Linux on System z
Performance Experiences with Databases

Session ID: 9292

Erich Amrehn, Eberhard Pasch

IBM Lab Boeblingen, Germany

Trademarks

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

DB2*
DB2 Connect
DB2 Universal Database
e-business logo
IBM*
IBM eServer
IBM logo*
Informix®

System z
Tivoli*
WebSphere*
z/VM*
zSeries*
z/OS*

ECKD
Enterprise Storage Server®
FICON
FICON Express
HiperSocket
OSA
OSA Express

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

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.

Agenda

- Best “practices”
  - Hardware
  - Setup
  - Linux
  - Database
  - Monitor your progress
  - Application
- Customer results

Think before you act!

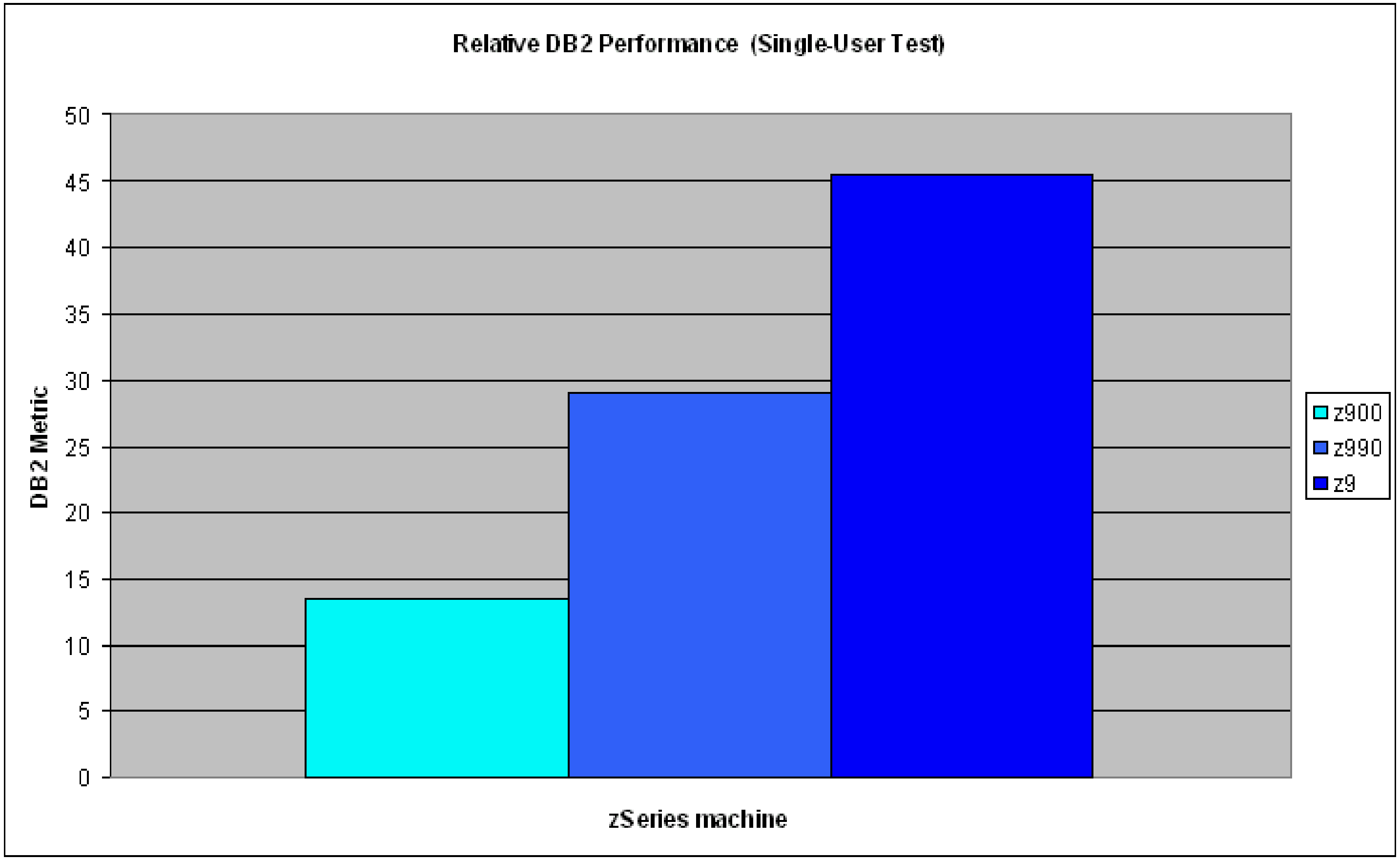
- What kind of database is this
  - Low / high utilized ?
  - Small / Medium / Large ?
  - Business critical / normal production / development / test ?

- Three categories for tuning
  - Just run
  - Apply basic best practices
  - In depth tuning and analysis

Bigger hardware is better – newer software as well

- Storage subsystem
  - Faster spinning disks better than slower disks
  - More small disks better than a few large ones
  - More cache (read) and non volatile storage (write)
  - More control units
  - More cables & paths
- System z
  - Faster is better
- Database upgrades
  - For DB2, Informix and Oracle we've seen release to release improvements that have been significant for specific workloads

A simple test on 3 different generations of machines

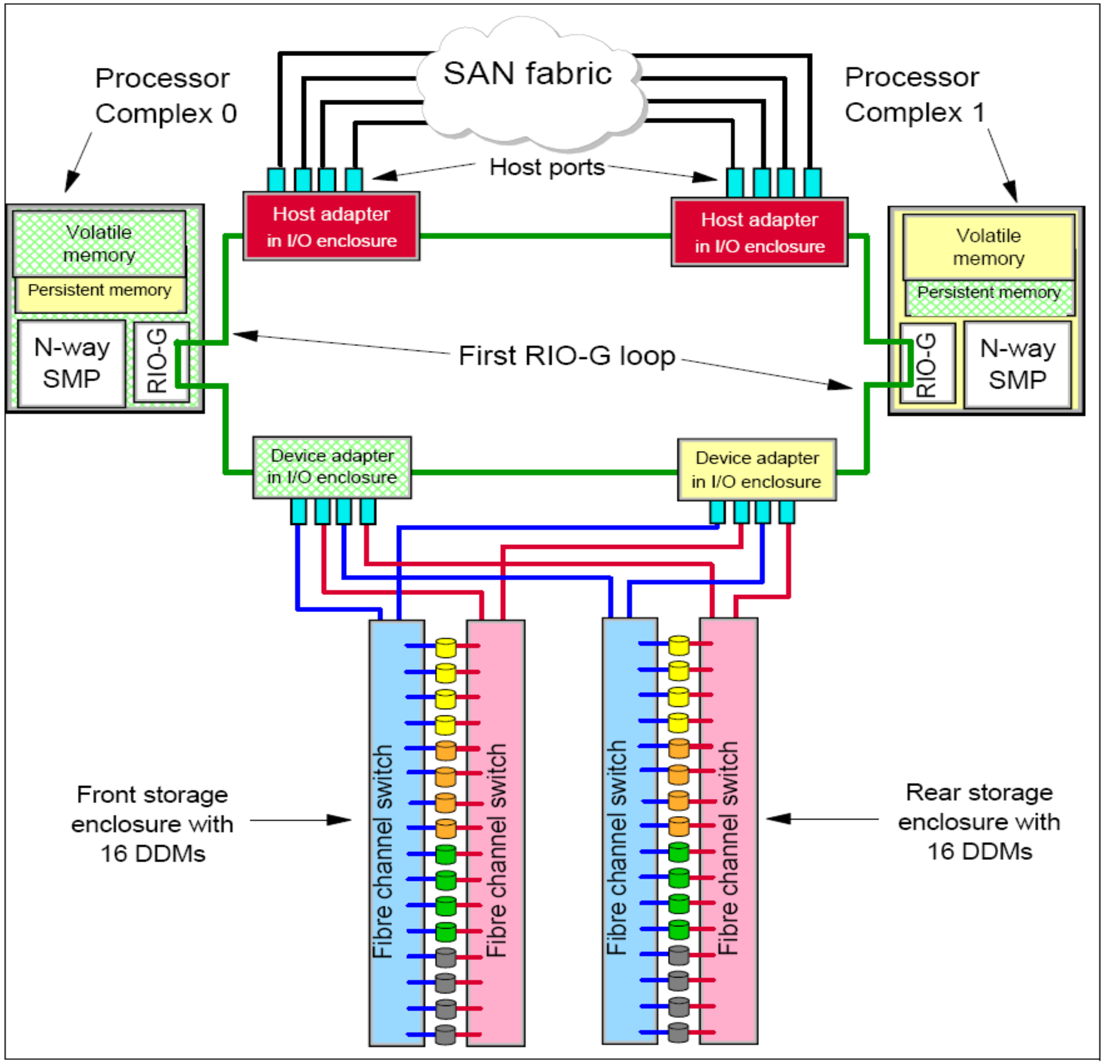


- DB2 metric: z900 to z990 = 2.2x → z990 to z9 = 1.6x
- Clock speed: z900 to z990 = 1.6x → z990 to z9 = 1.4x

(DS8000) Disk setup

- Don't treat a storage server as a black box, understand its structure
- Enable storage pool striping if available
- Principles apply to other storage vendor products as well
- You ask for 16 disks and your system administrator gives you addresses 5100-510F
  - From a performance perspective this is close to the worst case
- So - what's wrong with that?

DS8000 Architecture



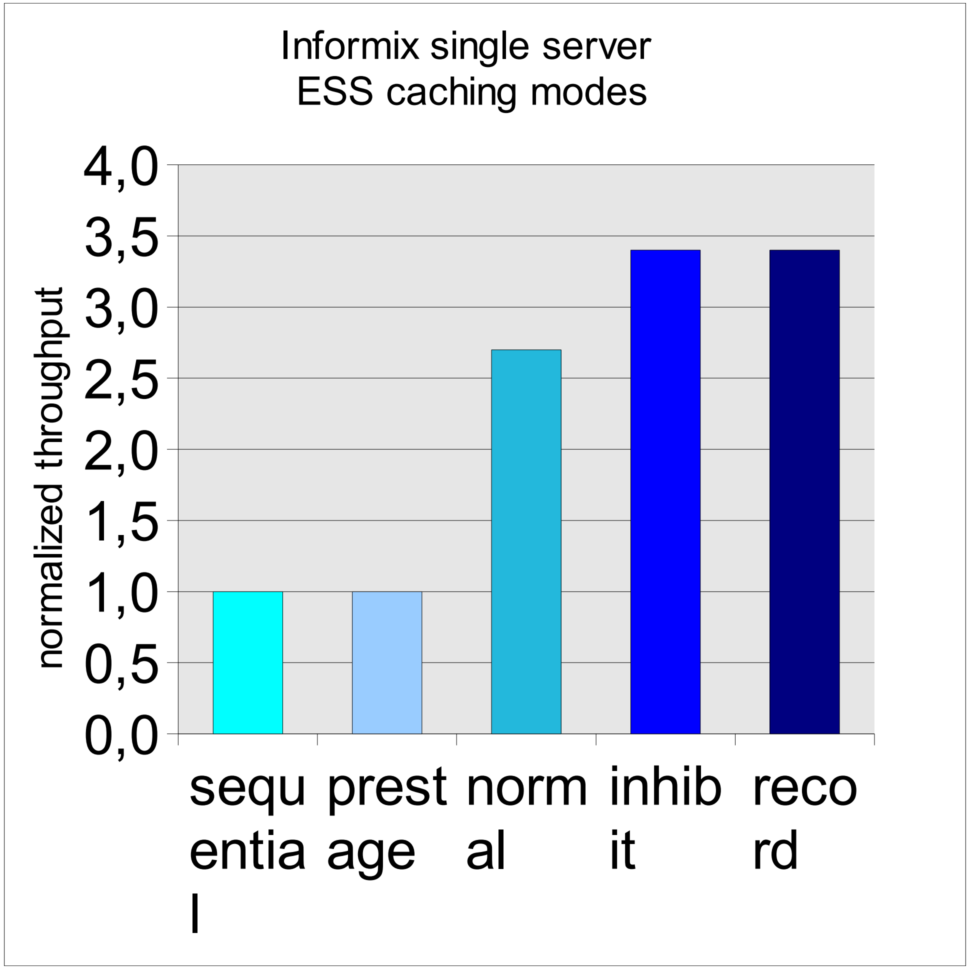
- **structure** is complex
  - disks are connected via two internal FCP switches for higher bandwidth
- the DS8000 is still divided into two parts named **processor complex** or just **server**
  - caches are organized per server
- one **device adapter pair** addresses 4 array sites
- one **array site** is build from 8 disks
  - disks are distributed over the front and rear storage enclosures
  - have the same color in the chart
- one **RAID array** is defined using one array site
- one **rank** is built using one RAID array
- ranks are assigned to an **extent pool**
- extent pools are assigned to **one of the servers**
  - this assigns also the caches
- **the rules are the same as for ESS**
  - one disk range resides in one extent pool

Rules for selecting disks

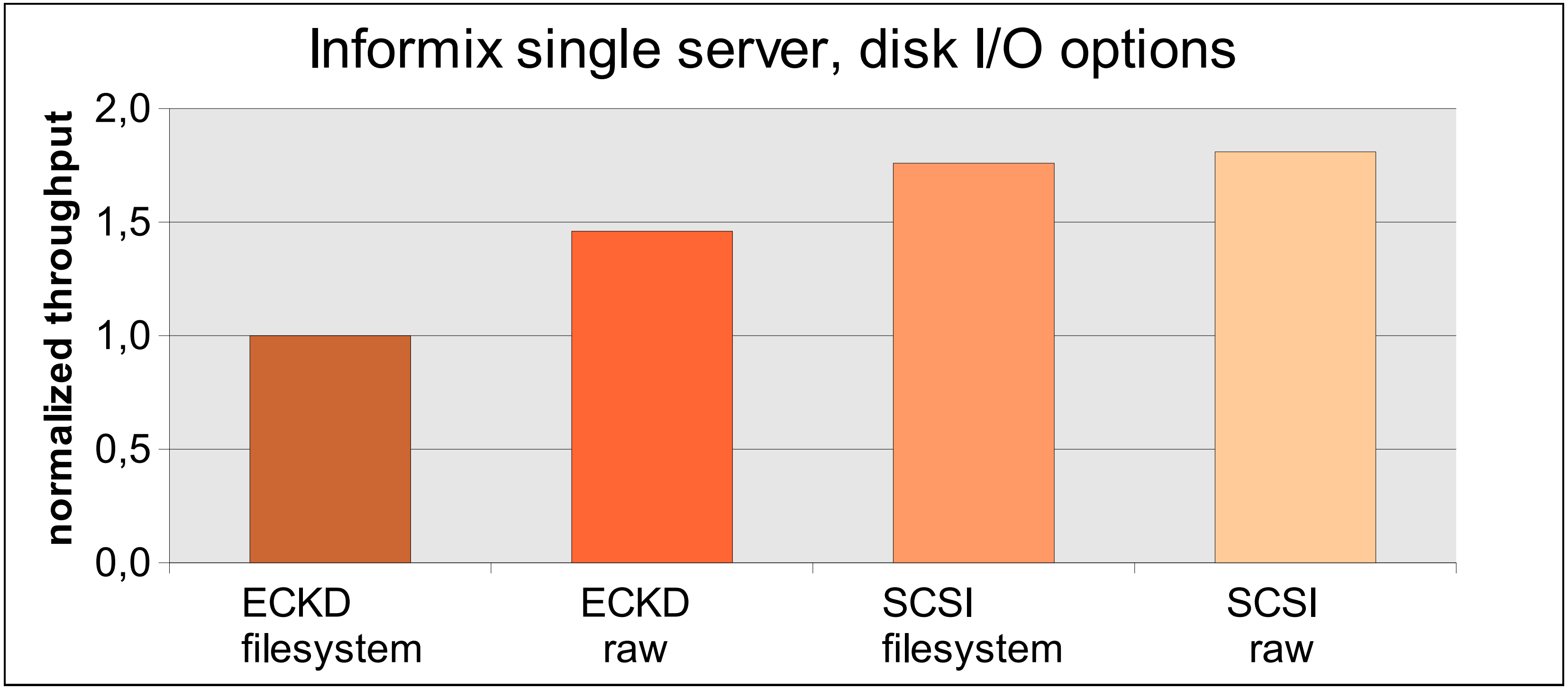
- **goal is to get a balanced load on all paths and physical disks**
- use as many paths as possible (CHPID -> host adapter)
  - for ECKD switching the paths is done automatically
  - FCP needs a fixed relation between disk and path
    - we establish a fixed mapping between path and rank in our environment
    - taking a disk from another rank will then use another path
- switch the rank for each new disk in an LVM
- switch the ranks used between servers and device adapters
- select disks from as many ranks as possible!
- avoid reusing the same resource (path, server, device adapter, and disk) as long as possible

ESS Cache Modes

- The caching mode “record” returns the best result.
- Caching modes are described in
  - Command Reference 2105 Models SC26-7298-xx
- The caching mode can be changed with the command "tunedasd"
- On DS6000 and DS8000 mode “SARC” (simplified adaptive replacement cache) divides cache between random and sequential classes



Disk I/O options

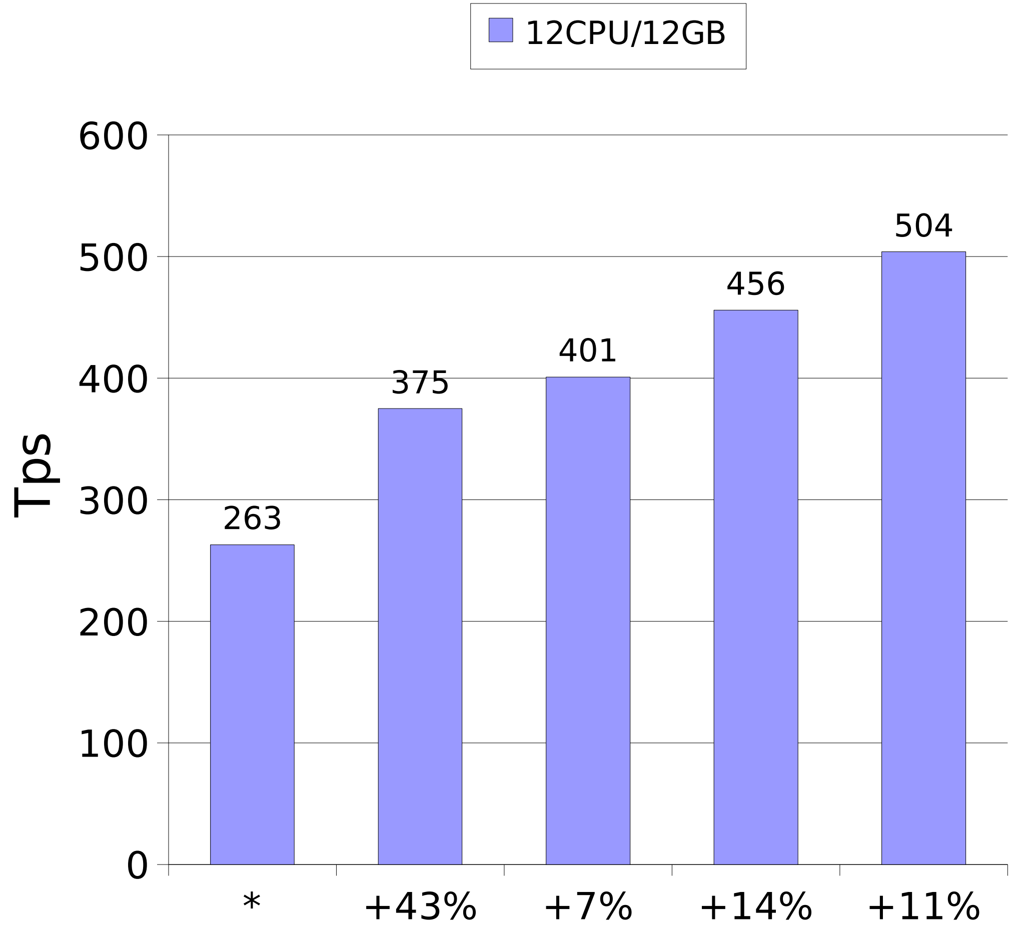


- File system type is ext2
- Best options are SCSI file system and ECKD raw devices
- SCSI file system was used for all following scaling tests with Informix

Read-ahead setup

- Database
  - Disable it by setting e.g. in Informix the onconfig parameters RA_PAGES and RA_THRESHOLD to 0
- LVM
  - Disable it by setting the read ahead to 0 pages with the command lvchange -r 0 /dev/<volume group>/<logical volume>
- Linux block device layer
  - Set the value to 0 using the blockdev command,
    for example: blockdev --setra 0 /dev/sda

DB2 8.2 Tuning Milestones



* starting point

+43%

tablespace prefetch 0

LVM readahead 0

+7%

CHNGPGS_THRESH from 30 to 60

+14%

extra bufferpools (data and index) for customer tablespace

+11%

pagesize 8K for customer index tablespace/bufferpool

Kernel parameters (1) – shared memory

- Kernel parameter changes should be configured in /etc/sysctl.conf
- Recommendations here for DB2, Oracle and Informix have similar ones
- Shared memory kernel parameters:
  - **kernel.shmall**: Available memory for shared memory in 4 K pages
  - **kernel.shmmax**: Maximum size of one shared memory segment in byte
  - **kernel.shmmni**: Maximum number of shared segments
- Shared memory is used for the buffer pools, so adaption might be needed to you specific DB workload

Linux memory	shmall	shmmni	shmmax
2 GB	400000	4096	1700000000
4 GB	912600	4096	3774873600
8 GB	1971200	4096	8074035200
12 GB	3020800	4096	12373196800
16 GB	4070400	4096	16672358400
20 GB	5120000	4096	20971520000
24 GB	6169600	4096	25270681600

More kernel parameters (2) – semaphores limits

- Kernel semaphores limits
  - The kernel semaphores limits were adapted according to the DB2 recommendations.
  - kernel.sem: Max. semaphores per array / max. Semaphores system wide / max. ops per per semop call / max. number of arrays

Kernel parameter	default	used in tests	usage
kernel.sem	250 32000 32 128	250 256000 32 1024	semaphore settings

More kernel parameters (3) – message limits

- Kernel message limits
  - The kernel message limits were adapted according to the DB2 recommendations.
  - kernel.msgmni: Maximum queues system wide
  - kernel.msgmax: Maximum size of message (bytes)
  - kernel.msgmnb: Default size of queue (bytes)

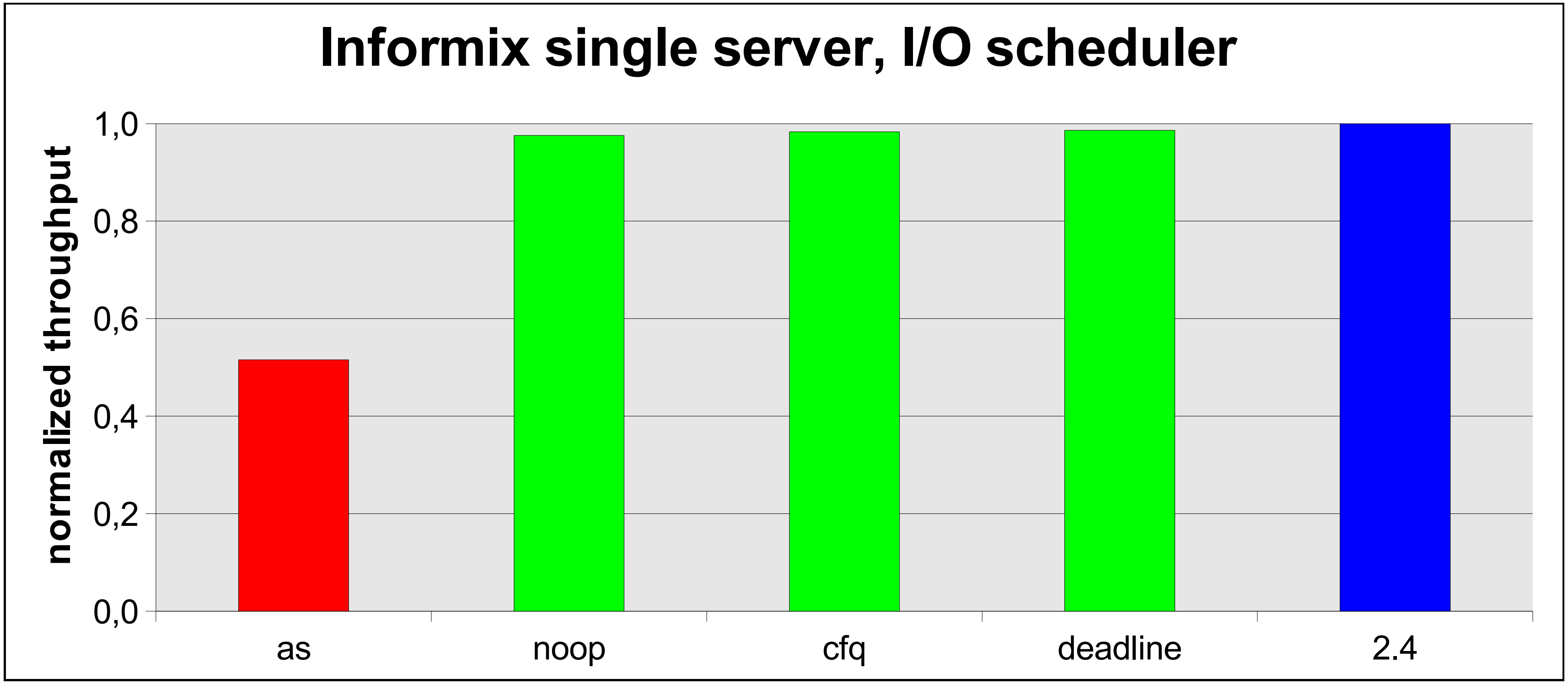
Kernel parameter	default	used in tests
kernel.msgmni	16	1024
kernel.msgmax	8192	65536
kernel.msgmnb	16384	65536

- All kernel parameters take effect after reboot
- Kernel parameter settings may be inspected in proc file system with cat /proc/sys/kernel/<parameter name>

Linux 2.6 I/O Schedulers

- Four different I/O schedulers are now available
  - noop scheduler
    only request merging
  - deadline scheduler
    avoids read request starvation
  - anticipatory scheduler (as scheduler)
    designed for the usage with physical disks, not intended for storage subsystems
  - complete fair queuing scheduler (cfq scheduler)
    all users of a particular drive would be able to execute about the same number of I/O requests over a given time.

Linux 2.6 I/O Schedulers - Results



- “as” scheduler is not a good choice for this environment
- All other schedulers show similar results as the kernel 2.4 scheduling

What to do with log files?

- Database data access is random I/O, writing a log is sequential I/O

- If database and log files are on the same disk
  - the sequential characteristics of log I/O get lost
  - I/O schedulers start perferering reads
  - it degrades the transfer reade and the priority of log writes which results in a limited transaction rate

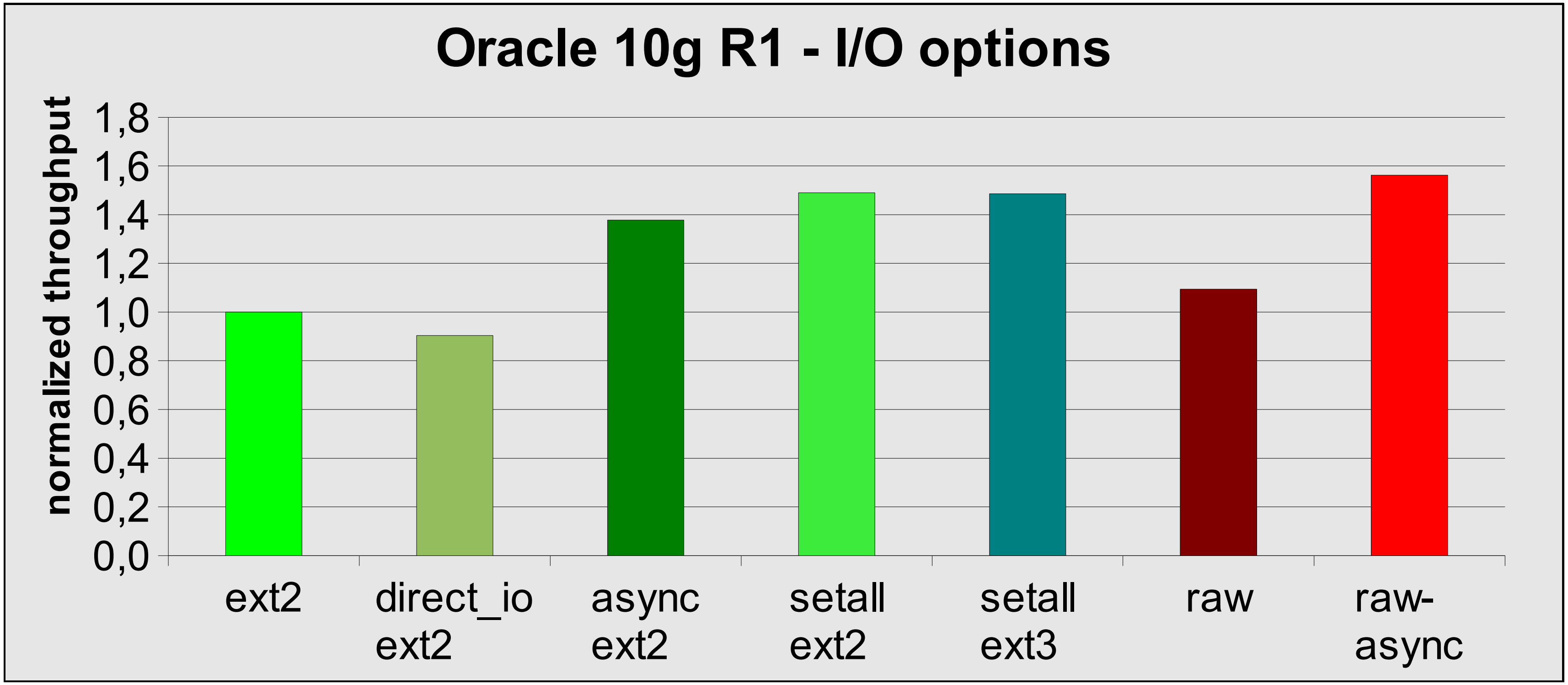
- Separate log and data devices, in the best case take
  - other ranks on the same storage server or
  - another storage server

- Especially important for workloads with many commits
  - Optimizing the log I/O is key for best performance

Linux 2.6 Disk I/O Options and Oracle 10g

- Direct I/O (DIO)
  - transfer the data directly from the application buffers to the device driver, avoids copying the data to the page cache
  - Advantages:
    - saves page cache memory and avoids caching the same data twice
    - enables larger buffer pools
  - Disadvantage:
    - make sure that no utility is working through the file system (page cache) --> danger of data corruption
- Asynchronous I/O (AIO)
  - The application is not blocked for the time of the I/O operation
  - It resumes its processing and gets notified when the I/O is completed.
  - Advantage
    - the issuer of a read/write operation is no longer waiting until the request finishes.
    - reduces the number of I/O processes (saves memory and CPU)
- Recommendation is to us both

Linux 2.6 Disk I/O Options - Results



- The combination of direct I/O and async I/O (setall) shows best results when using the Linux file system. Best throughput however was seen with raw I/O and async I/O.

- ext2 and ext3 lead to identical throughput

Linux 2.6 Disk I/O Options and DB2 UDB 9

- new I/O options supported with version 9 for:
  - async I/O
  - direct I/O – at least for SCSI disks (512 byte blocks)
  - use DMS containers, no LVM

Monitor your progress

- Database reporting tools work well on System z, e.g. AWR for Oracle
  - Watch out for buffer pools that are too small
- Monitor Linux OS as well (iostat, sadc)
  - Focus on IO
- Monitor z/VM
  - Database caches on z/VM paging space are not optimal for performance
- Correlate results
- Make one(!) change at a time

Applications with database calls

- Monitoring will show inefficient SQLs
  - Fix in application
  - Introduce new index
- Avoid many small database requests

Summary – best practices

- How much do you need/want to optimize
- Use up to date hardware and software
- Best practices
  - Distribute workload on storage server
  - Check readahead and adapt
  - Use the right Linux settings
  - Separate log IO from database IO
  - Use database caches
- Monitor your progress
- Optimize applications

Customer benchmark Agenda

- Background
- Goals
- Benchmark environment
- Benchmark plan
- Testing scenarios and results
- Conclusions

Background

- The benchmark was planned as a phase within the “Migración Backend NEWPROJECT”. This project’s primary goal is to assess the feasibility of the data migration and the transition to a new platform based on Oracle over Linux on System z.
- During the initial stages of the project the following milestones have been accomplished:
  - Scope and planning workshop. Intended to define in detail the phases and the tasks inside the project, and to output the state of work for the whole project.
  - Design workshop. Focused on building up the technical solution for the new platform.
  - Proof of Concept. It accomplished to validate, from a functional point of view, and to refine the initial technical solution.
- After the Proof of Concept phase the new platform was certified from an application and operational perspective, although their performance still had to be assessed.
- The benchmark should allow to right size the new platform configuration, in order to deliver the NEWPROJECT backend service with no degradation, from the user’s perception.
- In addition to those technical assessments, and as part of the “Migratión Backend NEWPROJECT” , the following deliverables will be output:
  - System Management Model
  - Implementation Plan (including the migration strategy)
  - Transition’s cost estimate

Agenda

- Background
- Goals
- Benchmark environment
- Benchmark plan
- Testing scenarios and results
- Conclusions

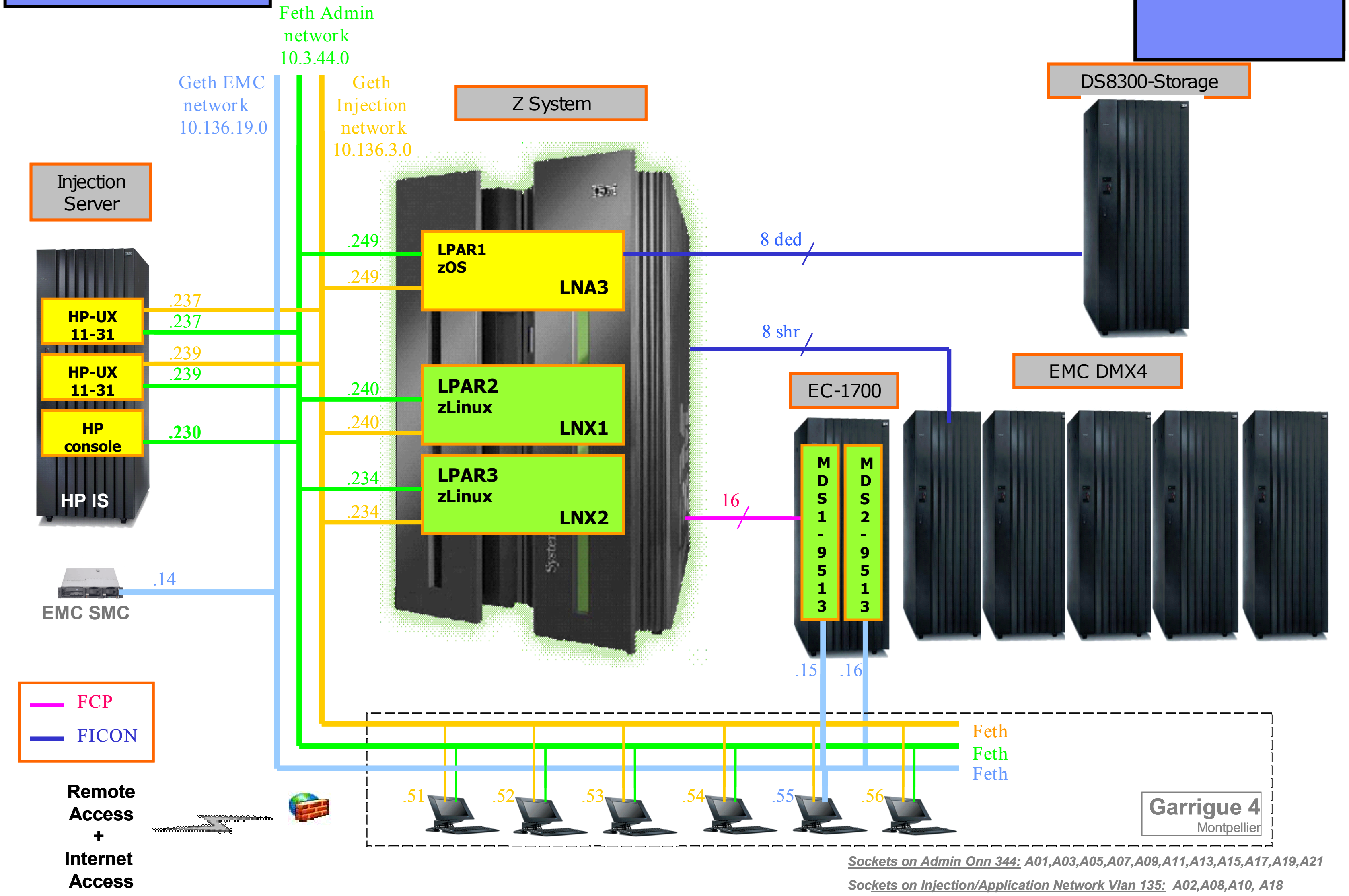
Goals

- The benchmark will evaluate the new platform, and will aim to achieve the following main goals:
  - To asses its general performance.
  - To asses its OLTP capability to meet customers current business needs.
  - To asses its Batch capability to meet customers current business needs.
  - To right size the system configuration.
  - To guarantee the migration process fits into the agreed window.
  - To guarantee the system will scale up according to increasing the business needs.
  - To evaluate the platform support level provided by IBM, Oracle and EMC.

Agenda

- Background
- Goals
- Benchmark environment
- Benchmark plan
- Testing scenarios and results
- Conclusions

Benchmark environment



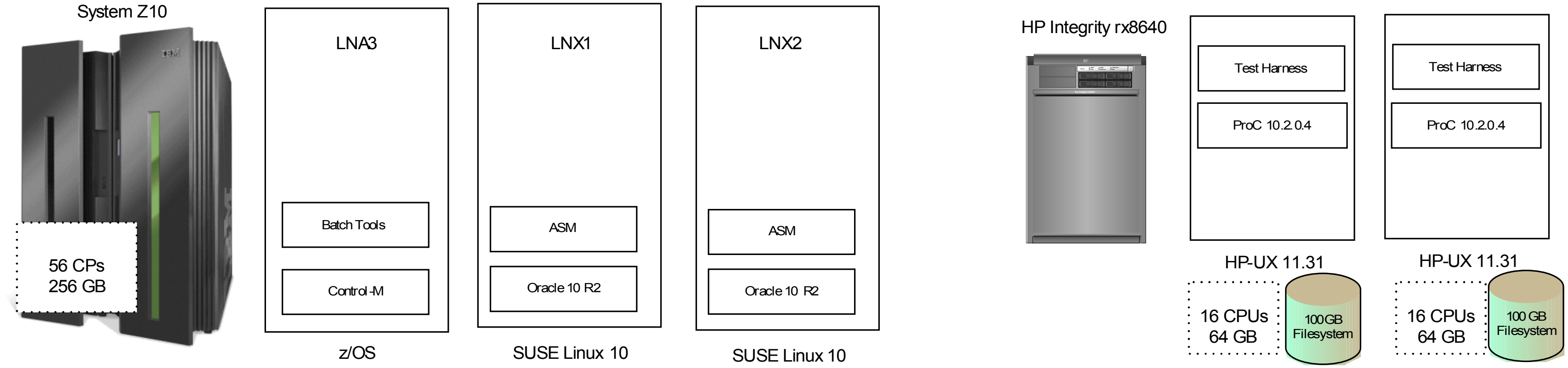
__Sockets on Admin Onn 344:__ A01,A03,A05,A07,A09,A11,A13,A15,A17,A19,A21

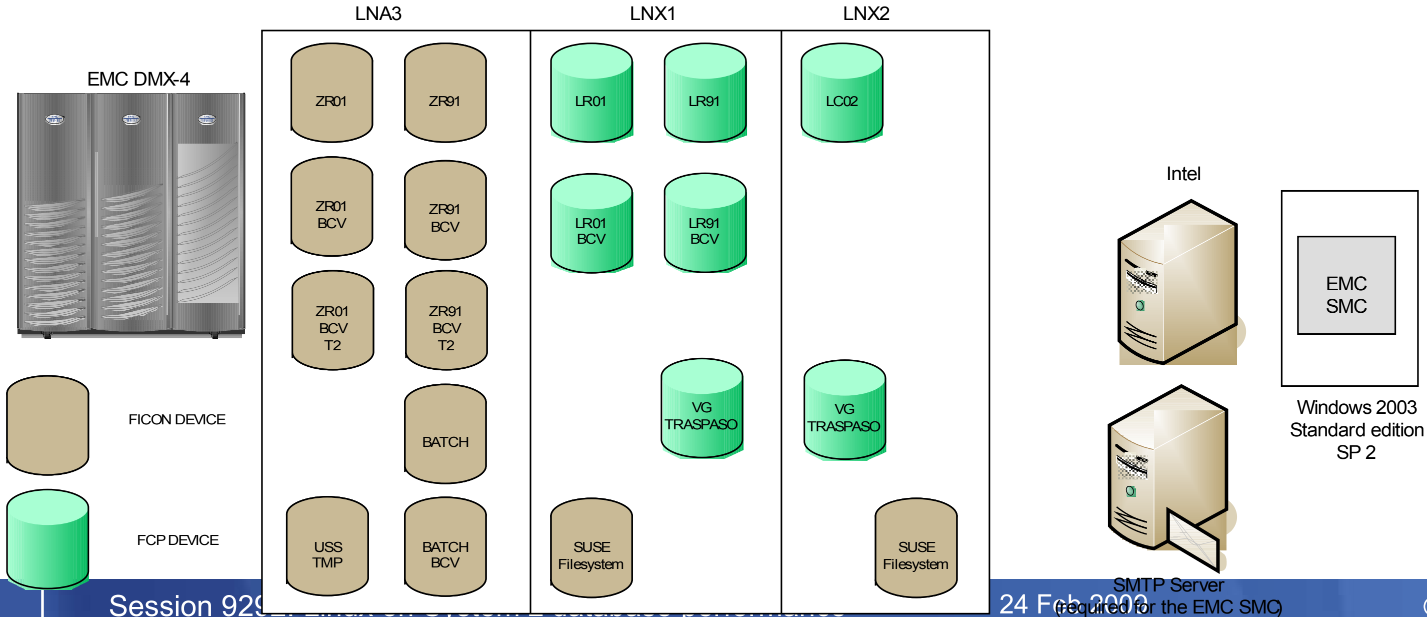
__Sockets on Injection/Application Network Vlan 135:__ A02,A08,A10, A18

__Sockets on Internet:__ A04,A06,A12,A14,A20,A22

Socketson EMC Admin network: A16

IBM Benchmark environment





Agenda

- Background
- Goals
- Benchmark environment
- Benchmark plan
- Testing scenarios and results
- Conclusions

Benchmark plan

Testing Scenario	Description	Goal	Succes Criteria	Duration
Performace	Simulation and comparison of real on-line transaction load (more than 1 million transactions) on the Linux on z and on the z/OS environments. The transaction load simulation will be accomplished by the execution of the Test Harness.	Certify that Linux on z will meet performance requirements for OLTP	Meet or exceed current performance requirements (for at least 95% of the total tested transaction executions)	12 days
Batch	Execution of end of the month batch. The batch process will be hosted on the z/OS partition and will hit, through hipersockets, both Linux on z databases.	Certify that Linux on z will meet performance requirements for batch process	End of the month batch process window execution is equal to current window execution (3-4 days)	21 days
Scalability	Simulation of real and stressed transaction load. The transaction load simulation will be accomplished by the execution of the Test Harness.	Certify that Linux on z will scale according to the transaction load growth	Constant path length for a workload increase of the 50%	7 days
Database migration	Database migration from z/OS to Linux on z	Certify that the database migration process execution will fit into the agreed unavailabity period	Migration completed in less than 40h	19 days

Agenda

- Background
- Goals
- Benchmark environment
- Benchmark plan
- Testing scenarios and results
- Conclusions

Testing scenarios and results - Performance

Constraints

- Real transactional load corresponding to the one experienced in Customer-TX as of 09/10/2008.
- There would be a control environment, based on Oracle on z/OS, similar to the existing one in Customer-TX, this will serve to validate the test dataset consistency (registered transactions in production).
- The load injection would be done by the execution of an in-house tool, fully developed by Customer, which would simulate the behavior of the elements responsible for generating database workload, according to Customer's application architecture.

Goal

- The primary target is to assess the capability of the new platform to support the OLTP workload, on the same way the current platform does.
- The success criteria is preset on a full workload, having 95% of the injected transactions, whose response time is equal or lower, compared to the same workload experienced in the production environment in Customer-TX. An error rate is accepted due to the benchmark environment limitations (non-complete environment basically).

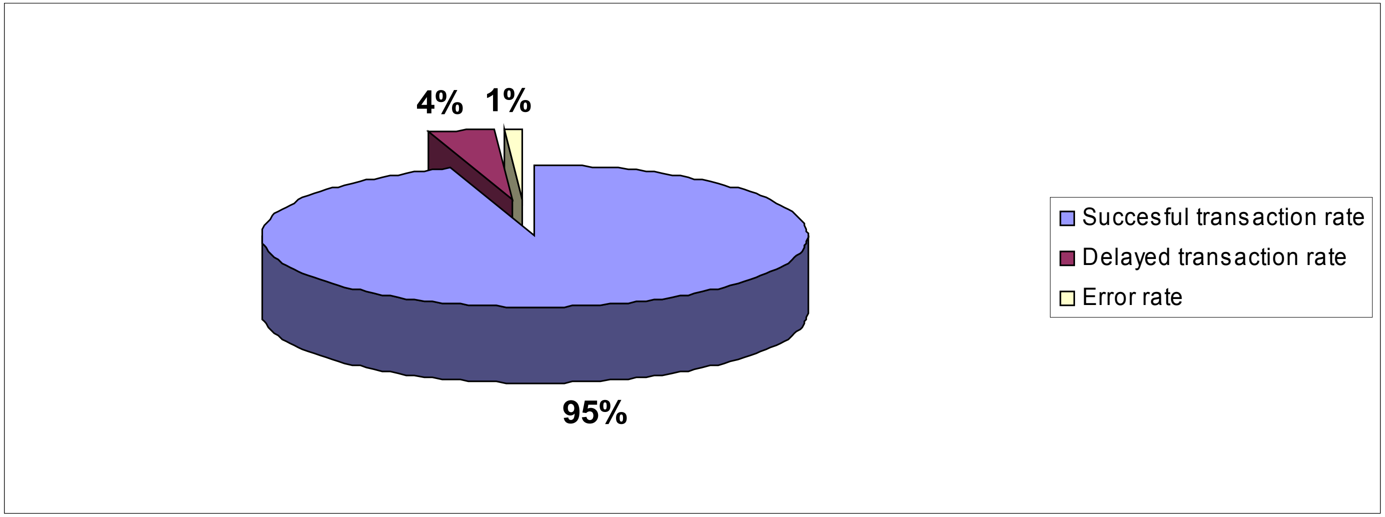
Testing scenarios and results - Performance

Results (1)

- The tests were finished in 12 days, according to the initial plan. 56 test runs were executed and each one followed this testing methodology:
  1. Testing environment configuration and preparation.
  2. Workload injection, and environment monitoring.
  3. Injection results analysis (transactions response time), and study of system behavior.
  4. System configuration parameters fine-tune.
- The results obtained from the injection on the target environment (Linux on System z) were compared to those output from the control environment (z/OS), to verify the data consistency, and to discard those errors produced by the non complete environment.
- At the end of the tests, we achieved results, out of 1,180,161 injected transactions:
  - 95% success transactions rate (equal or lower response time compared to production)
  - 4% delayed transactions (two decimal places, in milliseconds, accuracy)
  - 1% of the transactions returned an unexpected error.

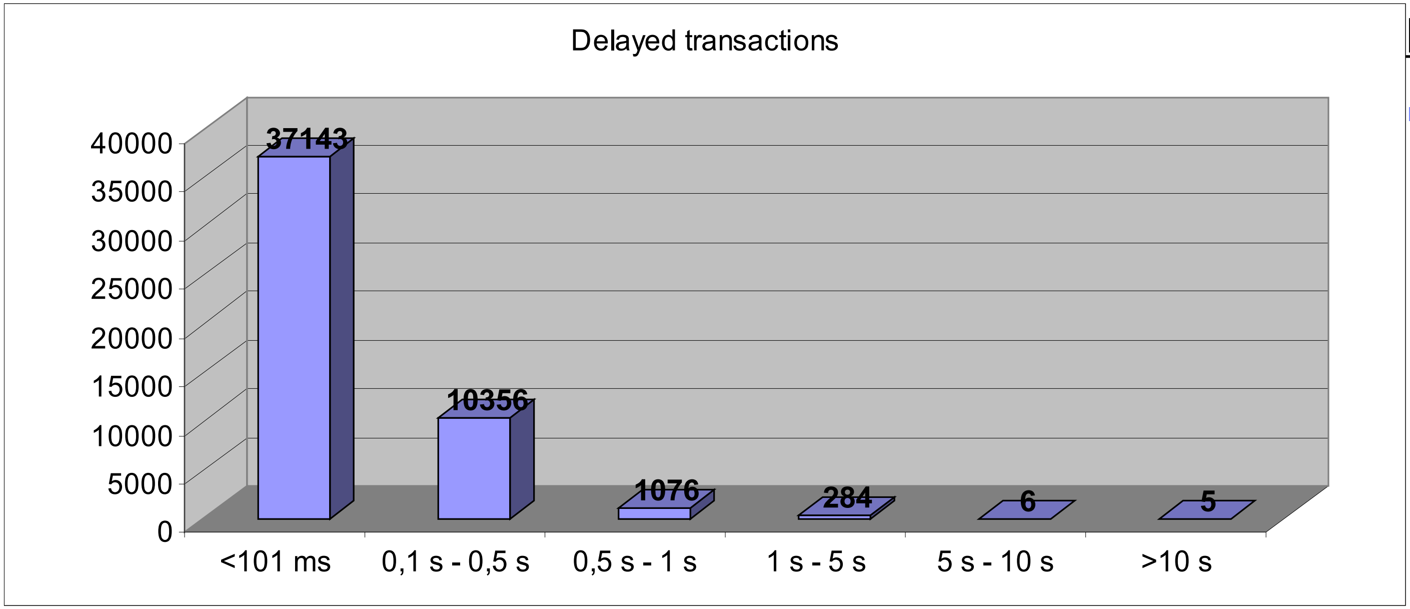
Testing scenarios and results - Performance

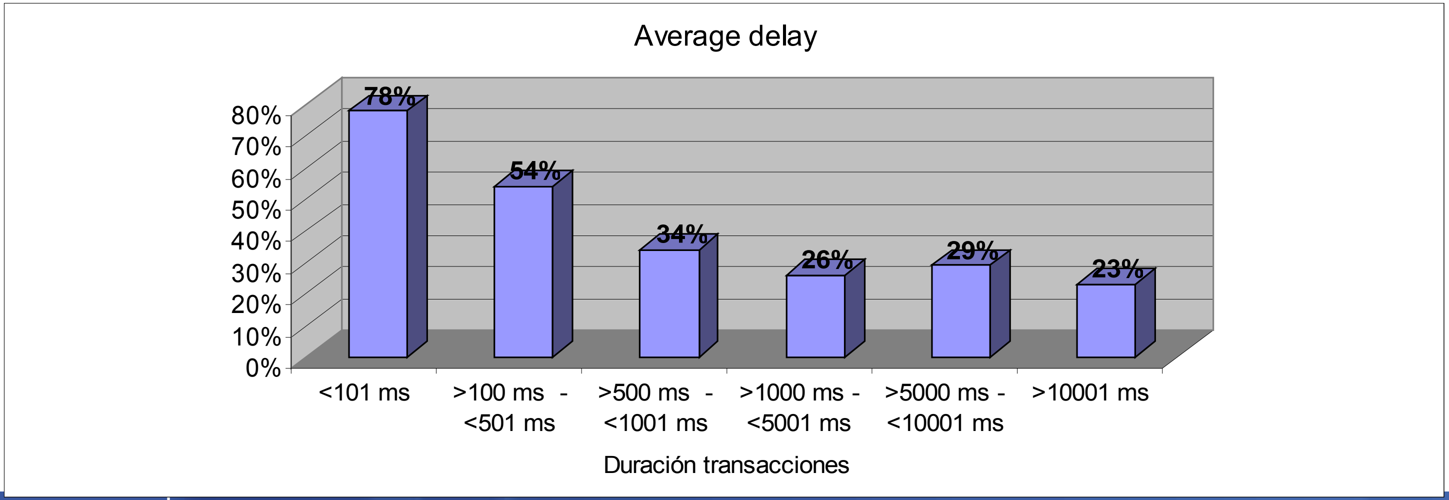
Results(2)



- A further analysis yields these conclusions:
  - There's a 50% of global response time improvement (the global response time represents the consolidated response time of all the injected transactions).
  - The rate of unexpected errors (1%) is caused by two main reasons:
    - Simultaneous transactions. The execution order is not registered for these, so every test run will output a different value.
    - Internal rowid usage. This id is inherent to the database. It will be different with each database, so will output different values.
  - Delayed transactions (4%) should be part of an in depth study, in order to determine their impact on the user's perception of the service.

Testing scenarios and results - Performance





Results(3)

- After studying the delayed transactions, we can conclude the following:
  - More than 97% of the delays are lower than 0,5 s.
  - Only 0,68% of the delays are higher than 1 s.
  - The mean delay gets decreased as the transaction duration gets increased. The maximum mean delay (78%) is observed for the shortest transactions (duration shorter than 101 ms), whereas transactions with a duration longer than 1s have a mean delay around 23%-29%.

Testing scenarios and results - Batch

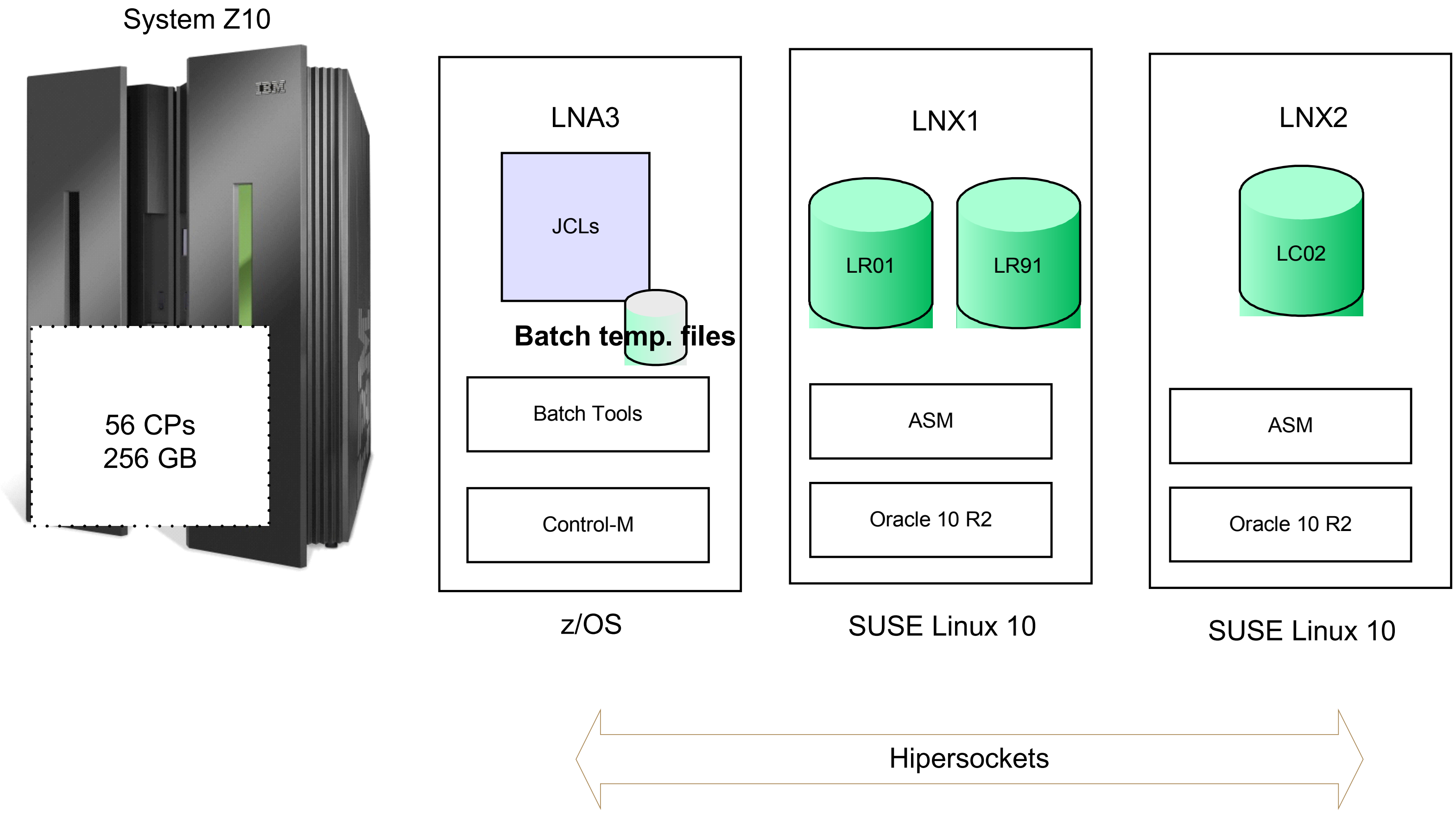
Constraints

- Batch testing should simulate the batch process execution experienced in the production environment in Customer-TX. The end of the month process made a good test case because o being business critical, and having a considerable duration.
- Due to the Benchmark environment limitations, some of the steps and jobs which required access to the complete environment were skipped. In addition to this, and for security reasons all the jobs which implied a physical copy to an external device were also discarded.
- The job execution had to be done and controlled by the same means used in Customer-TX.

Goal

- The primary goal is to asses the capability of the new platform to deliver the batch service whil maintaining the same service level for the most demanding (in terms of duration and quality of results) and critical processes (end of the month)
- The success criteria is preset to being able to execute these processes within the time window established in Customer-TX.

Testing scenarios and results - Batch



Testing scenarios and results - Batch

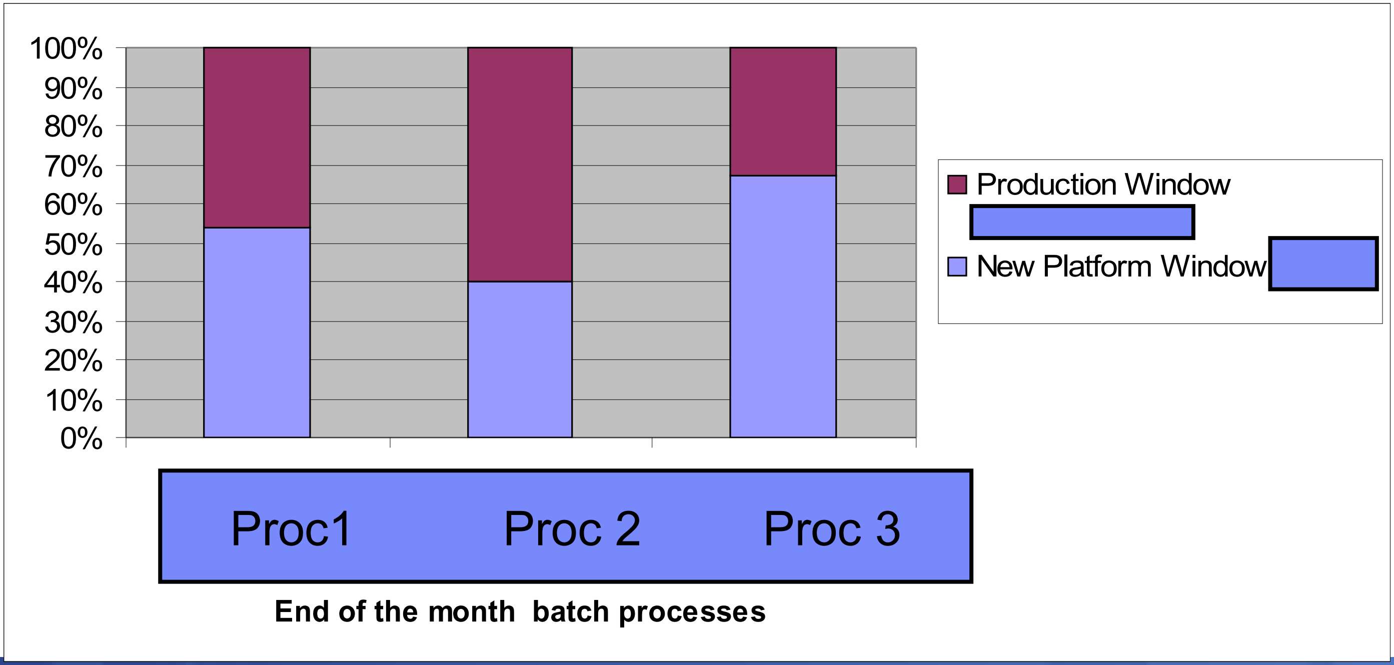
Results (1)

- The tests were finished in 10 days. During this period, the following testing methodology was applied to each test run:
  1. Testing environment configuration and preparation. It required the initial migration of the NEWPROJECT backend (data model, data, and PL/SQL code), and its recovery after each run. In addition, the jobs had to be prep to be executed without the satellite environment.
  2. Jobs execution and system monitoring. Jobs were scheduled and controlled for possible cancellations or wrong returned codes. At the same time the system hardware resources consumption was monitored and functional checks were run to validate the results.
  3. System fine-tuning. The hardware resources sizing determined the maximum capacity required for the z/OS partition, (will only be used by the batch processes), and the minimum for the Linux partition (the max size will be set by the more demanding OLTP workload).
- According to this methodology there were 3 full executions of customer defined processes .

Testing scenarios and results - Batch

Results(2)

- The batch execution has been fitted into the established window, and substantially reduced, as shown in the figure (new platform's window compared to production's window for each process)
  - Process 1 – done in 54%
  - Process 2 – done in 40%
  - Process 3 – done in 67%



Testing scenarios and results - Batch

Results(3)

- Below are shown the individual statistics for each process:

	Total number of compared jobs	Percentage over the total of existing jobs in the process	Jobs with a reduced elapsed time	Total improvement rate
Process 1	449	48%*	399	46%
Process 2	111	99%	104	60%
Process 3	97	97%	86	33%

* Only taken for consideration those jobs which are longer than 5 minutes

Testing scenarios and results - Migration

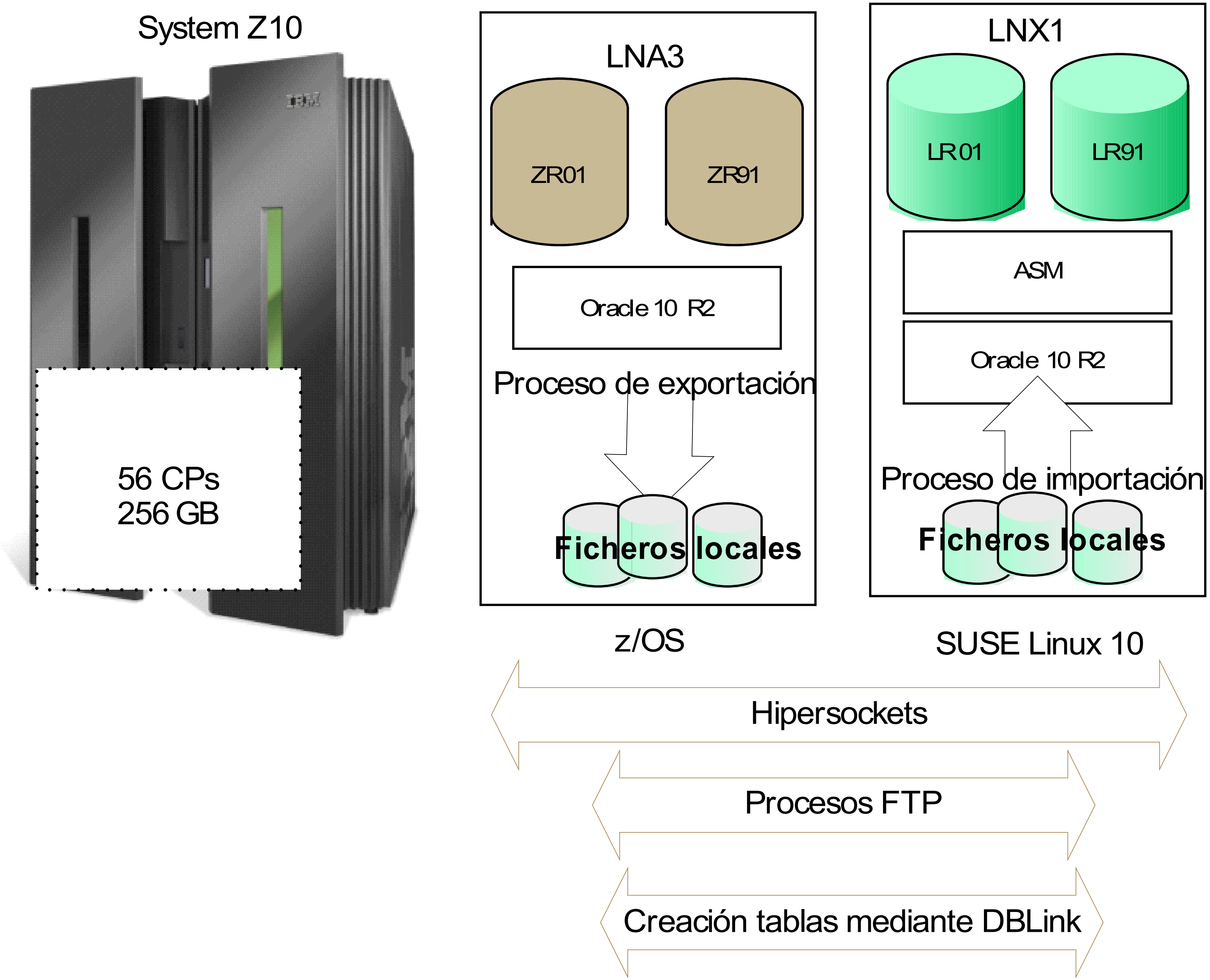
Constraints

- This process is responsible for the migration of NEWPROJECT back end (data structure, data and programs) to the new platform.
- It should maximize the use of hardware/software resources so the execution time is reduced to a minimum.
- The migration process should be completed within the transition window, this is initially estimated in 40 h.

Goal

- The main target is to size the system for the migration execution, and optimize the process to be run within the agreed window.
- The success criteria is set to obtain a complete migration within a window of 40 h.

Testing scenarios and results - Migration



System Z10

56 CPs
256 GB

LNA3

ZR01

ZR91

Oracle 10 R2

Proceso de exportación

Ficheros locales

z/OS

LNX1

LR01

LR91

ASM

Oracle 10 R2

Proceso de importación

Ficheros locales

SUSE Linux 10

Hipersockets

Procesos FTP

Creación tablas mediante DBLink

Testing scenarios and results - Migration

Results(1)

- Due to the previous work during the POC, the migration process was already designed, this allowed to have several unattended test runs, and therefore anticipate the test results.
- The migration runs also served to prepare the required environment for the rest of the tests. Ultimately, there were 12 fully accomplished migrations during one month of the benchmark.
- The tests were according to the following methodology:
  1. Migration execution and systems monitoring.
  2. Migration validation.
  3. Resources consumption analysis (z/OS and Linux partitions).
  4. Process optimization, and systems fine tuning. (taking advantage of the new platform improved capability for increased memory addressing)
- The migration window has managed to be set in 15 hours.

Testing scenarios and results - Migration

Results(2)

- These are the migration process' main features which have been optimized during the benchmark:
  - Export data process from z/OS database to conventional files.
  - Parallel ftp processes for transferring data from the z/OS partition to the Linux one.
  - Adjustments for obtaining the maximum bandwidth provided by hipersockets.
  - Table and structures creation in the Oracle database in Linux.
  - Database configuration.
  - Import data process, programs and properties, from conventional files to the Oracle database in Linux.
  - Index construction.

Testing scenarios and results - Scalability

Constraints

- Customers growing business demands from the new platform to present a remarkable capability to scale up.
- There have been scheduled several testing scenarios for evaluating the platform’s scalability. The system hardware resources have been resized for the increased workload.
- The stress test has to be conducted by the use of the same injection tool employed during the performance phase.

Goal

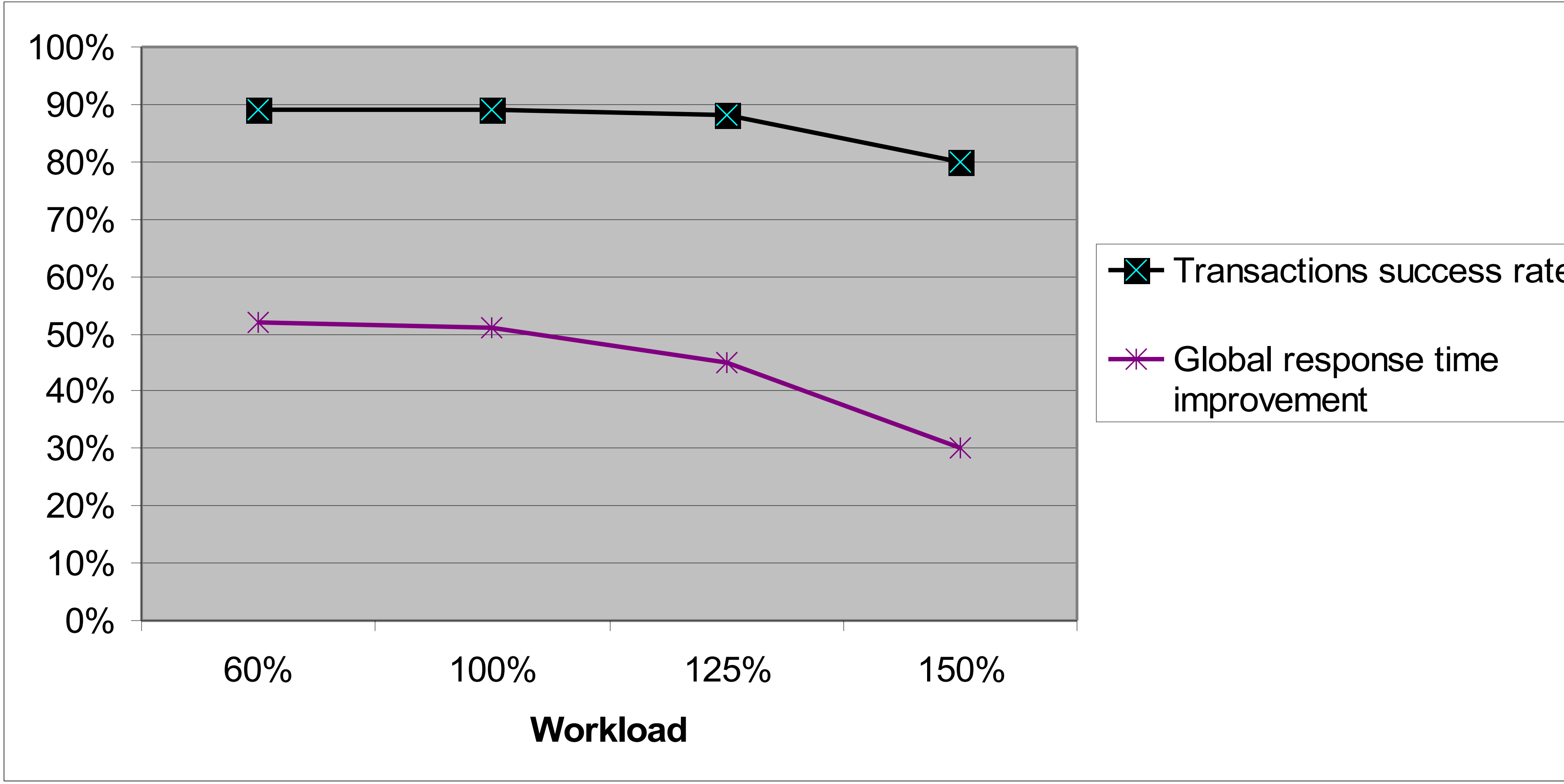
- To obtain a constant transactional response time while increasing the OLTP workload and the hardware resources.
- The success criteria is to reach a maximum workload increase of 150%, keeping the same ratio in between the hardware resources and the workload.

Testing scenarios and results - Scalability

Results

- The scalability phase was completed in 11 days.
- There were set up 4 different scenarios with different OLTP workload, 60%, 100%, 125% and 150%. The 60% scenario corresponds to the workload experienced from 11:30 to 13:30 on a day in September, the rest of the scenarios have been generated by compressing this same workload into a smaller time window.
- All the scalability scenarios total 36 test runs.
- The testing methodology was similar to the one followed during the performance phase.
- The rate of transactions which performed better compared to their performance in production (Transactions success rate) remained constant for the three first scenarios, with a slight decrease when peaking the workload.

Testing scenarios and results - Scalability



The graph only shows the results for the qualified run of each scalability scenario.

Agenda

- Background
- Goals
- Benchmark environment
- Benchmark plan
- Testing scenarios and results
- Conclusions

Conclusions – customer benchmark

- The new platform has achieved all the primary targets for the performance, batch, migration and scalability phases.
- It has been validated a new platform based on:
  - System z10 servers, optimized for database transaction execution.
  - 64 bit Oracle, which expands the memory addressing capability.
  - The usage of additional tools to optimize the new platform’s performance and availability.
- It has been obtained a substantial performance improvement, 50% on OLTP processing, and 46% on the batch processing.
- Valuable information has been collected to right size the new platform.
- Defined the Support Plan to guarantee the support level demanded by the business service.

Visit us !

- Linux on System z: Tuning Hints & Tips
  - http://www.ibm.com/developerworks/linux/linux390/perf/

- Linux-VM Performance Website:
  - http://www.vm.ibm.com/perf/tips/linuxper.html

- IBM Redbooks
  - http://www.redbooks.ibm.com/

- IBM Techdocs
  - http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs

Questions