

**Performance and scalability of a large OLTP workload
with DB2 9 for System z on Linux
December 2007**



Performance and scalability of a large OLTP workload with DB2 9 for System z on Linux

**Performance and scalability of a large OLTP workload
with DB2 9 for System z on Linux**

Table of Contents

Executive Summary	3
Introduction	3
Environmental Setup.....	3
Server software	4
Linux Kernel tunables.....	5
I/O scheduler.....	5
Database Implementation	5
Asynchronous I/O	6
Direct I/O.....	6
Workload Description.....	6
Results	6
Summary.....	8

Performance and scalability of a large OLTP workload with DB2 9 for System z on Linux

Executive Summary

This paper demonstrates data server scalability for DB2[®] 9 running on IBM System z[™] with Linux[®]. It captures the best practices for deploying IBM DB2 9 for mission critical Online Transaction Processing (OLTP) workloads in their product environment.

IBM System z technologies offer a rich set of platform-deployment features ranging from well-known mainframe attributes such as reliability, availability, scalability, and security to features like the very high capacity and bandwidth of the I/O subsystem. DB2 9, a leading data server in the information management market, offers rich value added capabilities to help customers meet their transaction processing demands. These new capabilities include automatic storage, self-tuning memory management, deep compression, and XML. In addition to these capabilities, DB2 9 on System z has added support for asynchronous and direct I/O – interfaces known to improve data server performance for OLTP workloads. This combination of DB2 9 on Linux on System z is ideal for customers looking to drive the performance and scalability of their growing business.

Introduction

DB2 9 is an extremely powerful and flexible data server that offers industry-leading OLTP performance. It is highly optimized for each of the operating system environments it supports. This paper showcases performance and scalability of DB2 running on Linux on System z using an OLTP workload. It describes the behavior of DB2 in a high-throughput transactional environment on a "mid-sized" IBM System z9[™] system. It describes the best practices used to configure the data server by including a description of the hardware configuration, operating system tuning, and database implementation. This paper also demonstrates the impact on performance as a database grows over time. The amount of data accessed per transaction remained the same to emulate typical customer-like scenarios where the most recent data is referenced. Finally, the paper compares the performance of DB2 running on a System z LPAR versus a guest z/VM[®].

Environmental Setup

The LPAR environment consisted of one LPAR on an 18-way IBM System z9 Enterprise Class (z9[™] EC), model 2094-S18 equipped with eight CPUs and 59 GB of main memory. The guest z/VM environment was similarly configured, but had an additional 1 GB of memory for expanded storage. The client workstations were connected via a 1 GB Ethernet LAN.

The storage server was a DS8000[™]. The disk drive modules had a size of 73 GB each and 15000 RPMs. They were configured as one RAID5 array per rank.

Performance and scalability of a large OLTP workload with DB2 9 for System z on Linux

Each rank has either seven (plus one spare drive) or eight physical disks, so that a total of 120 disk drives were used. For the Linux operating system, DB2 application, and DB2 logging we used one rank with ECKD™ 3390 mod 9 disks connected via eight FICON® express channels. For the database data disks we used 15 Ranks of FCP disks connected Point to Point via eight FPC 2 Gb/sec channels.

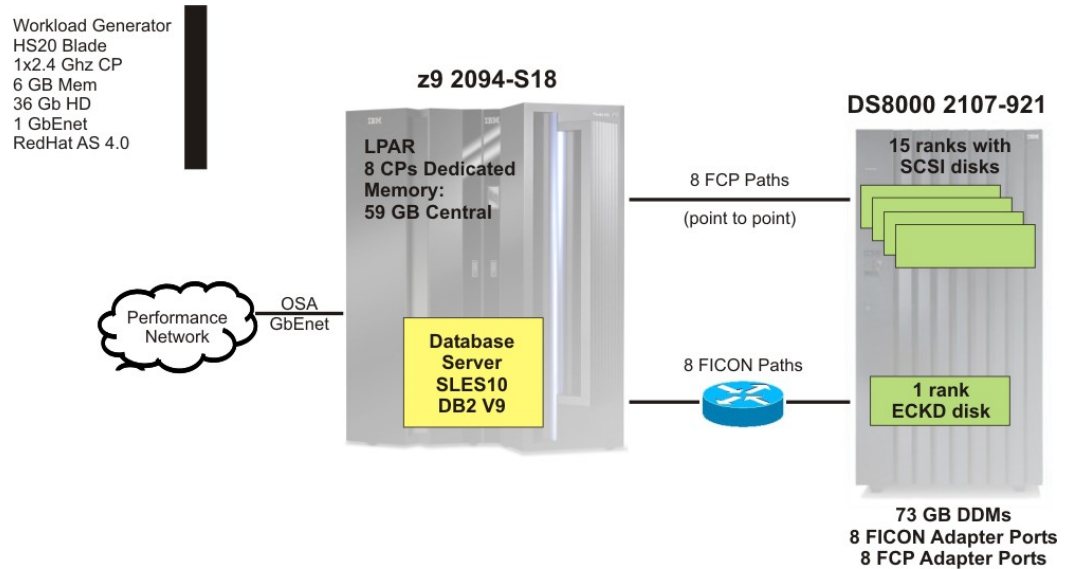


Figure 1. System setup for the OLTP workload on the large database

Server software

Table 1 lists the software and levels used for this test.

Table 1. Software used

Software	Version
SUSE Linux Enterprise Server 64-bit	10
IBM DB2 Enterprise	9.1 ¹
z/VM	5.2.0 Service Level 0602 (64-bit)

¹ DB2 9.1 FP1 or later is recommended because this level of Linux has the fix for STMM to grow and shrink shared memory.

Performance and scalability of a large OLTP workload with DB2 9 for System z on Linux

Linux Kernel tunables

I/O scheduler

The following Linux kernel parameters were configured on the data server. We selected the deadline I/O scheduler¹ and used the following parameters:

- *front_merge 0 (default 1) - avoids scanning of the scheduler queue for front mergers, which rarely occur. This was expected to save CPU time.*
- *write_expire 500 (default 5000) - force the same timeout for write requests as for read requests.*
- *writes_starved 1, (default 2) - process the next write request after each read request (per the default, two read requests are done before the next write request).*

For the shared memory parameters, we set:

- *kernel.shmmax = 63350767616 (maximum size of a shared memory segment in bytes)*
- *kernel.shmmni = 4096 (maximum number of shared memory segments)*
- *kernel.shmall = 15466496 (available memory for shared memory in 4 K pages)*

The above values enabled the entire 59 GB memory size for one shared memory segment (shmmax, in bytes) to have a maximum of 4096 segments and 59 GB total shared memory (shmall, in pages). This enabled large segments to be created and avoided the need for thousands of small shared memory segments with their accompanying overhead.

Database Implementation

The database was created across all 15 ranks on SCSI disks while the logs are placed on a separate rank with ECKD disks. To enable parallelism, we allocated 15 containers for each table and index and placed each container on a SCSI disk from a different rank. Each rank contained eight SCSI disks of 47 GB size and two SCSI disks of 6 GB size. We created EXT3 file systems on all the disks and spread the large tables evenly across them. We created 14 database-managed table spaces with the NO FILE SYSTEM CACHING clause (see [Direct I/O](#)) – using a mixture of regular and large table spaces. Large table spaces, introduced with DB2 9, differ from the regular type primarily in that they can store table objects that are thirty-two times larger in size compared to regular table spaces. Furthermore, large slots, which are also available, allow more than 255 rows per data page.

¹ The default I/O scheduler is the Completely Fair Queuing (CFQ) scheduler.

Performance and scalability of a large OLTP workload with DB2 9 for System z on Linux

For the memory configuration, approximately 42 GB of RAM was allocated for the database memory. 37 GB of that RAM was used for buffer pools.

Another tuning implementation was the enablement of asynchronous I/O (AIO) for the page cleaners.

Asynchronous I/O

AIO permits a single application thread to overlap processing with I/O operations. In other words, the application thread does not have to wait for an I/O request to complete before it resumes its regular processing. AIO is enabled via the registry variable, DB2LINUXAIO.

Direct I/O

Direct I/O (DIO), or unbuffered I/O, is an I/O operation that avoids copying the data from a user space buffer to the page cache from the Linux kernel. This requires that the application handle all I/O optimizations, for example, caching and request merging, by itself. DIO is only available with FCP disks and is enabled via create/alter table space commands using the NO FILE SYSTEM CACHING clause.

Workload Description

The workload used for this study models an order-entry wholesale part supplier business. It simulates users executing transactions against the database. These transactions include order entry and delivery, payment entry, order status queries, and stock level queries.

Results

We began our experiment by running under a single LPAR. We ran the OLTP workload with 200 concurrent users against a 238 GB database. We then scaled the database to 1.4 TB while keeping the number of users and working set size constant. The result (shown in [Figure 2](#)) shows the throughput remains relatively equal even though the database size increased by a factor of six. The average I/O rate for both tests was approximately 18,000 I/Os per second with an average log rate of 6.7 MB per second.

**Performance and scalability of a large OLTP workload
with DB2 9 for System z on Linux**

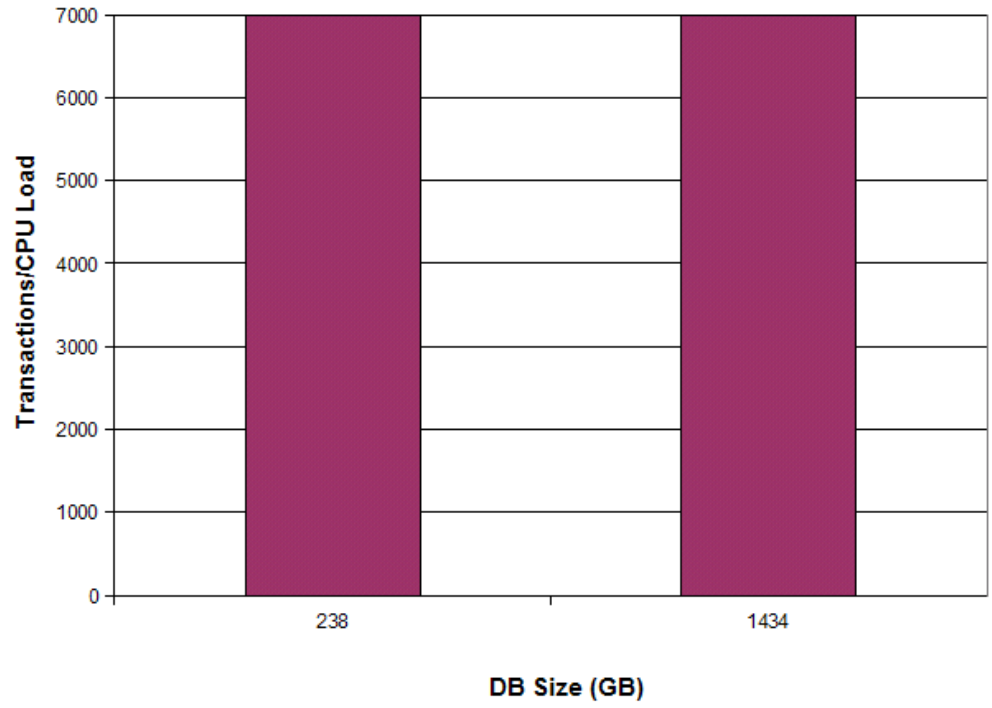


Figure 2. Relative performance for varying database size

Performance and scalability of a large OLTP workload with DB2 9 for System z on Linux

For the second part of the experiments, we ran the workload on a guest z/VM. To keep the experiment consistent with that of the LPAR, we ran 200 concurrent users against the 1.4 TB-sized database. The results, shown in [Figure 3](#), show the performance on an LPAR is approximately 11% better than that of the guest z/VM.

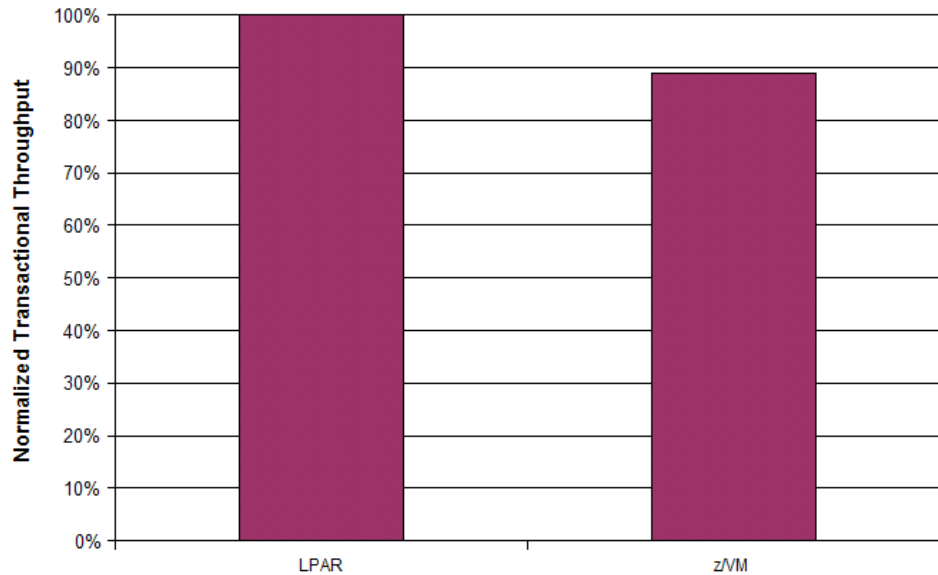


Figure 3. Relative performance of DB2 on System z LPAR versus guest z/VM

Summary

DB2 9 on Linux on System z is a scalable solution for large OLTP workloads. This paper illustrated that a high volume of transactions can be processed on a large database. It demonstrated that the database size can grow by a factor of six to approximately 1.4 TB with no marked performance change. The paper also shows that DB2 9, running on a System z LPAR and z/VM, performs very well. Customers looking to implement their mission-critical data servers can confidently rely on the combination of the extensive value added capabilities and performance enhancement features of DB2 9 with the scalability and robustness of Linux on System z to implement their business solution.

Performance and scalability of a large OLTP workload with DB2 9 for System z on Linux



© Copyright IBM Corporation 2007

IBM Corporation
New Orchard Rd.
Armonk, NY 10504
U.S.A.

Produced in the United States of America
12/07

All Rights Reserved

IBM, IBM logo, DB2, DS8000, ECKD, FICON, System z, System z9, z9 and z/VM are trademarks or registered trademarks of International Business Machines Corporation of the United States, other countries or both.

The following are trademarks or registered trademarks of other companies

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Intel is a trademark of Intel Corporation in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds in the United States and other countries.

SUSE is a registered trademark of Novell, Inc., in the United States and other countries.

Other company, product and service names may be trademarks or service marks of others.

Information concerning non-IBM products was obtained from the suppliers of their products or their published announcements. Questions on the capabilities of the non-IBM products should be addressed with the suppliers.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

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.

THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS DOCUMENTATION, 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 DOCUMENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS DOCUMENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE

ZSW03028-USEN-00