2012
IBM System z Technical University
Enabling the infrastructure for smarter computing

Practical consolidation experience with Oracle and Linux on System z

zLG25

Siegfried Langer

Abstract

zLG25 Practical consolidation experiences with

Oracle on Linux on System z

Siegfried Langer

Lecture — Intermediate

Many customers have successfully consolidated Oracle databases on the mainframe. Linux on System z provides the strategic platform for Oracle on the highly scalable servers of the IBM System z and zEnterprise. The consolidation of distributed Oracle servers explores a large potential for savings on software license cost and operational cost, such as service personnel, power, space, network cost, and air-condition.

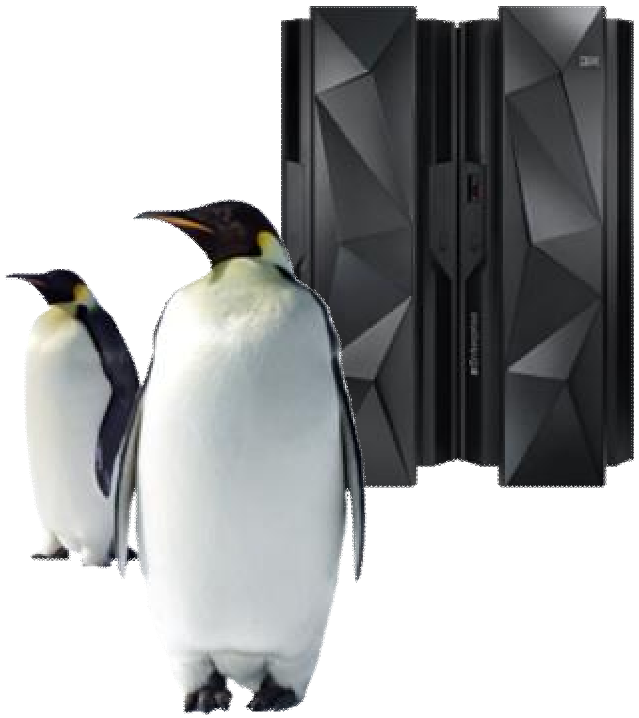
Based on the high degree of integration and virtualization, it is possible to centralize systems management, reducing security risks and management efforts. Centralized backup, high availability, and disaster recovery solutions are easier implemented. Based on practical experience with large scale consolidations, this session will address some of the architecture, services, pitfalls, and migration considerations.

Discussion Topics

Consolidating Oracle database servers

A Real Customer Example

- **Migration services**
- **Performance tuning results**
- **Best practices**



A Real Customer Example

- Large Oracle database consolidation project
  - Oracle 10*g*R2 databases (including a few 11*g*R2 databases)
- Consolidation from x86 (HP ProLiant blade servers) to z196
  - 16 IFL
  - DS8800 with FICON attached ECKD
  - z/VM V6.1
  - RHEL 5.6
- Migration of individual databases over a longer time period
  - Utilizing IBM Migration Services (“Migration Factory”)

Problem statement:

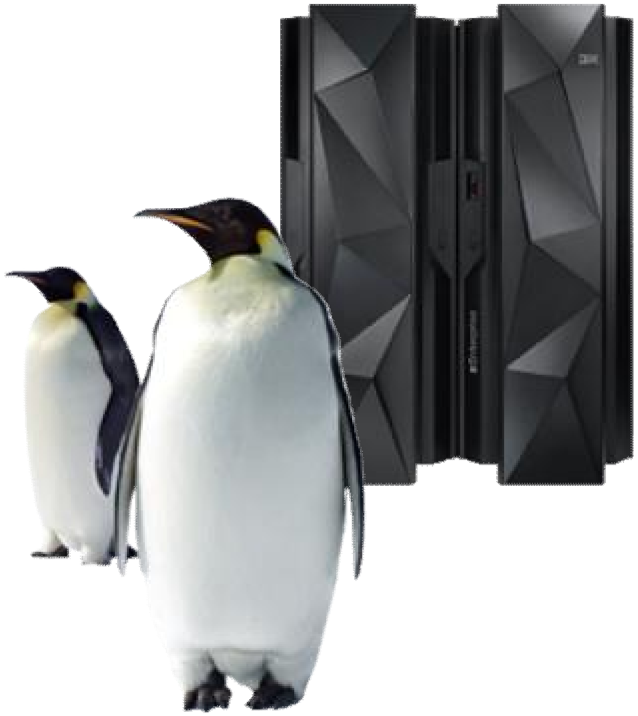
- Customer reported application performance issues with 3 out of approx. 50 databases
  - Business analytics application ‘A’: not completing within expectation
  - Business analytics application ‘B’: not completing within expectation
  - Application ‘C’: increasing number of time-outs (transactions exceeding 1 minute)

Discussion Topics

Consolidating Oracle database servers

A Real Customer Example

- **Migration services**
- **Performance tuning results**
- **Best practices**

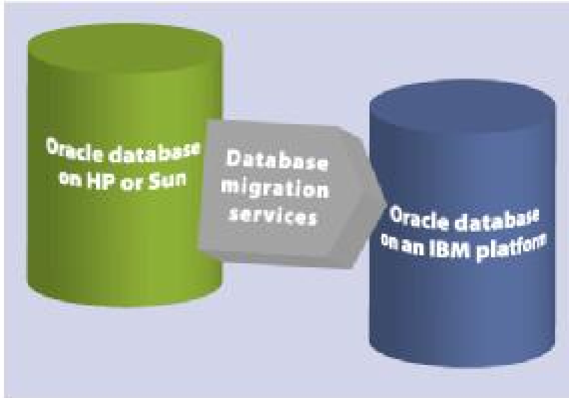


Oracle Database Migration Services

IBM Migration Factory (MF)

How does it work?

- Review your current database environment in a planning session with the MF team
- We tell you how long it will take and how much it would cost.
- We perform automated data collection to establish the metrics for the databases to be migrated.
- We work with you to establish testing requirements and a cutover strategy.
- We prepare a detailed project plan.
- We manage and perform the migration of the required databases according to the plan to help ensure that risk, schedule and cost are correctly managed.
- We confirm that the migrated databases meet your testing requirements.
- We support you during cutover into production.
- We provide basic skills transfer for an established number of your personnel on the migration tasks performed during these services.



THE IBM MIGRATION FACTORY HELPS ANSWER KEY QUESTIONS

- "Can it be done?"
- "How is it done?"
- "What will it cost?"
- "How long will it take?"
- "What are the risks?"

DB Migration Approach – Supported by IBM Migration Factory

Mock migration → Test by application team → if required: repeat Mock migration → Final migration for production

Mock migration

- Take flash copy of DB
- Uses special tool (Xenobridge)
- Highly parallel copy to IFL environment
- Mock migration process can take several hours depending on DB size
- Short outage of production server to take flash copy

Test by application team

- Verify correct application functionality
- Customer task

if required: repeat Mock migration

Final migration for production

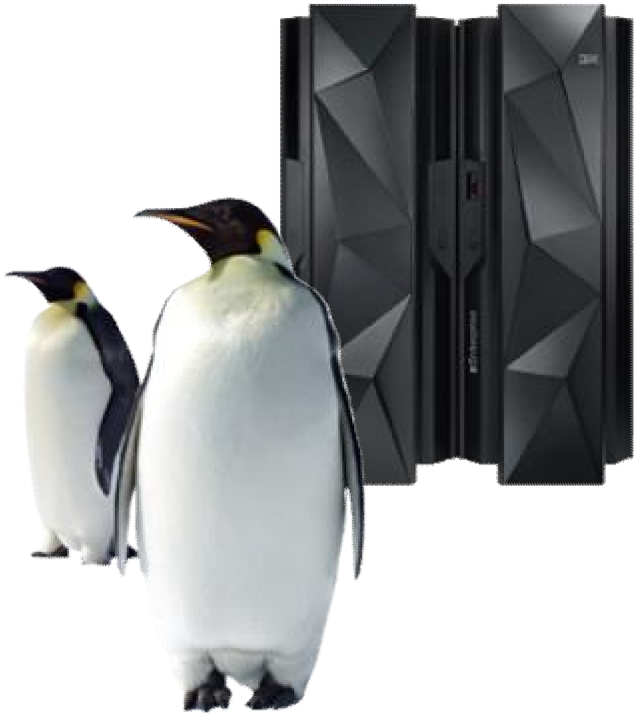
- Re-run of mock
- Uses special tool (Xenobridge)
- Highly parallel copy
- Mock migration process can take several hours depending on DB size
- Application not available during this time
- Switch IP address, assign source DB server IP (x86) to new server on IFL

- Additional service offerings/tools available to minimize outage time during migration
  - Continuous data replication (“CDC”)
  - More complex set-up

Consolidating Oracle database servers

A Real Customer Example

- Migration services
- **Performance tuning results**
- Best practices



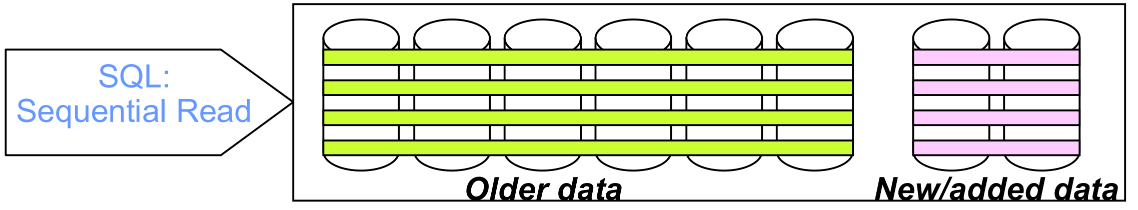
Typical performance challenge

Customer reported performance issue:

- Excessive run time for monthly business analysis run
- Application team states that no changes were made to the application

However…..

- Database size increased significantly
  - by about 12% in 3 month only
    - April +45 GB, May +27 GB, June +32 GB
  - Added 2 Mod A disks (approx. 360 GB)
- Adding disk volumes has an impact on striping
  - New data striped over 2 volumes only (2 disks instead of 6)



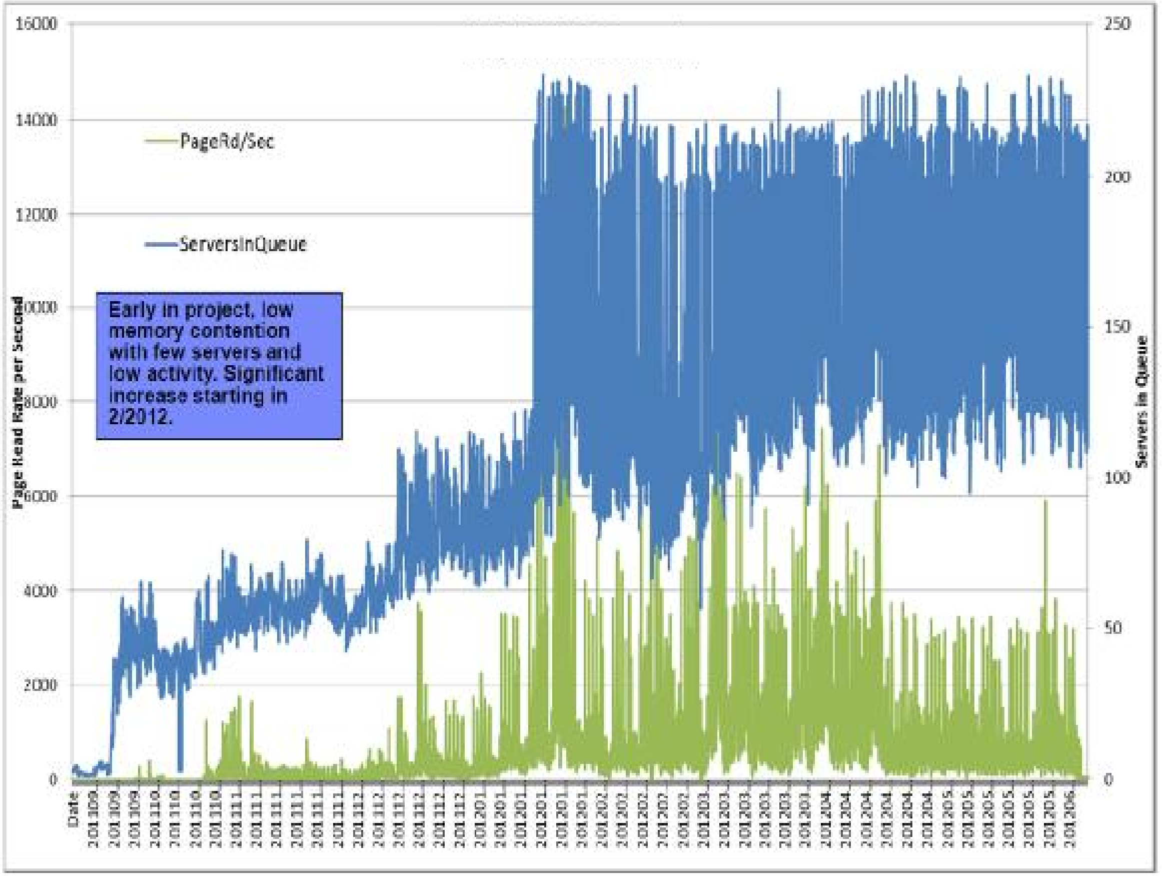
Performance Degradation Over Time

Problem:

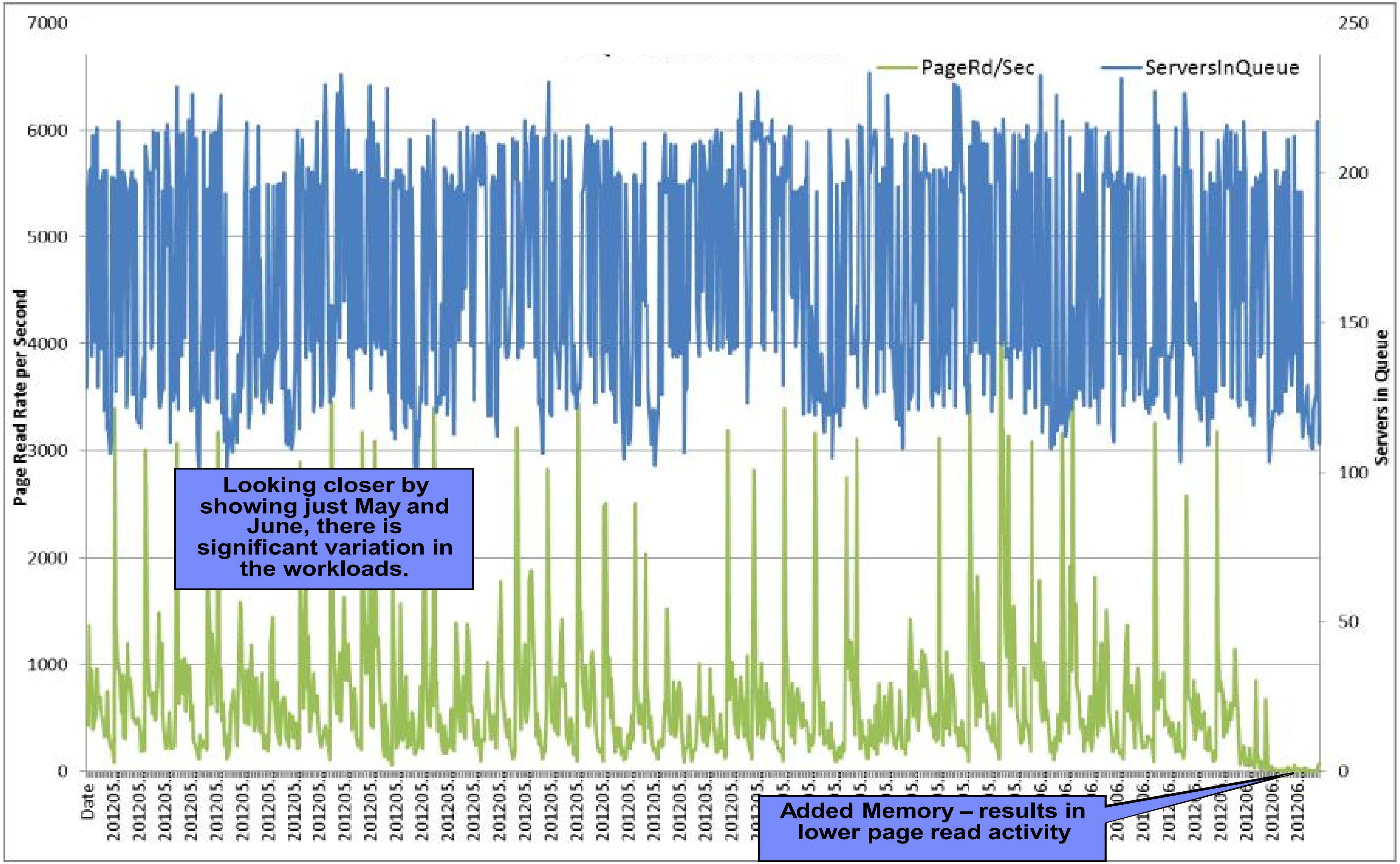
- The performance of selected servers/DB applications became worse over time with increased load on system

Root cause:

- The add'l servers and increased activity led to increased memory contention
- Memory contention led to high paging rates to disks and internal systems management overhead (competing for memory between servers)

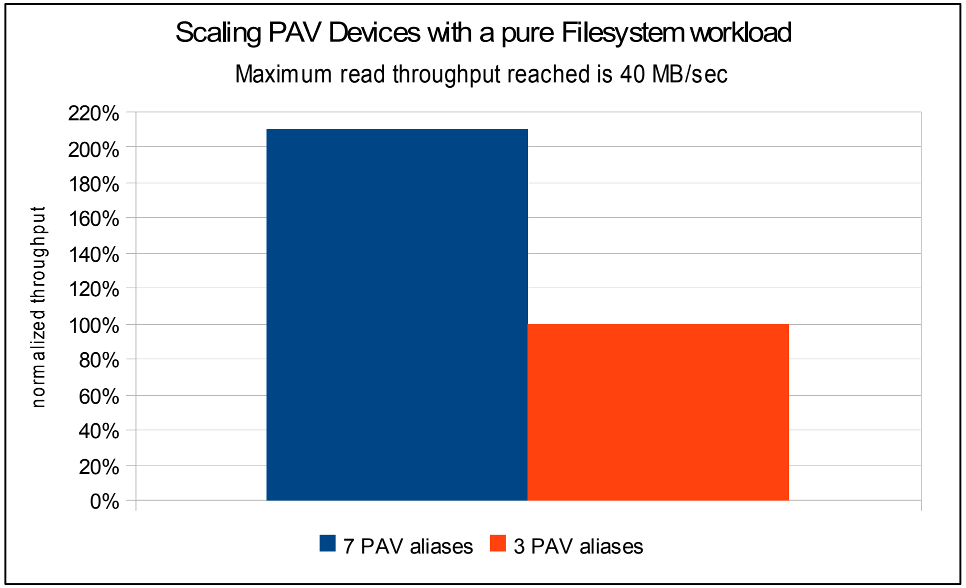


Memory Over-commitment Changes



PAV – Parallel Access Facility

- DASD and PAV devices are directly attached to the guests
  - For disk I/O intensive database workloads is this is the recommended setup
  - It is a requirement for using HyperPAV in Linux
- In case of Minidisk usage
  - Virtual PAV devices and a multipath setup for the Linux guest is required **and**
  - Physical PAV or HyperPAV devices in z/VM are required
- The amount of PAV devices is a critical parameter for disk throughput
- With 7 PAV devices the system can drive 2x more I/Os than with 3 PAV devices
- **Measurements showed that disk access is not a bottleneck with 7 PAV devices**
- *Measurement results are random I/O access pattern (not sequential I/O)*



Notes:

- *HyperPAV is not supported with RHEL 5.6 (supported with RHEL 6 and SLES 11)*
- *HyperPAV substantially reduces disk management (PAV-aliases do not need to be considered)*

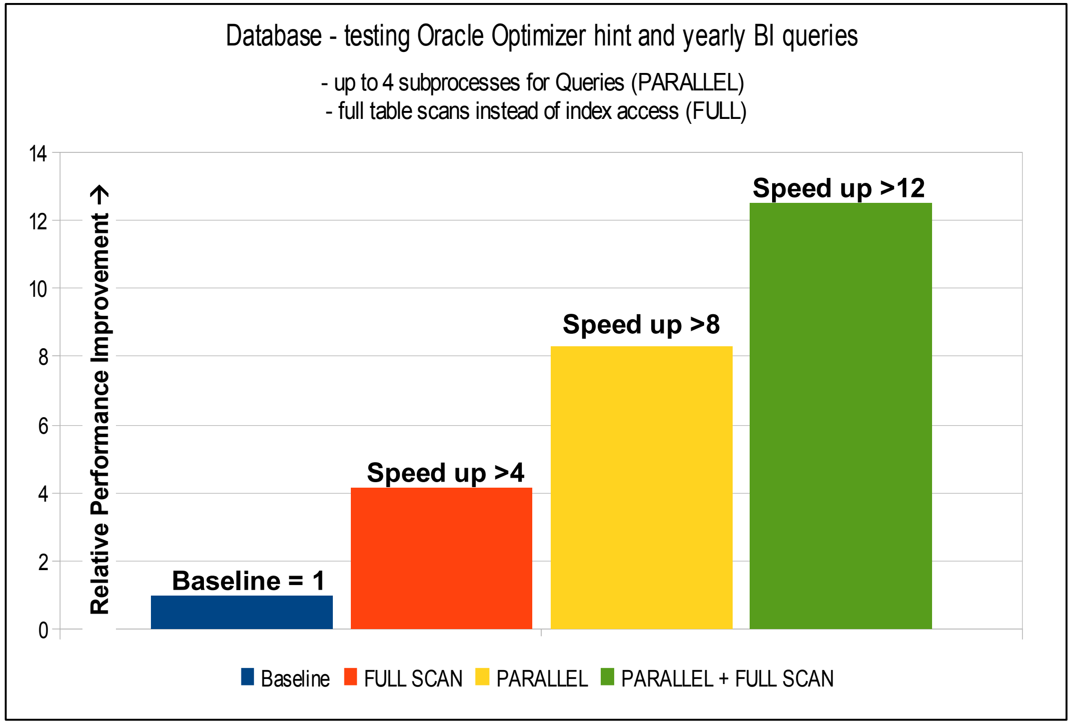
Oracle DB Tuning Activities – Business Analytics Application ‘A’

- **Actions taken – results:**
  - DB and application copied to a “sandbox” environment
    - ➢ Recreation of problem successful
    - ➢ Test runs with historical data from 2011
  - Used FIO (flexible I/O) tool to emulate a database like disk load and stress the disk devices (test achievable disk subsystem bandwidth)
    - Number of PAV devices (data striping – parallel access) increased from 3 to 7 per disk volume
    - ➢ Bandwidth increased from 4 MB/s to 8 MB/s
    - *rr_min_io* changed from 1000 to 1 (Linux default = 1000)
    - ➢ Bandwidth increased from 8 MB/s to 20 MB/s (in test)
  - ➢ **Significant throughput increase for queries in monthly/yearly run**

- **Tests with Oracle optimizer show dramatic further speed-up**

Oracle DB Tuning Activities – Business Analytics Application 'A'

- **Oracle optimizer hints** are specific for the SQL statement where specified
  - 'FULL' force table scans vs index access
  - 'PARALLEL' forces breaking up the statement into parts which can be executed in parallel in the same time
  - 'PARALLEL' and 'FULL'



- Risks
  - Forcing a table scan can result in a severe performance degradation, when index access is the appropriate access method
  - There might be reasons that a certain statements can not be executed parallel, then the behavior will not change

Oracle DB Tuning Activities – Business Analytics Application ‘B’

- Multi-part workflow for data analysis
  - DB copied to a “sandbox” environment, directed the original workload against the “sandbox” system
  - Workload consist of
    - 3 steps (S, R, and D) with different workflows
    - only the last two steps (R and D) are performance critical

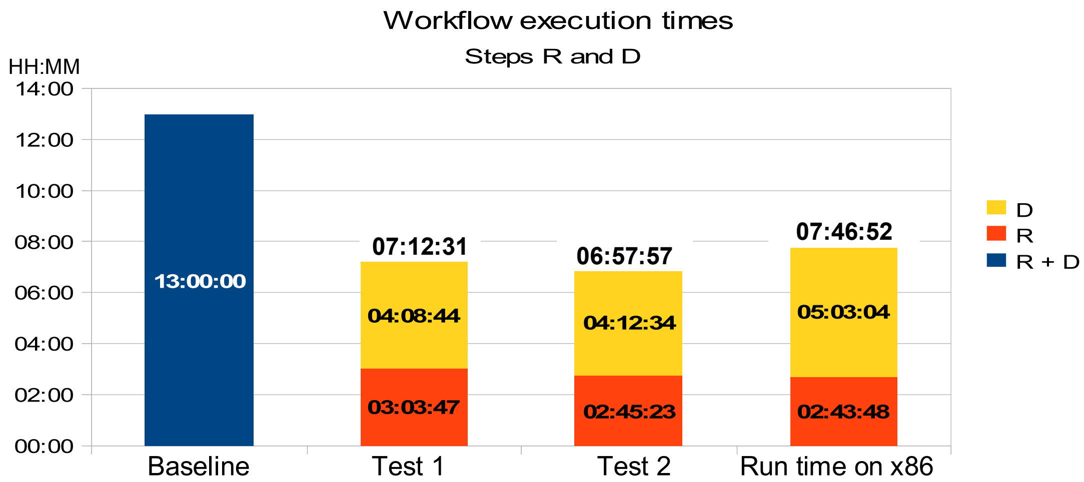
- **Baseline**: **13 hours** run time for analysis with full year data
  - Initial migrated setup

- **Test 1** (run time 07:12:31)
  - Environment related tuning (memory, disk setup, etc.)
  - Nearly factor 2x improvement

- **Test 2** (run time 06:57:57)
  - All tuning changes from Test 1 and
  - Database specific tuning (Oracle parameters)
  - Both tuning steps together provide an improvement of slightly more than factor 2x against the baseline

Oracle DB Tuning Activities – Business Analytics Application 'B'

Parameter changes:



- Test 1 (run time 07:12:31)
  - Added memory to LPAR
  - Enabled 7 PAV devices per DASD device, directly attached to the guest,
  - Multipath setup: round robin with *rr_min_io=1*
- Test 2 (run time 06:57:57)
  - Ensure that huge pages are really used → caused a SGA reduction from 8192MB to 7600MB *(better solution would have been to increase the amount of configured huge pages)*
  - Profile parameter changes:
    - *db_writer_processes=2* (prior 8),
    - *filesystemio_options=setall* (prior asynch),
    - *parallel_degree_policy=auto* (prior manual),
    - *pga_aggregate_target=3700M* (prior 3,221,225,472)
  - Added parameters:
    - *log_buffers=104,857,600*
  - Removed parameters:
    - *disk_asynch_io,*
    - *log_checkpoint_timeout,*
    - *optimizer_index_caching,*
    - *optimizer_index_cost_adj,*
    - *shared_pool_size*

Oracle DB Tuning Activities – Application 'C'

Oracle back-end for Windows application server - transaction workload

- **Critical limit:**
  - Requests should finish within 60 seconds
  - Only 30 time-outs (>60 sec) are acceptable within 24 hour window

	Known as Good case	Problem Case	After tuning action part 1	After tuning action part 2
Measurement Duration	24 h	23 h	17.25 h	48 h
Less than 3 Sec	91,79%	88,37%	88,31%	99,97%
3 to 5 Sec	2,74%	3,35%	3,69%	0,02%
5 to 10 Sec	2,74%	3,50%	3,20%	0,01%
10 to 60 Sec	2,58%	4,48%	4,27%	**0,00%**
More than 60 Sec	0,16%	0,30%	0,53%	**0,00%**
More than 60 Sec	**13 requests**	**29 requests**	**24 requests**	**0 requests**

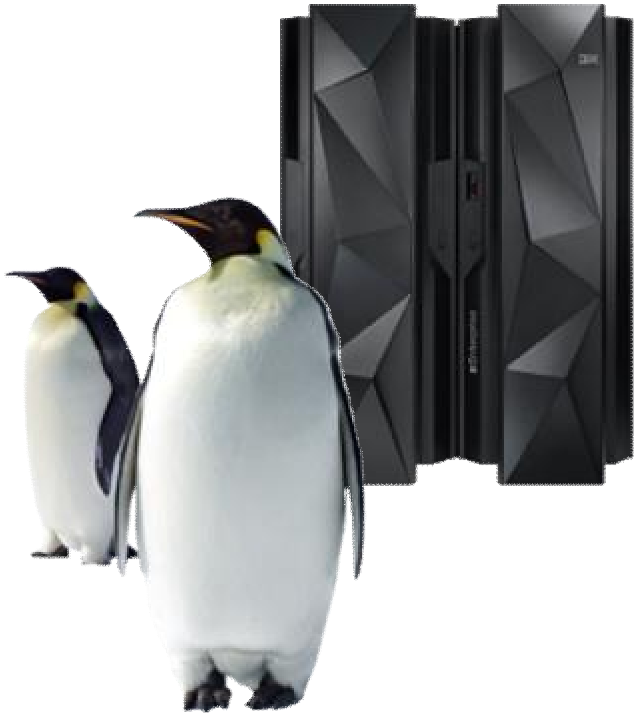
- **Tuning actions part 1:**
  - Increased PAV devices from 3 to 7
  - *rr_min_io = 1*
  - Shut down inactive servers (reducing memory pressure)
  - Further analysis showed a correlation with swapping activities - increased virtual memory size of Linux guest by 2 GB and activate direct I/O
    - ➢ Environment monitoring showed good results, still getting time-outs
- **Tuning actions part 2:**
  - Increased number of vCPUs from 2 to 4, increased SGA by 2 GB
    - ➢ **Dramatic improvement – no time-outs**
    - ➢ **Results confirmed by longer term monitoring**

Discussion Topics

Consolidating Oracle database servers

A Real Customer Example

- Migration services
- Performance tuning results
- **Best practices**



General Recommendations – Monitoring

Establish permanent monitoring

- z/VM Performance Toolkit
- Linux sadc/sar
- Tivoli OMEGAMON® XE on z/VM® and Linux
  - Tivoli Composite Application Manager (ITCAM) for Applications – Oracle Agent

- **Pro-active systems management**
  - **Detect potential problems/bottlenecks before users complain**
- **Capacity planning**
- **Accounting – charge back**

General Recommendations – z/VM

- **z/VM Performance Toolkit**
- Ensure the virtual to real memory ratio stays in an appropriate range for the workloads
  - Indicators of impact:
    - z/VM Paging activity
      - Report 'User Paging Activity and Storage Utilization' (UPAGE, FCX113)
      - Columns: 'X>DS' paging to DASD, critical: Reads paging from DASD
    - z/VM Guest Waits
      - Report 'Wait State Analysis by User' (USTAT,FCX114)
      - Especially columns %PGW, %PGA, and %CFW
    - z/VM CPU load

      Report 'System Performance Summary by Time' (SYSSUMLG, FCX225)

      Report 'General CPU Load and User Transactions' (CPU, FCX100)

- Disable Page reorder for guests larger than 8 GB
  - Find more information at http://www.vm.ibm.com/perf/tips/reorder.html

General Recommendations - Linux

Two possible disk devices for System z:

- Fixed (512-byte) blocks SCSI, connected with Fiber Channel Protocol (FCP) connection technology
  - SCSI storage can be faster because it supports multiple parallel I/Os to a storage device
  - FCP requires that you manually install FCP and configure multipath

- DASD Disk I/O (FICON attached ECKD disks)
  - Required: sufficient PAV devices (minimum 7 per disk) or HyperPAV (20 per LCU)
  - In case of MDISKs use virtual PAV devices in Linux and physical PAV devices in z/VM. Use of HyperPAV would be the preferred method (supported in RHEL 6 and SLES 11).
  - Multipath setup: *set rr_min_io* parameter to *1*

General Recommendations - Linux

Memory requirements:

- Don't over-configure Linux memory because -
  – Excess memory allocated to the Linux guest is used by Linux for I/O buffer and File system cache
  – In a virtualized environment under z/VM, oversized guests place unnecessary stress on the VM paging subsystem
  – Real memory is a shared resource, caching pages in a Linux guest reduces memory available to other Linux guests.
  – Larger virtual memory requires more kernel memory for address space management.

- Consider setting *vm.swapiness* to 0 (sysctl.conf) for all systems which are running primarily databases using page cache I/O
  – Defines a preference to reuse page cache pages instead of swap application pages

General Recommendations – Linux
Huge Pages

- If huge pages are configured, this amount of memory is no longer available for applications using 4K pages
  - Oracle 11*g* can use huge pages automatically
    - If the SGA can not be allocated as a whole in huge pages, the fall back is to allocated the whole SGA in 4KB pages, which can produce a heavy memory pressure.
  - Ensure to have enough huge pages defined that the full SGA from **all** Oracle 11*g* databases in that system server fits into

- Check /proc/meminfo
  - HugePages_Total: configured huge pages,
    - e.g via *vm.nr_hugepages*
  - HugePages_Free: unused part from HugePages_Total,
    - but might be, not all are allocate-able due to memory fragmentation
  - HugePages_Rsvd: these are huge pages in any case available
  - pre-allocate huge pages on the kernel boot command line by specifying the "*hugepages=N*" parameter, where 'N' = the number of huge pages requested.
    - This is the most reliable method for pre-allocating huge pages as memory has not yet become fragmented!

- To verify usage of Hugepages
  - Monitor value of HugePages_Free: When starting Oracle 11*g* the amount value of HugePages_Free must be lower (reduced by the SGA size)

General Recommendations – Oracle parameters

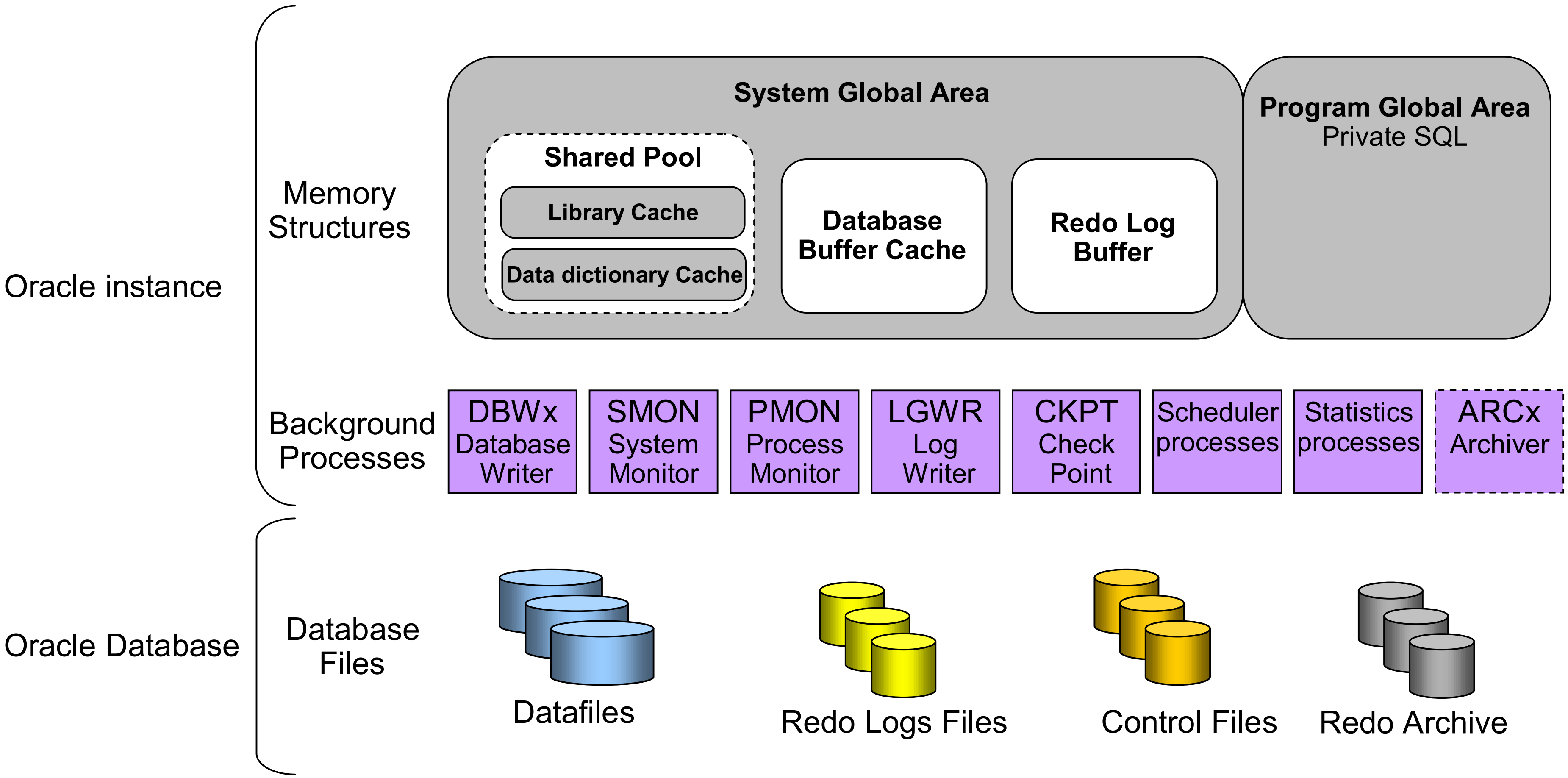
- **Highly recommended:** parameter *filesystemio_options=setall*
  - In combination with this, remove definitions of parameter disk_asynch_io
- When defining **SGA_TARGET**, Oracle Database 10*g* automatically sizes the most commonly configured components, including:
  - The shared pool (for SQL and PL/SQL execution)
  - The Java pool (for Java execution state)
  - The large pool (for large allocations such as RMAN backup buffers)
  - The buffer cache
  - The Streams pool
  - Consider removing the existing definitions (if not sure) and let Oracle handle the sizing
    - It defines lower limits and reduces the range Oracle can manage the buffers dynamically
- **Remove parameter *.log_checkpoint_timeout=0.**
  - It is not recommended to set this parameter unless FAST_START_MTTR_TARGET is set.
  - It is known as a potential cause for performance issues.
- **Define log_buffer = 104857600** or larger
- Be careful with specifying optimizer parameters (optimizer_...) as global parameters, because it might be an advantage only for some workloads.
  - Optimizer hints in the SQL statements are probably better because given for specific select statements

General Recommendations – Oracle parameters

- **Log Setup**
  - Place redo logs on separate disks
    - Single disks are sufficient, striped LVM not needed
    - Ensure to have no other activity on these disks
  - Recommendation: Usage of larger log files
    - e.g. 4x 1 – 1.5 GB to reduce the frequency of log switches

- **Review existing optimizer hints!**
- Customer workload specific experience with Oracle optimizer hints:
  - Got very good improvements with the hints FULL(<table name>) and PARALLEL(<table name>, <number of CPUs>) for BI queries
  - Suggest to review existing optimizer hints. Examples:
    - Combination of full(t) and parallel_index(t, 12) seems to be contradictory because usage of full table scan or index are mutually exclusive
    - Degree of parallelism specified with 12 seems to be much too high for a system with 4 vCPUs. A typical level for parallelism is <amount of vCPUs> or <amount of vCPUs + 1>, the upper limit is no more than 2X the number of cpus/virtual cpu
  - For Oracle 11g consider to specify parallel_degree_policy=AUTO instead of explicit optimizer hints to let Oracle decide about parallelism

Oracle server architecture



Example of memory sizing

- Standard Memory estimation = sum of:
  - Memory required for Linux Kernel: 512 MB
  - Memory required for Oracle SGA: As per DBA estimation
  - Memory required for Oracle PGA: As per DBA estimation
  - Memory required for Oracle ASM: 256 MB to 512 MB (If ASM is used)
  - Memory required for additional agents like OEM, Tivoli etc., as needed by the application
  - Linux Overhead requirements: 5 % of the total memory

Starting size = SGA + PGA + 0.5GB for Linux + ASM (if used)

- Memory over-commitment (relationship of virtual to real memory)
  - Limit/avoid memory over-commitment for critical production databases
  - Test/development guests can benefit from z/VM memory over-commitment capability

ASM

- Oracle ASM is an Oracle instance with a smaller SGA than regular database
- Oracle ASM is Oracle's methodology for striping database files across as many disk devices as possible.
- Oracle ASM is a form of software striping to raw or block devices
- When configuring ASM make sure that Disk/LUNs are assigned with the same size, type, and speed.
- Oracle ASM for Oracle 11g utilizes a 1 MB stripe size to stripe the database files across all the disk devices assigned to a particular disk group.
- Oracle REDO logs are also striped across the disk devices in the disk group, but are internally striped with a 128 KB stripe size.
- Oracle recommends the SAME approach for ASM files as well, by having one or two disk groups (if utilizing a Flash Recovery Area) and not separating the data and index data files into different disk groups.

Best practices – Oracle and Linux on System z

- Big database servers (SGA >100 GB) should be run in LPAR rather than as z/VM guest
- As z/VM guest use as few virtual processors as possible
  - The number of guest processors (virtual CPU) should be less or equal to the number of processors of z/VM LPAR
- Busy Linux database servers as z/VM guest should be given enough guest memory so that paging for this guest can be minimized
- There should be at least 2 GB of Expanded Storage defined for z/VM
- Size a Linux database server as z/VM guest that it just does not swap
- Use direct I/O for database files
  - Right-sizing the buffer pool is more beneficial than having additional Linux page cache
- Separate database disks and disks for logging/archive log
- Define sufficient I/O bandwidth for database disks
  - For SCSI discs, define multipathing and failover (understand & consider disk architecture)
  - For ECKD disks, use HyperPAV (SLES 11, RHEL 6) or define PAV aliases (more is better)
- Use data striping
  - ASM is Oracle's methodology for striping database files across as many disk devices as possible
  - XIV disk storage system has its own internal striping

Questions?

Siegfried Langer
Business Development Manager
z/VSE & Linux on System z

IBM Deutschland Research
& Development GmbH
Schönaicher Strasse 220
71032 Böblingen, Germany

Phone: +49 7031 - 16 4228

Siegfried.Langer@de.ibm.com

Notices

Trademarks

- This presentation contains trade-marked IBM products and technologies. Refer to the following Web site:

  http://www.ibm.com/legal/copytrade.shtml