December 2003

DB2 Information Management
Software

THREE-TIER SAP® SD STANDARD APPLICATION BENCHMARK - IBM® DB2® UNIVERSAL DATABASE™, IBM *e*server xSERIES® AND LINUX

TECHNICAL WHITEPAPER

By

Guiyun Cao and John Tran

DB2 Advanced Technology and Performance

IBM Software Group

1. Introduction

The Linux operating system is rapidly maturing into a reliable platform for enterprise database servers. Performance scalability enhancements have made it a viable platform solution for data management businesses. Furthermore, the lower cost of ownership has made it an attractive alternative for those businesses trying to shrink their IT budgets. Many such companies are replacing costly Microsoft® Windows® servers with Linux servers because of the cost-benefit gains. IBM is aggressively exploiting the value and efficiencies of the advanced database technology on Linux systems by bringing IBM® DB2® Universal Database™ (DB2 UDB) proven performance, scalability and features to the Linux environment.

To demonstrate DB2's performance and scalability on Linux, IBM executed a three-tier SAP® Sales and Distribution (SD) Standard Application Benchmark with DB2 UDB Version 8.1 on IBM @server® xSeries® x440 servers and SuSE Linux Enterprise Server V8. The SAP SD Standard Application Benchmark is not only one of the most demanding benchmarks in the industry today for evaluating the performance of ERP solutions, but it can also help companies evaluate hardware and software configurations.   The result of this SAP SD Standard Application Benchmark with 5,790 SAP SD users[1] shows that DB2 running on the IBM x440 provides superior performance and capacity. This configuration outperformed other SAP SD Benchmark results with the same number of database CPUs by a significant margin. Backed by the power and scalability of the Linux operating system, IBM DB2 UDB V8.1 is the database of choice for mission-critical applications.

The purpose of this paper is twofold. First, it describes the benchmark system and the industry-leading three-tier 4-way SAP SD Standard Application Benchmark result that was published on it.   The benchmark system description includes the hardware and software configurations.   The software configuration includes a DB2 V8.1 database server, the SuSE SLES8 operating system, and the SAP R/3® software system. This paper also discusses some performance inefficiencies associated with the SLES8 Linux kernel and how these inefficiencies were identified and eliminated. The second purpose of this paper is to describe how to configure and tune an SAP R/3 database system for optimal performance. It focuses on key DB2, SAP R/3, and Linux kernel parameters and design issues that have the most performance impact.

---

[1] The SAP Sales and Distribution Standard Application Benchmark was certified on March 17, 2003. mySAP™ business solution component: SAP R/3 Release 4.6C. Three-tier configuration, using IBM eServer xSeries x440, DB2 UDB v8.1 and SuSE Linux Enterprise Server 8.0. 5,790 SAP SD users, certification number 2003018. For more information, see http://www.sap.com/benchmark.

2. Three-Tier SAP S D Standard Application Benchmark Overview

The SAP SD Standard Application Benchmark has become a *de facto* standard for evaluating the performance of many ERP solutions. Its high demand on CPU and other database resources is similar to that in extremely high-volume OLTP environments. It can be configured as either two-tier or three-tier. Within the two-tier configuration, the database server and the SAP R/3 application server run on the same physical machine. In the three-tier configuration, the presentation tier, application tier, and database tier each run on separate physical servers. The three-tier configuration benchmark is more representative of today's enterprise-class SAP solution-based production environments.

The SD benchmark driver provided by SAP simulates end users driving business transactions in dialog steps. During the benchmark run, the database system is fully stressed to support the maximum number of users while maintaining the average, weighted dialog response time of less than two seconds. The user think time is set as ten seconds between dialog steps, which closely approximates the behavior of an experienced power user in a real-world environment.

The SAP SD Standard Application Benchmark workload consists of the following transactions:
- Create an order with five line items (transaction VA01)
- Create a delivery for this order (VL01)
- Display the customer order (VA03)
- Change the delivery (VL02) and post goods issue
- List 40 orders for one sold-to party (VA05)
- Create an invoice (VF01)

3. System Configuration

3.1 Hardware Configuration

The hardware configuration for the benchmark system is illustrated in Figure 1. Eleven application servers are connected to the database server via Ethernet. Also on the Ethernet network is an IBM RS/6000® system acting as the SAP benchmark driver. Directly attached to the database servers are four EXP300s. These disks are used for database logs and tablespaces. Detailed information on each server is listed in Table 1.
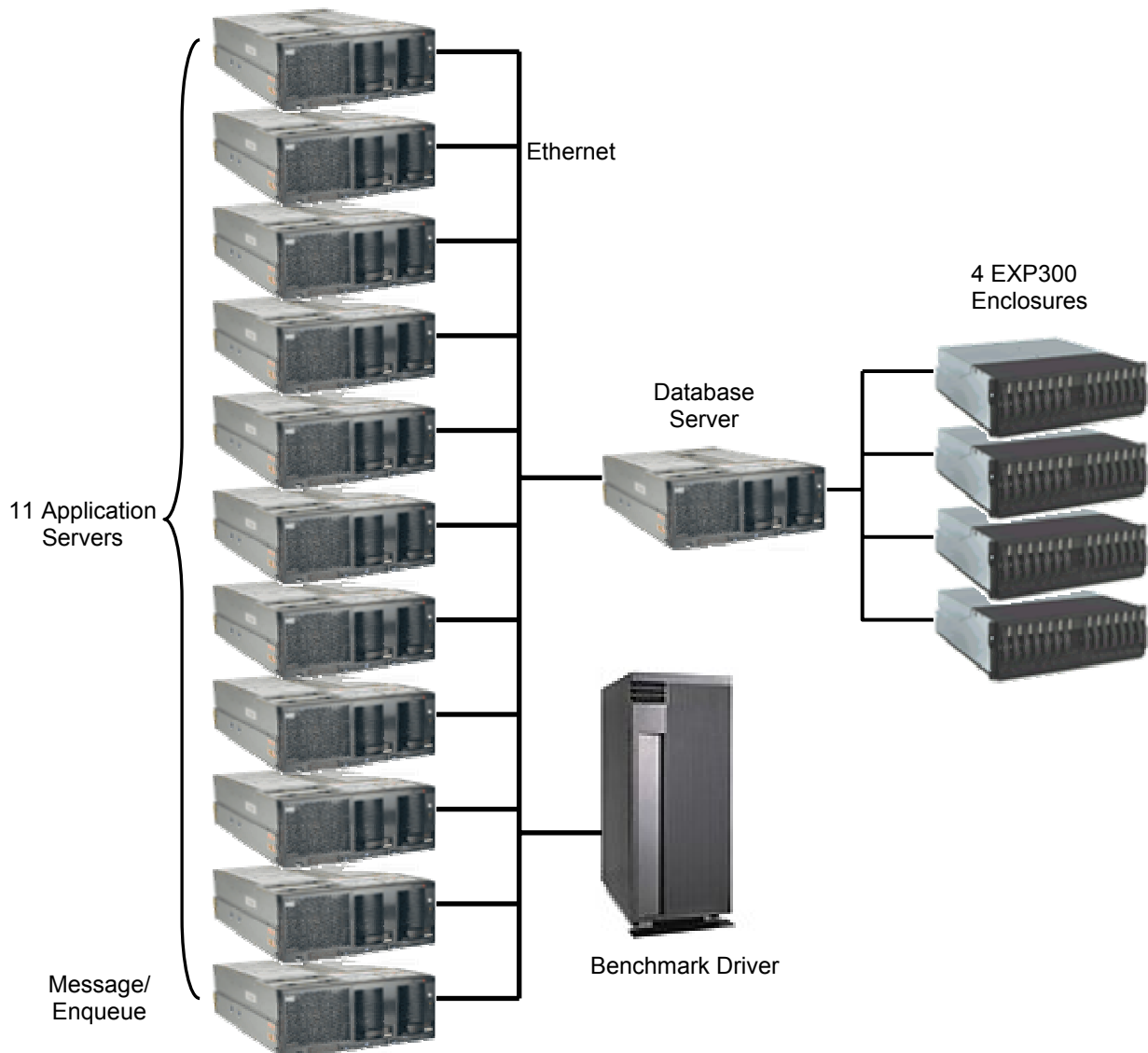

Figure 1: Hardware configuration for the three-tier SAP SD Benchmark system.

Server	Database server	SAP Message/Enqueue	SAP Dialog/Update server	Benchmark Driver
Server Name	walrus	penguin1	penguin2 - penguin8 penguin10 - penguin12	saph70
Model	IBM *e* server xSeries x440	IBM *e* server xSeries x440	IBM *e* server xSeries x440	IBM RS/6000 Enterprise Server Model H70
Processor type & clock rate	Intel® Xeon MP CPU/2GHz	Intel® Xeon MP CPU/2GHz	Intel® Xeon MP CPU/2GHz	340 MHz RS64 II 64-bit processors
Number of processors	4	4	8	4
Cache (level/size)	L2 cache 512KB L3 cache 2048KB	L2 cache 512KB L3 cache 2048KB	L2 cache 512KB L3 cache 2048KB	L2 cache 256KB
Memory (GB)	4	4	4	8
Operating System	SuSE Linux Enterprise Server 8.0	SuSE Linux Enterprise Server 8.0	SuSE Linux Enterprise Server 8.0	AIX 4.3.3
Total disk space (GB):	2052	36	36	54

Table 1: Server specification for the benchmark system.

The network configuration for this three-tier SAP SD Benchmark system is as follows:

Network configuration		
Type	Number of segments	Speed
Benchmark Driver to App Servers	1	100MB/s
App Server to DB Server	11 (one for each app server)	DB server 1000MB/sec (Fiber Gigabit) SAP app server 100MB/sec port

Table 2: Network configuration for servers.
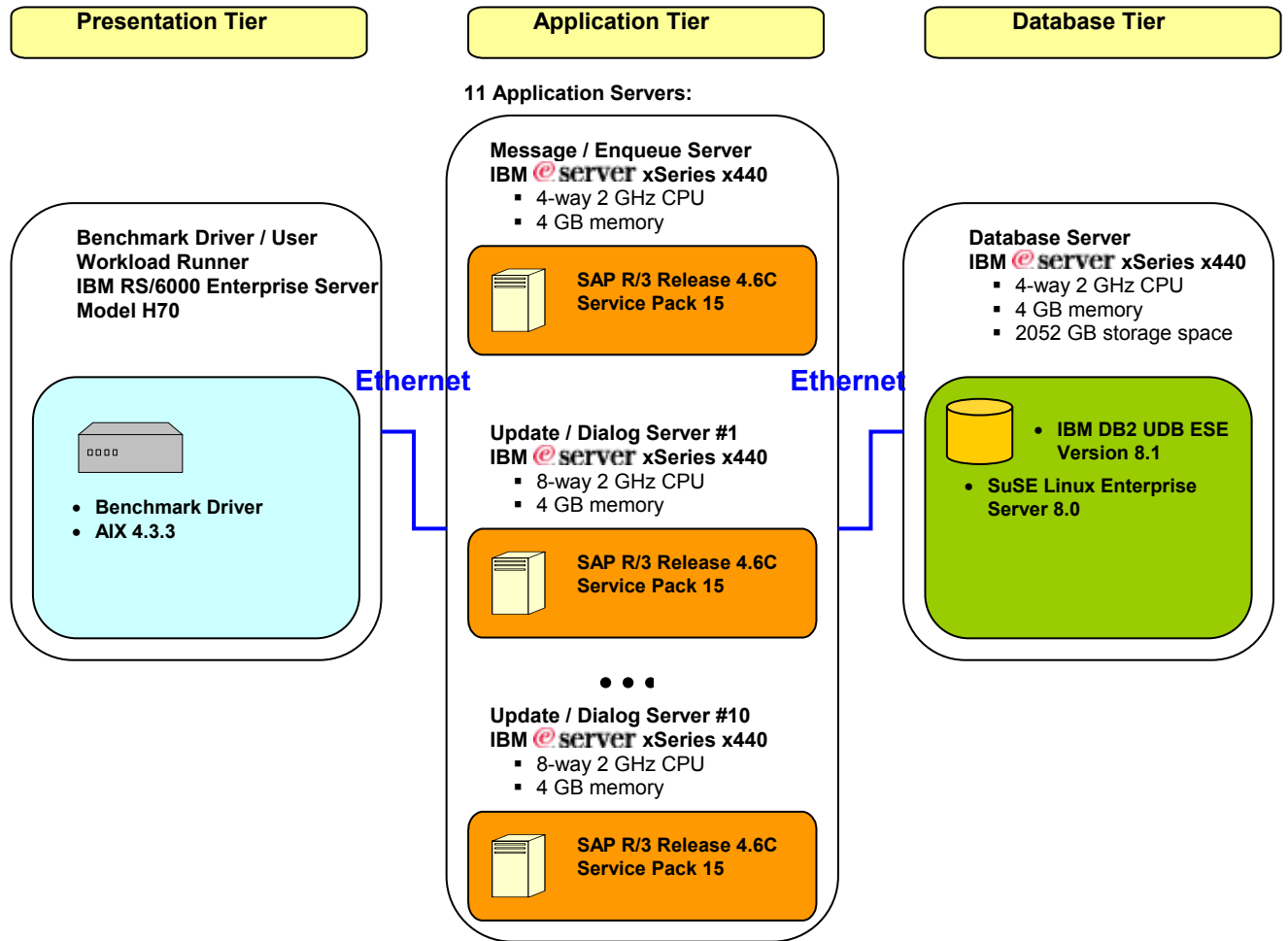
3.2 Software Configuration

Figure 2: Overview of SAP benchmark configuration.

The layout of the three-tier SAP benchmark system is illustrated in Figure 2. The presentation tier is the benchmark driver which simulates end users by driving business transactions in dialog steps.

The application tier contains the SAP R/3 business logic to implement its sales and distribution application. SAP R/3 systems are actually the middle servers.

The database tier is the core of the benchmark system. It provides high-performance, robust persistent storage and relational capabilities to the application tier. The database management system is DB2 UDB Version 8.1 for 32-bit Linux.

The operating system used for both database and application servers is SuSE Linux Enterprise Server (SLES) 8.0 with kernel level 2.4.19-SMP. The operating system for the benchmark driver machine is IBM AIX[®] 4.3.3.

4. System Tuning and Enhancements

Achieving good performance characteristics for the benchmark system involves proper configuration and extensive tuning exercises.   This section describes tuning exercises that were done to the SLES 8 operating system, the database, and SAP R/3.   The emphasis of the descriptions is on the rationale behind the tuning decisions. The following section on operating system enhancements presents Linux-specific issues that need to be enhanced in order to obtain better overall system performance.

4.1 Operating System Enhancements

4.1.1 Larger Shared Global Area

Memory is an important resource for database systems. Allocating more memory for database buffer pools improves database performance by increasing the amount of data cached and thus reducing the number of physical disk reads.   For Linux IA32 systems with more than 4GB of memory, the total size of DB2 UDB buffer pools is constrained by the process's virtual memory map. DB2 UDB attaches buffer pool pages to contiguous shared memory segments after the memory region allocated for shared libraries.   By default, shared libraries are mapped to a starting address defined by TASK_UNMAPPED_BASE (defined in the Linux source code to 0x40000000). This restricts DB2 UDB to use shared memory pages above address 0x50000000.

The SLES 8 distribution of Linux provides an interface to dynamically configure the memory mapped address of a process.   The kernel parameter, /proc/*<PID>*/mapped_base, can be tuned to alter the memory mapped address of the process, *PID*.   By default, the value of mapped_base corresponds to TASK_UNMAPPED_BASE (107374198 in decimal).   If the value of mapped_base is lowered to 0x20000000 (or 536870912 in decimal), DB2 UDB is able attach buffer pool pages starting at address 0x30000000. The gain in buffer pool size is illustrated in Figure 3. This change effectively gives DB2 UDB an additional 500MB of memory for buffer pools. DB2 UDB v8.1 FixPack 2 provides the functionality to automatically lower the value of mapped_base to 536870912 to make use of the additional memory.
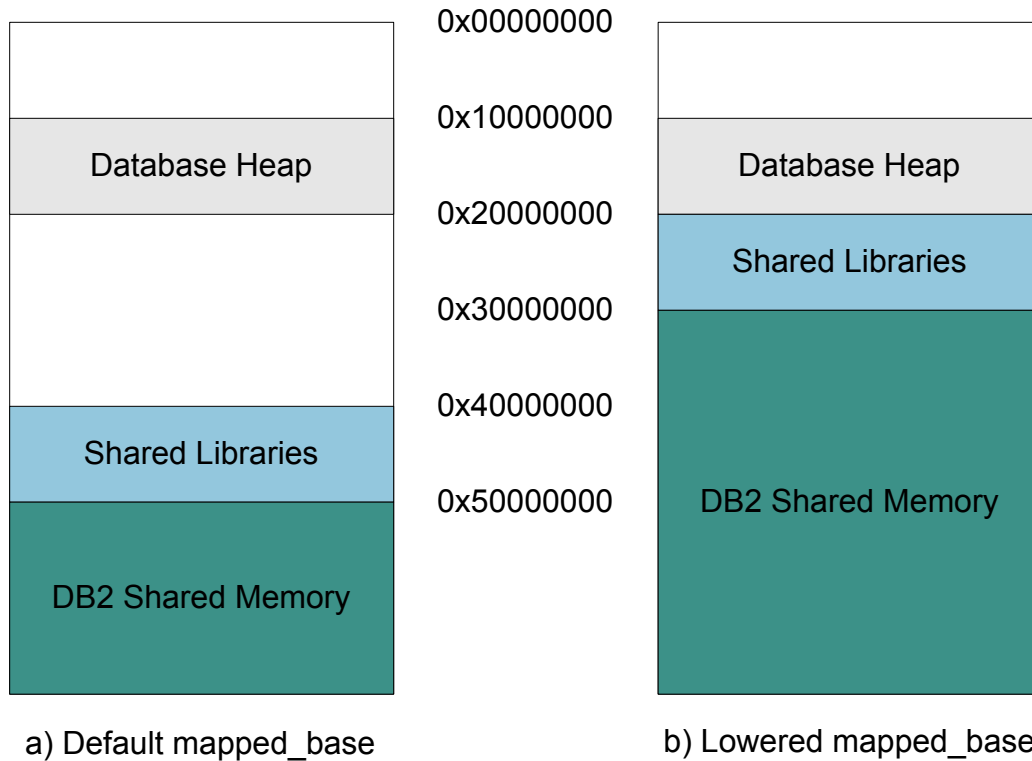
| 0x00000000 |
| 0x10000000 |
| 0x20000000 |
| 0x30000000 |
| 0x40000000 |
| 0x50000000 |

a) Default mapped_base

b) Lowered mapped_base

Figure 3: DB2 UDB process memory map.

4.1.2 Bounce Buffering

On an IA32-based Linux system with more than 1GB of physical memory, the first 1GB is referred to as *low memory*, and the memory above the 1GB range is referred to as *high memory*. In early versions of the Linux kernel, a PCI device could not perform a direct memory access (DMA) IO to high memory. Instead data transferred from high memory to the device had to be copied first to a "bounce buffer" in low memory before the DMA IO operation could occur. A bounce buffer was also required when data was transferred from the device to high memory. This copy between high memory and low memory, which is referred to as "bouncing" or "bounce buffering", can severely impact system performance for an IO intensive workload, such as the SAP SD Benchmark in two ways:

1. Large numbers of bounce buffers can create a shortage of memory.
2. Bouncing is done in kernel space, which adds an extra load on the system.

The SLES 8 distribution of Linux was implemented to eliminate bounce buffering. However, the implementation needs to be done on the PCI device driver as well as on the IO subsystem. During the benchmark execution, profiled data gathered on the database server indicated that the `ips` driver for the ServeRAID host adapters were using bounce buffers[2]. Inspection of the driver code revealed that

---

2An indication that bounce buffering is occurring is the presence of the function create_bounce() in a kernel profile.

support for high memory DMA was excluded. Obtaining an updated `ips` driver to eliminate bouncing resulted in a large performance improvement.

4.1.3 Page Size IO

Earlier Linux kernels also suffered from an inefficient raw interface. These kernels broke IO requests into 512 byte buffer heads and transferred data between the buffers and the device. For read requests, the buffer heads were coalesced back to the original IO request size before the data is returned to the user. The overhead of breaking and coalescing IO requests into 512 byte buffer heads can severely degrade system performance. Specifically, this implementation impacts the performance of a DB2 UDB database server where the bulk of the IO requests are done in a minimum of 4KB pages.

SLES 8 has optimized the raw interface to handle 4KB page size IO. That is, IO requests can now be broken into buffer heads whose size can be as large as 4KB. This greatly reduces the overhead and thereby improves IO performance. Unfortunately, like bounce buffering, support is also required by the device driver. And like bounce buffering, the ips driver did not have the necessary code to support page size IO. An updated `ips` driver which supported 4KB page size IO provided additional database performance improvements.

4.1.4 Network Subsystem Tuning

For a network-intensive workload like that used during the three-tier SAP SD Benchmark, it is vital that the network latencies are minimized. The Linux kernel provides several parameters that can be dynamically tuned to reduce the chance that the network layer becomes a bottleneck under heavy load. All network-related parameters are located in /proc/sys/net with TCP/IP specific parameters in /proc/sys/net/ipv4.

Parameters /proc/sys/net/core/wmem_max and /proc/sys/net/core/rmem_max control the maximum buffer size for the send and receive sockets respectively. Increasing these values to 256KB reduces the chance that DB2 UDB agents are blocked on a socket full condition.

Other parameters, /proc/sys/net/ipv4/tcp_sack and /proc/sys/net/ipv4/tcp_timestamps, are disabled to reduce the size of the TCP stack. The parameter, proc/sys/net/ipv4/tcp_sack, enables or disables the TCP selective acknowledgment (SACK) mechanism. This mechanism adds extra overhead to the TCP stack and does not provide any performance benefits for a reliable network.

4.2 Database Tuning

4.2.1 Table and Tablespace Configuration

A good tablespace design is required in order to achieve optimal database IO performance. Several design guidelines are outlined below:

1. Evenly distribute the tablespace across many physical disks.
2. When data pages are referenced infrequently, consider using raw device containers for the tablespace to avoid buffer cache misses and to take advantage of a more efficient raw IO interface. Conversely, if data pages are frequently referenced, use DMS file or SMS containers to leverage the file system buffer cache.

One of the database tuning activities in this benchmark was to identify transaction-intensive tables. These tables were separated from the predefined SAP tablespace and placed into DMS raw tablespaces. In addition to the performance gain of using raw device containers, the tables benefited from having their own buffer pool which could be finely tuned to improve buffer hit rates. Basically, these transaction-intensive tables can be classified into four categories:

1. VB tables (including VBMOD, VBHDR, and VBDATA),
2. Insert-only tables,
3. Volatile tables, and
4. Mixed tables.

VB tables are the most transaction-intensive tables for SAP SD Benchmark applications. The VBDATA, VBHDR and VBMOD tables were separated into tablespaces with their own buffer pools. The tablespace for the VBDATA table was created using 32KB pages since it has a VARCHAR field of 31000 bytes.

Insert-only tables are tables that have only insert transactions and few or no reads on the inserted data. Because of this characteristic, the buffer pool for these tables was small, thus freeing up more memory for other buffer pools that could benefit from higher cache hit rates. The tablespace for insert-only tables was created using 4KB pages.

Volatile tables are tables where inserts and deletes are performed during the course of the application, so the sizes of the tables fluctuate greatly during execution. The contents of these tables can vary from empty to very large at run time. The volatility in size of this type of table makes reliance on the statistics collected by RUNSTATS inaccurate. To prevent this, the tables are declared VOLATILE using the ALTER TABLE statement. With the table declared VOLATILE, the optimizer will always prefer to use an index scan rather than a table scan.

Finally, mixed tables are those tables that have a mixture of transactions. These include inserts, deletes, and selects.

4.2.2 Database Parameters

Table 3 lists the database parameters that have the most impact on the performance of an SAP database. These parameters are finely tuned to maximize database performance.

DBHEAP = 10000	Dbheap must be larger than LOGBUFSZ. We used the database system monitor to track the highest amount of memory that was used for the database heap.
LOGBUFSZ = 128	This is larger than the default size of 8 pages, but small enough that memory is not wasted here, and so can be put to better use in the bufferpools.
BUFFPAGE= 24900	BUFFPAGE is the size for IBMDEFAULTBP. All read-only tables were placed in IBMDEFAULTBP for this benchmark.
LOCKLIST = 5000	To choose the right size for LOCKLIST, we used the database system monitor to determine if lock escalations were occurring. LOCKLIST was set to the smallest size which avoided escalations.
SORTHEAP = 2000	The sort heap was kept small since there were no sort overflows reported in the database snapshot.
CHNGPGS_THRESH = 25	For an update-heavy transaction workload, we needed to make sure that there were enough clean pages in the buffer pool by setting the parameter to be lower than the default value.

NUM_IOCLEANERS = 6 NUM_IOSERVERS = 1	To determine the value of NUM_IOCLEANERS and NUM_IOSERVERS, the database system monitor helped tune this configuration parameter by providing information about write activity in the buffer pools: *The parameter can be increased if both of the following conditions are true:*
	    pool_data_writes > pool_async_data_writes
	    pool_index_writes > pool_async_index_writes.
	However, too many IOCLEANERS will impact the performance negatively.

Table 3: Key database parameters for an SAP database.

4.2.3 Buffer Pools

Access to data in main memory is normally 10 to 100 times faster than access to physical disks. If too little memory is available for the data volume, then data will be forced out of the buffer onto disk whenever new data is needed. Generally speaking, the higher the buffer pool hit ratio, the better the performance.

To achieve better performance, indexes and data should be buffered as much as possible. However, there is a size limit for total buffer pool allocation for 32bit DB2 UDB due to the constraint of address space on every 32bit operating system. Under this limitation, not all the indexes and data can be buffered into memory. To deal with this limitation, multiple buffer pools may be defined based on the data access characteristics of a table. Multiple buffer pools can be individually tuned to achieve better buffer pool hit rates for certain tables.

In this three-tier SAP SD Standard Application Benchmark, seven buffer pools were configured (see Table 4). To determine an appropriate size for each buffer pool, database snapshots were collected and the buffer pool hit rates were calculated. .

BPNAME	BUFFERPOOLID	NPAGES	PAGESIZE (K)
IBMDEFAULTBP	1	-1	4
INS4K	2	25000	4
VBDATABP	3	8000	32
MIX4K	4	90000	4
MIX4KIX	5	40000	4
VOL4K	6	11000	4
VBHDRBP	7	18000	4

Table 4: Buffer pool configuration.

4.3 Tuning of SAP R/3

The SAP R/3 software solution consists of numerous components such as R/3 work processes, R/3 buffers, and R/3 instances. The following section describes analysis and tuning of these components used to achieve optimal performance in the benchmark. These techniques are also applicable to production environments.

4.3.1 SAP R/3 Memory Configuration

Proper configuration of R/3 memory directly contributes to better SAP R/3 performance. The R/3 Memory Configuration Monitor can be used to optimally tune R/3 memory. To configure the R/3 memories properly, the following guidelines are used:

1. The hit ratio for the R/3 buffers should generally be 98% or better. Hit ratios of less than 98% can be regarded as acceptable only for the program buffer, the single record buffer, and the export/import buffer.

2. There should be no swaps (displacements) in the buffers of the R/3 system. If there are swaps, the memory size or the maximum number of entries should be increased.

3. For the R/3 roll area, the roll file should not be used. Typically, both rdisp/PG_SHM and rdisp/ROLL_MAXFS were set the same to avoid using the roll file.

4. There should be sufficient free extended memory. Experience shows that performance is dramatically reduced when R/3 extended memory is full. Since unused R/3 extended memory is swapped out by the operating system, an overly large extended memory does not adversely affect performance.

5. For the SAP SD Standard Application Benchmark, the R/3 profile parameters rdisp/PG_MAXFS and rdisp/PG_SHM are always set the same to make sure that R/3 does paging in memory instead of on disks.

4.3.2 Configuring the Number of Servers, Instances and Work processes

Even though an estimate of the total number of SAP R/3 servers must be made in conjunction with hardware configuration before the execution of a benchmark, the optimal number of SAP R/3 application servers might change during the course of benchmarking. The optimal number of servers was determined experimentally during this benchmark. Eleven SAP R/3 application servers were used for the final benchmark certification runs.

The number of R/3 instances per server depends on hardware configuration as well. For example, it is reasonable to install more than one R/3 instance on each server that has more than four processors and 4GB of main memory. With only one R/3 instance, we cannot effectively use the physical memory because of 32-bit address space constraints. However, too many instances on one server may cause performance degradation. Tuning experiments indicated that three instances on each dialog/update server gave the best results.

The proper configuration of the number of R/3 work processes per instance relies on analysis of the database request time, wait time of dispatcher, roll time and ABAP program buffer contentions. SAP R/3 ST03 monitor data can be used to help determine the number of processes on each instance. The statistics reported by ST03 can show the database request time and wait time for both update and dialog processes. Having too many R/3 work processes configured for one application server is likely to cause excessive wait time for dispatcher, or ABAP program buffer contention. Dispatcher wait time can be reduced by configuring fewer work processes per instance. However, if user requests are waiting for work

processes in the R/3 dispatcher queue, more work processes are necessary. To configure the optimal number of work processes, the following guidelines are considered:

1. Keep the number of R/3 update processes per instance as small as possible, as long as the update processes can keep pace with dialog processes.

2. Avoid configuring too many SAP R/3 work processes. These may reduce performance on the database server due to excess DB2 connections and the resources they consume.

In this benchmark, four dialog processes and one update process were configured on each dialog/update server.

4.3.3 Update Work Process Distribution

In SAP R/3, update work processes can be configured either on a dedicated server (an R/3 instance used mainly to host update processing) or be distributed evenly over all the application servers. A distributed update configuration was chosen for this benchmark because distributed update work processes can handle temporary load peaks better than a dedicated update server because they distribute the load over more than one server.

4.3.4 Number Range buffering

Number range object buffering optimizes the assignment of new document numbers during business processing. Buffering of number ranges increases document throughput by avoiding lock problems. R/3 offers two forms of buffering for number ranges—buffering in main memory and local file buffering. Local file buffering uses an additional database table called NRIV_LOKAL. Ranges RF_ BELEG, MATBELEG, KONV, AENDBELEG and RV_BELEG were buffered in main memory with the following sizes:

RV_BELEG   9000
MATBELEG   3000
AENDBELEG 3000
KONV   3000

4.3.5 Workload Analysis of SAP SD Standard Application Benchmark

4.3.5.1 Workload Monitor

Performing a workload analysis provides reliable data on throughput, load, and response times for the R/3 solution-based system and its components. The workload monitor provides a comprehensive overview of load distribution within the SAP R/3-based system. The initial screen of the Workload Monitor is accessed via Transaction Code ST03. The following criteria may help to identify possible performance bottlenecks:

1. **Dispatcher wait time is much higher than 50 ms**. This implies that too few work processes have been configured for the instance.

2. **Database request time is much greater than 400ms**. This indicates that either the database has a performance bottleneck or too many SAP R/3 processes in total have been configured and are causing contention in the database.

3. **The enqueue time is more than 5ms**. This indicates that enqueue is a bottleneck. The number of enqueue processes may need to be increased, or the network communication may need to be sped up.

4. **Roll in time and roll out time are much higher than 20 ms**. This signals that the R/3 roll buffer or the R/3 extended memory is too small.

4.3.5.2 Identifying Expensive SQL Statements

Expensive SQL statements are main causes for performance problems. Some hold locks for longer duration which causes lock contention. SAP R/3 offers a feature called SQL trace to identify those expensive SQL statements. To collect the SQL trace data, call transaction code ST05 from SAPGUI. By sorting the data in the compressed summary according to the elapsed database time from the SAPGUI menu, it is easy to identify the tables that were accessed for the longest time. Together with this facility, DB2 UDB snapshots on the database can be used to identify the performance bottleneck at the statement and table level.

5. Results

The benchmark result is shown in Table 5. The official certification of this benchmark is shown in Appendix A.

Component	SAP R/3 4.6C, Sales and Distribution
Number of simulated benchmark users	5790
Number of dialog steps / hour or SAPS for SD	1,743,000 DS/h
Average dialog response time	1.960s
Average CPU utilization of Application Servers	60% (dia/upd: 61%, msg/enq: 4%)
Average CPU utilization of Database Server	94%
Average Database request times	D: 319ms /U: 83ms
Fully Processed Order Line items / hour	581,000
SAPS:	29,050

Table 5: Results of the three-tier SAP SD Standard Application Benchmark.

This SAP SD Standard Application Benchmark result remains the highest of all similar three-tier SAP SD Benchmarks published on Linux[3]. The combination of DB2 UDB and the x440 outperformed other vendors with same number of database server CPUs by a significant margin. This benchmark shows that IBM x440 with Intel Xeon MP 2.0GHz processors and the IBM DB2 UDB server provides superior performance and data processing capacity.

This benchmark is another strong testament to IBM's commitment to provide enterprise class database software on Linux. DB2 software for Linux means:

- Flexibility – through support of open standards
- Reliability – from the proven technology of DB2 Universal Database
- Cost effectiveness – cross-platform portability protects skills investments and facilitates hardware choices
- High performance – the current benchmark result

---

[3] As of December 3, 2003

6. Summary

This paper presented an overview of the recent three-tier SAP SD Standard Application Benchmark, including both hardware and software configurations. Tuning steps for Linux, DB2 and SAP R/3 were explained, illustrating how this excellent result was achieved.

The result of this benchmark shows that the combination of DB2 UDB, SAP R/3, IBM @server xSeries servers and Linux can provide superior performance and capacity. Customers with increasing IT demands can be confident as their business grows that DB2 UDB V8.1 will continue to meet their needs. Backed by the power and scalability of Linux, IBM DB2 UDB is the database of choice for any mission-critical application.

7. Appendix A. Certification of This SAP SD Standard Application Benchmark by SAP AG

CERTIFICATION
SAP Standard Application Benchmarks

The SAP SD Standard Application Benchmark performed on March 15, 2003,
by IBM in Toronto, Canada was certified on March 17, 2003, with the following data:

Number of benchmark users & comp.:	5,790 SD (Sales & Distribution)
Average dialog response time:	1.96 seconds
Throughput:	
Fully Processed Order Line items / hour:	581,000
Dialog steps / hour:	1,743,000
SAPS:	29,050
Average DB request time (dia/upd):	0.319 sec / 0.083 sec
CPU utilization of database server:	94%
CPU utilization of application servers:	60% (dia/upd: 61%, msg/enq: 4%)
Operating System all servers:	SuSE Linux Enterprise Server 8.0
RDBMS database server:	DB2 8.1
R/3 Release:	4.6 C
Total disk space:	2,052 GB

Certification Number: 2003018

1. Database server: IBM eServer xSeries x440, 4-way SMP, Intel Xeon MP, 2 GHz 4-way 2 GHz CPU
2. 4 GB memory
3. 2052 GB storage space

11 application servers:
4. 10 Dialog/4-way 2 GHz CPU
5. 4 GB memory

xSeries x440, 8-way SMP, Intel Xeon MP, 2 GHz, 512 KB L2 cache, 2 MB L3 cache, 4 GB
main memory

1 Message/enq. server: IBM eServer xSeries x440, 4-way SMP, Intel Xeon MP, 2 GHz,
512 KB L2 cache, 2 MB L3 cache, 4 GB main memory

Certification Number. 2003018

SAP™