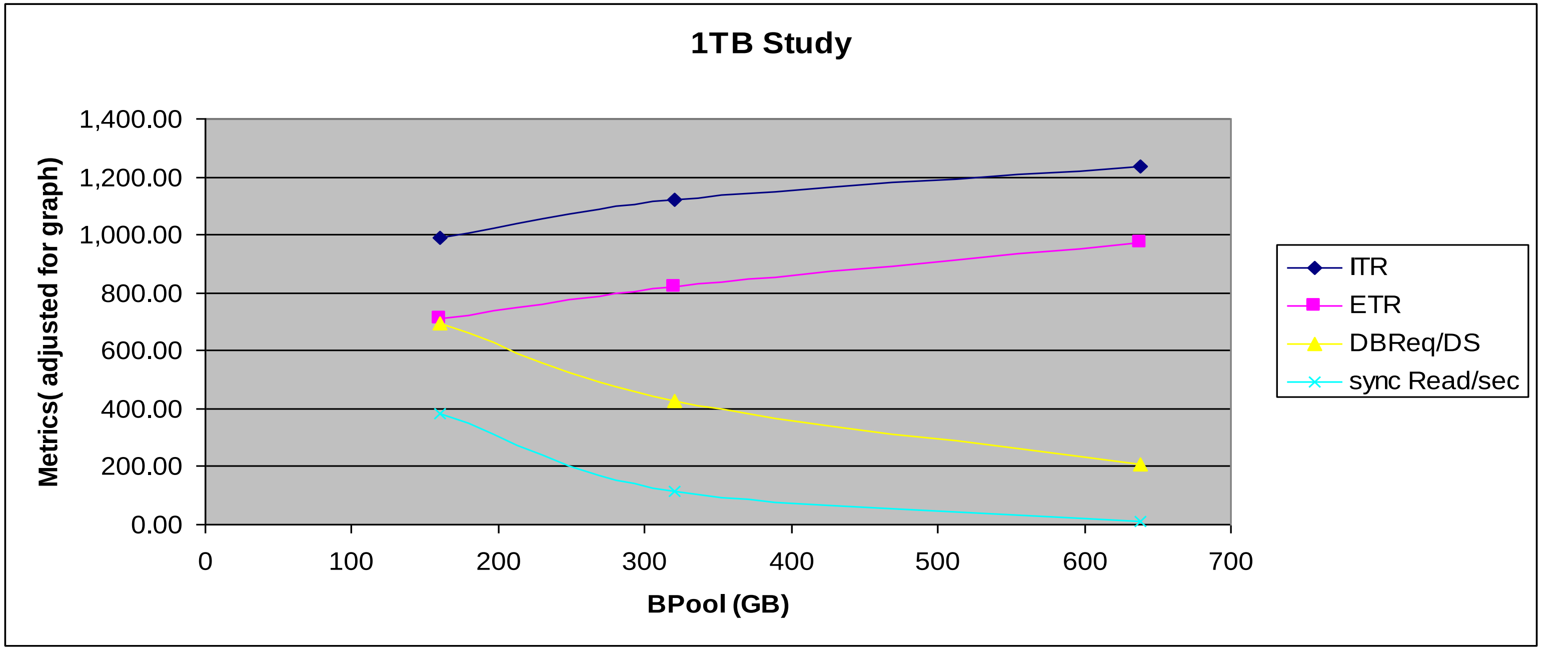
Figure 4: Buffer Pool Scaling Effects on DB Request Time

www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102461

© 2015 IBM Corporation

SSI : Online banking workload 12w DB2 V11 z/OS1.13

Memory	BP Size	CPU %	ITR	ITR Delta	ETR	ETR Delta	Txn response time(sec)	Response time delta	Sync Read IO/sec	Sync IO delta
256 GB	160 GB	72	992	n/a	709	n/a	.695	n/a	38.4k	n/a
512 GB	320 GB	73	1124	13.3%	819	15.5%	.428	-38%	11.7k	-69%
1024 GB	638 GB	79	1237	24.7%	976	37.7%	.209	-70%	0.9k	-97%



© 2014 IBM Corporation

Buffer Pool Simulation

- Simulation provides accurate benefit of increasing buffer pool size from production environment
- -ALTER BUFFERPOOL command will support
  - SPSIZE (simulated pool size)
  - SPSEQT (sequential threshold for simulated pool)
- -DISPLAY BPOOL DETAIL and Statistics Trace will include
  - # Sync and Async DASD I/Os that could have been avoided
  - Sync I/O delay that could have avoided
- Cost of simulation
  - CPU cost: approximate 1-2% per buffer pool
  - Real storage cost: approximate 2% of simulating pool size for 4K pages (1% for 8K, so on…)
    - For example, to simulate SPSIZE(1000K) 4K pools requires approx. 78MB additional real storage
- V11 APAR PI22091 for Buffer Pool Simulation now available

© 2015 IBM Corporation

CPU reduction from IO Avoidance

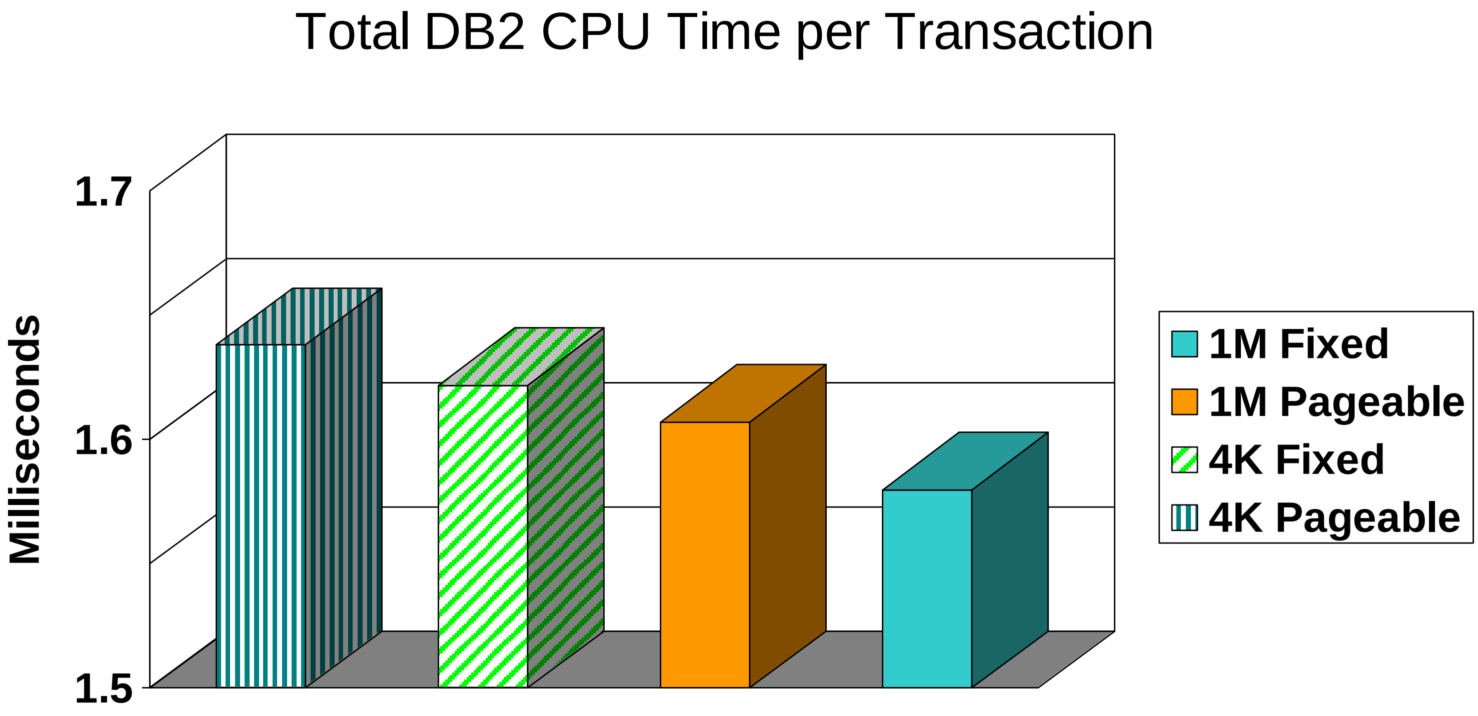
- We have measured very wide range between 20 usec to 70 usec
- CPU saving on zEC12 from the various workloads with steady 70-80% CPU utilization
- z13 is 5-10% better
- The variation depends on SQL workload and technical configuration
  - # of concurrent threads
  - Access pattern
  - Dedicated CPs
  - I/O saved came from GBP dependent getpage or not
- On z13, range is 20-40 usec

© 2015 IBM Corporation

Pageable 1MB Frames – Example from IBM Brokerage Workload

All of buffer pools are backed by real storage – DB2 10

- zEC12 16 CPs, 5000-6000 tps (simple to complex transactions)
  - 120GB real storage with 70GB LFAREA configured for 1MB measurements
- 1MB Pageable frames are 2% better than 4KB pageable frames for this workload
  - 70GB buffer pools are used, 8-10 sync I/O per transaction
- 1MB frames with PageFixed is the best performer in general



IMS and Large Memory Benefits

- **IMS**
  - –**Page fix buffers** reduce I/O delays up to 3% CPU
  - –Exploit IMS 12 ability to dynamically resize database buffer pools
    - Use IMS Buffer Pool Analyzer to view buffers by total buffer life.
  - –IMS program specification block (PSB) pool with large, infrequently used PSBs.
  - –IMS V12 large memory for IMS log buffers to improve online logging throughput.
  - –Dynamic database back out. Larger real memory allows the read process to be successful more frequently reducing the need for batch back-out.

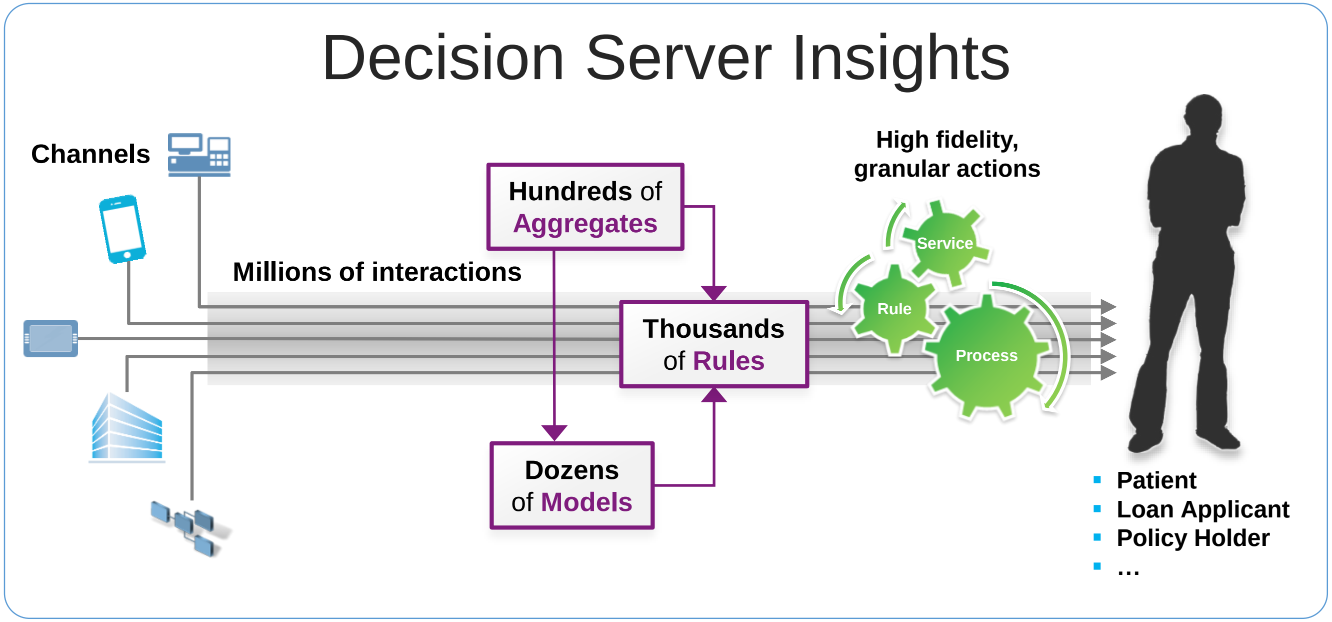
© 2015 IBM Corporation

Outline

- Large Memory Customer Value
- Example of Large Memory Benefits you can Leverage Today
- **Large Memory and Analytic Workloads**
- z/OS Potential Items for future Memory Management Enhancements

Optimize Your Business Decisions at the Time of Interaction

A new key component in ODM



Decision Server Insights is all about combining business rules, events, and predictive and real-time analytics into a single platform. It is an integrated, easy to operate, elastic platform for detecting events, patterns, and situations; updating the context; and pushing out actions---all at the same time.

Combines events, rules and predictive analytics to detect Risks and Opportunities at the time interaction.

© 2015 IBM Corporation

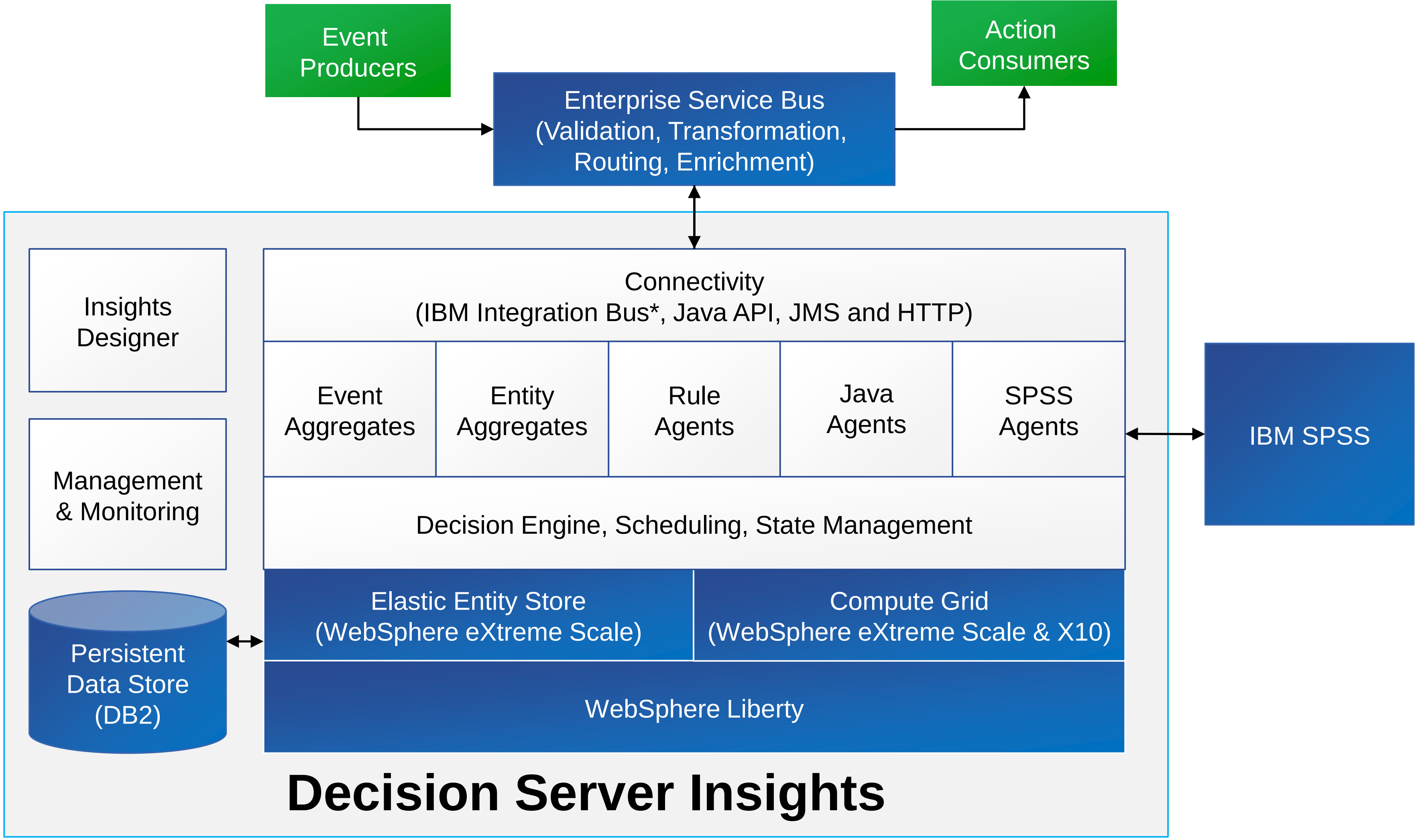
ODM Decision Server Insights

- **Why z/OS?**
  - High-performance, scale-up/scale out architecture using an in-memory compute and data grid.
  - Data analyzed at its source minimizing data movement and maximizing performance
  - Trends and patterns can be monitored reliably over extended periods
  - Leverage our superior synchronous and asynchronous replication for disaster recovery and back up
  - Fast Cache/DB coherency by exploiting our clustering technology

© 2015 IBM Corporation

High Level Architecture

Integrating business rules, events, predictive analytics capabilities in a single platform



Event Producers

Action Consumers

Enterprise Service Bus (Validation, Transformation, Routing, Enrichment)

Insights Designer

Management & Monitoring

Persistent Data Store (DB2)

Connectivity (IBM Integration Bus*, Java API, JMS and HTTP)

Event Aggregates

Entity Aggregates

Rule Agents

Java Agents

SPSS Agents

IBM SPSS

Decision Engine, Scheduling, State Management

Elastic Entity Store (WebSphere eXtreme Scale)

Compute Grid (WebSphere eXtreme Scale & X10)

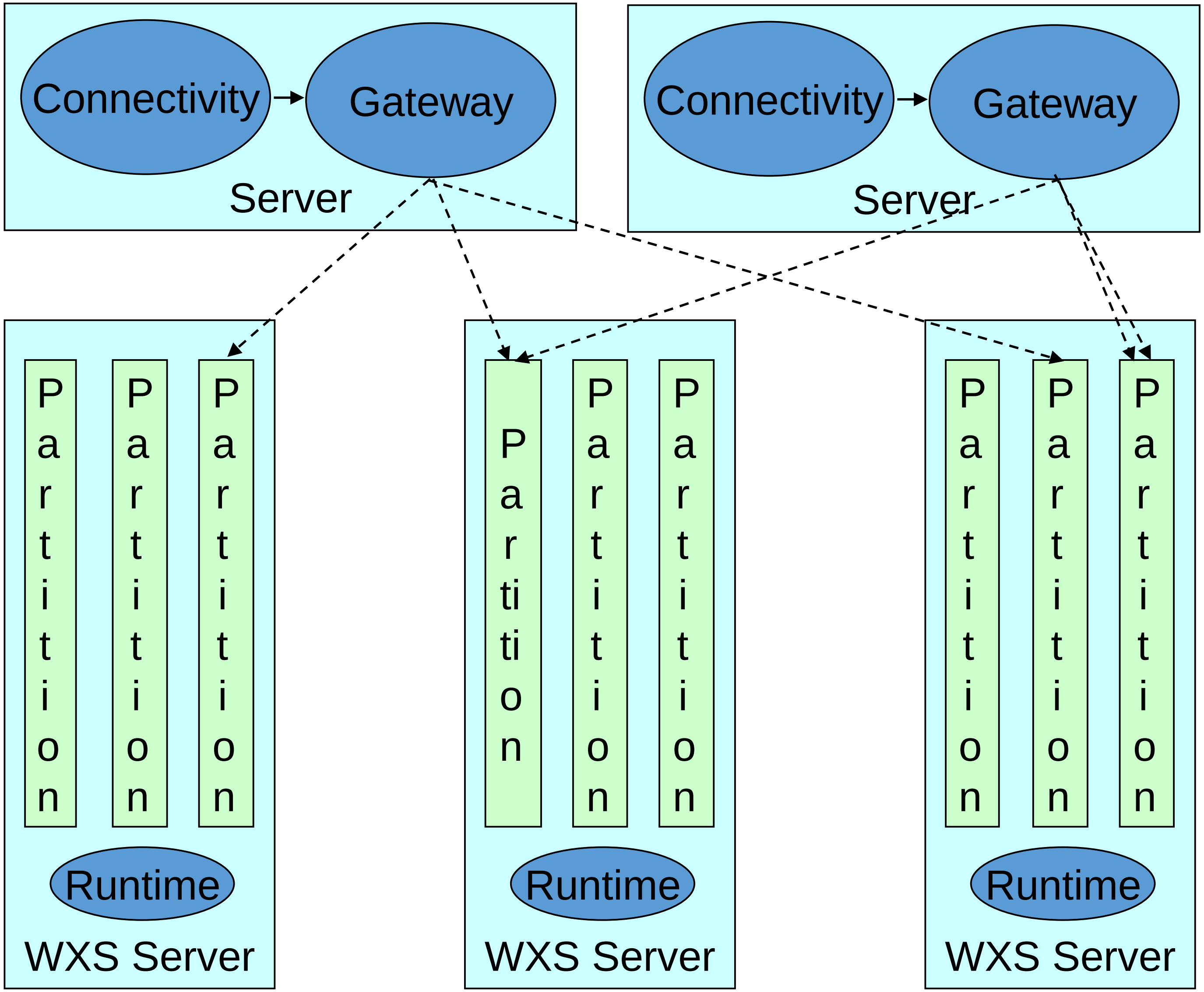
WebSphere Liberty

Decision Server Insights

*IBM Integration Bus is included as a Supporting Program, which can only be used for development and test purposes.

© 2015 IBM Corporation

Elastic, Highly Available Cluster



© 2015 IBM Corporation

Linux Example

- A modeling experiment to predict patient response to treatment required large amounts of memory
- Having 150 GB memory helped reduce the need to partition work and use ETL
- Using a few large data passes vs smaller computations with remerged results, produced a more accurate and faster solution
- Having more data in memory enabled more complex analysis- 33,500 rows of data were read in at once to enable complex analysis
- Models requiring weeks to run on x86 systems were reduced to hours running on Linux® on z Systems using 150 GB memory
- Avoid ETL which can consume additional overhead
- If this value is achievable with only 150 GB***, what are the possibilities with more memory?***

Memory was a Contributor to Efficient Analysis

Rheumatoid Arthritis Responder DREAM Challenge
Sage Bionetworks
DREAM
CORRONA
ARTHRITIS FOUNDATION
Take Control. We Can Help.
MERCK
Be well
novo nordisk

Large memory can deliver significant performance improvements for Linux Workloads

- Team used 150 GB on a IBM zEnterprise® EC12 (zEC12) data cloud.
- Less memory would mean smaller and fewer tests, and potentially fewer opportunities for analysis

Large Memory Feedback Questions

EXPECTED USE

- Do you best envision memory use for performance, in memory tables, availability or spikes?
- Can you envision use cases for 25 TB, 50 TB, or more?
- We delivered 10 TB memory with z13; how might you use it?
  - Do nothing different – use it for tuning
  - Make changes to existing workloads to use the new memory. (e.g. Larger DB2 buffer pools)
  - Add new workloads that would use additional memory
  - **How long does it take you to deploy and use large memory?**
  - **What is your timeline to production?**

VALUE

- What *business value* does large memory have (value of faster DB2 transactions, etc.)?
- Do you have examples of improved availability when using more memory in your shop?
- How do you validate the benefits of extra memory in your environment?

OPERATIONS

- Are there any operational challenges you see to using more memory?
- Are there any inhibiters to experimenting with large memory? How do the applications ask for it?
- Do you have specific tooling needs?
- What tools do you use for *memory* capacity planning and tuning? Do you tune for
  - Performance, In memory tables, Availability, Spikes

Summary

- **Large Memory has a large number of benefits including:**
  - Improving user transaction response times and increasing overall throughput for OLTP workloads
  - Enabling faster real time data analysis for Analytic workloads by reducing the time it takes to get from raw data to business insight
  - Processing Big Data more efficiently by increasing the speed at which large amounts of data can be scanned
  - Simplifying the deployment of scalable applications within cloud infrastructures

© 2015 IBM Corporation

Outline

- Large Memory Customer Value
- Example of Large Memory Benefits you can Leverage Today
- Large Memory and Analytic Workloads
- **z/OS Potential Items for future Memory Management Enhancements**

Outline

- Reconfiguration
- DB2 DISCARDDATA Pages
- SVC Dump

© 2014 IBM Corporation

Real Storage Reconfiguration

- Plans in next release of z/OS to allow more LFAREA when storage is brought online
- Likewise to decrease LFAREA when taking storage offline

© 2014 IBM Corporation

Reporting on DB2 DISCARDDATA Pages

- Working with DB2 on a new parameter for IARV64 REQUEST=COUNTPAGES, DISCARDPAGES={NO|YES} that will return the number of pages in the range that have been discarded with keepreal=yes.
- Processing to identify a discarded page requires the use of a special instruction that is CPU intensive. This type of counting cannot be done frequently.
- Evaluating the performance cost and the value to DB2 statistic reporting

© 2014 IBM Corporation

SVC Dump

- Plans in next release to optionally reserve real memory for SVC Dump usage
- Considering reserving real for system use in next release
  - Such as SQA,XCF, etc when there is a critical storage shortage

© 2014 IBM Corporation

Learn More

FW

DB2 memory white paper
www.ibm.com/support/techdocs/atsmastr.nsf/5cb5ed706d254a8186256c71006d2e0a/292109d2cfbe681586257d07007903d7/$FILE/LargeMemoryOverview_v1.pdf

Advantages of Configuring more DB2 Buffer Pools
http://w3.ibm.com/support/techdocs/atsmastr.nsf/3af3af29ce1f19cf86256c7100727a9f/8c521707def5c03686257d07007903cd/$FILE/LargeMemoryOverview_v1.pdf

SAP memory with paper
www.ibm.com/support/techdocs/atsmastr.nsf/5cb5ed706d254a8186256c71006d2e0a/8a1c8a3f19418bd386257d03005d051c/$FILE/Large_Memory_withSAP.pdf

Thank You