ZACHMANNTEST - A SYNTHETIC MEMORY BENCHMARK FOR SAP ERP SYSTEMS

André Bögelsack, Holger Wittges, Helmut Krcmar
Technische Universitaet Muenchen, Boltzmannstraße 3, Garching, Germany

Hannes Kühnemund
SAP AG, Walldorf, Germany

Keywords: SAP ERP system, Benchmark, Main memory, Zachmann test, Performance measurement.

Abstract: SAP ERP systems are the backbone of today's enterprises and business processes. The technical performance of such systems directly influences the business performance. Estimating or measuring the performance of SAP ERP systems is a hard task due the diversity of the measurement process. We present a synthetic benchmark called Zachmanntest which measures the performance of the SAP ERP system focusing on main memory operations. The internal flow logic is presented as well as the usage of the Zachmanntest. Results of the peak performance measurement and scalability of heterogeneous servers are presented.

1 INTRODUCTION

SAP Enterprise Resource Planning systems are the backbone of today's business processes (Krcmar 2009). The performance of such systems directly influences the performance of the core businesses: e.g. a long response time results in a decreased business performance due to the increased pass-through time.

Testing and quantifying the performance of a SAP ERP system is a hard task to fulfill. There are standardized SAP application benchmarks available for testing the performance for either the application server or the database management system (SAP 2010). Drawbacks of these benchmarks are that they are hard to implement and the results from benchmark runs on different hardware platforms cannot be compared adequately. Another drawback is that the benchmarks only aim at one specific functional unit in the SAP system, e.g. the SAP SD benchmark for sales and distribution operations.

However, this paper postulates the need for a more technical benchmark, which is independent from the SAP business processes and SAP system. The need for such a technical benchmark derives from the fact that a company's business processes are usually completely different from the business processes which are processed during an application benchmark. So the results of the application benchmark runs may not be meaningful enough for the daily performance of a SAP ERP system. The proposed solution is the introduction of a synthetic benchmark for SAP ERP systems, which we call Zachmanntest. The name derives from the original developer of the test.

The target audience of this paper and the new benchmark can be divided into two groups. First, the benchmark presented here can help hardware and operating system vendors to verify and compare the performance capability of their servers. In fact, we show some results for such a case. Second, existing and new SAP customers can easily apply the Zachmanntest into the existing or a test SAP ERP system and run benchmarks for estimating and comparing the performance results against other configurations or hardware.

Overall the paper makes the following contributions:

- It presents a main memory benchmark for the operations of a SAP ERP system when working with buffers.
- It explains the benchmark in detail and explains the performance metric.

Bögelsack A., Wittges H., Krcmar H. and Kühnemund H..
ZACHMANNTEST - A SYNTHETIC MEMORY BENCHMARK FOR SAP ERP SYSTEMS.
DOI: 10.5220/0003437903480355
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 348-355
ISBN: 978-989-8425-53-9

- The paper shows the application of the benchmark for two purposes: a performance comparison and the exploration of the scalability of a SAP ERP system.

The rest of the paper is organized as follows: section 2 explains the architecture of a SAP ERP system. In section 3 we discuss how SAP benchmarking is currently done in the real world and highlight some difficulties, which lead to the development of a new synthetic benchmark. Chapter 4 focuses on the derivation of the benchmark requirements, which are later on used to check the alignment of the Zachmanntest. Chapter 5 presents several performance results and we show for what purpose the Zachmanntest can be utilized. Chapter 6 is designated to the limits of the Zachmanntest and chapter 7 summarizes the paper and provides an outlook.

2 SAP ARCHITECTURE

The SAP ERP system uses a typical three layer architecture consisting of a graphical user interface, an application server (AS) and a database management system (DBMS). All of the components can but do not have to be on different servers.

The graphical user interface, called SAPGui, is part of a fat client installation on the user's PC. As the performance of the SAPGui is not the limiting factor for the performance of the SAP ERP system (Schneider 2005), it will not be considered anymore in this paper.

Actually, the performance of a SAP ERP system is mainly influenced by the performance of the AS and DBMS. The application server is developed and delivered by SAP itself, whereas the database management system can be chosen by the customer from a list of fully supported and certified DBMS vendors, e.g. IBM DB2, SAP MaxDB or Oracle. Due to the fact that the architecture and performance factors of DBMS are completely different, the DBMS will not be considered anymore in this paper. For testing and verifying the performance of DBMS there are other more powerful benchmarks available (Doppelhammer et al. 1997).

The application server itself is based on a SAP kernel, which is a huge set of C programmed executables (Gradl et al. 2009). The SAP kernel contains the following processes (SAP 2001): a dispatcher process for distributing the workload to the so called work processes for short term user requests and batch process for long term user requests. All processes use one central enqueue process. Besides there are also gateway processes, several spooling processes, update processes and message server processes. Coupling all these processes together form the central instance of a SAP ERP system (also known as the application server). In order to handle a lot of user requests and user input such a central instance can be supported by dialog instances on satellite servers. Dialog instances provide the same services for SAP users as the central instance does with one except of the Enqueue server.

As mentioned before all of the processes are C programmed executables, which use the Inter Process Communication (IPC) features of the specific operating system. In fact, two mechanisms are used intensively: semaphores and shared memory segments. For exchanging data between several processes shared memory segments are used for storing and manipulating data, which is used by all process (Bögelsack et al. 2010a; SAP 2001). For example, for buffering often used database data, the SAP ERP system uses internal buffers by creating huge shared memory segments, which are used by all processes. That way, it is possible to store frequently accessed data from the database in the main memory of the application server instead of accessing it from the database several times.

3 SAP BENCHMARKING

3.1 Application Benchmarks

For a lot of different SAP systems, not for all, SAP defines application benchmarks (see (SAP 2010)). The most popular one is probably the SD benchmark for SAP Enterprise Resource Planning systems (Lober, Marquard 2003). The SD benchmark defines typical sales and distribution activities in the SAP system. Those activities produce a heavy workload on the SAP ERP system. The SD benchmark is still used to determine the overall performance of a SAP ERP system and the underlying hardware. Moreover, it can be used to compare server's performance of different hardware vendors. Besides the SD benchmark there are several other benchmarks for other dedicated SAP systems available.

One major is the fact that the benchmarks only focus on a small set of functions/programs of the entire SAP system (Marquard, Götz 2008). To be clear: a SAP ERP system consists of more than 160,000 programs, whereas the SD benchmark only tests 6 of them. But based on the results from those 6 programs, the performance of the entire server hardware is deduced.

Another drawback is the ability of tuning the benchmarks to the maximum of the possible performance. Of course this is suitable and reasonable for the hardware vendors, the results from those test runs can never be achieved in the real world as customers are not able to tune the application in this manner. During the performance benchmarks of the hardware vendors all SD benchmark activities are performed in the main memory of the SAP ERP system, whereas a typical customer installation of the SAP ERP systems shows a completely different performance behavior.

In summary, SAP offers several application benchmarks, but the usage of those benchmarks is critical, as performance results usually are not valid for real world installations and are not comparable due the fact of very high tunable components and the hardware/software variability.

3.2 Synthetic Benchmarks

In order to solve the problems of the application benchmarks, usually performance measurements are made with synthetic benchmarks. A synthetic benchmark is characterized as a fixed sequence of functions or programs . By executing those functions or programs the performance of a software system can be evaluated. The main advantage of such synthetic benchmarks is the comparability of the results.

In the SAP ecosystem there are currently no SAP-specific synthetic benchmarks available. In the most of the cases synthetic benchmarks from other domains are used to measure some very specific performance aspects. For example, (Kemper et al. 1999) used the TPC-C benchmark for evaluating the performance of the underlying database management system. The TPC-C benchmark has very close regulation how to perform the benchmark and which data set to use etc (Poess, Floyd 2000). However, besides of this example, there are no synthetic benchmarks available.

In order to solve the problems of the application benchmarks, we argue that the development of a synthetic benchmark for the performance of the application server can solve the issue. For testing the databases performance, some synthetic benchmarks are available. For the synthetic performance measurement of an application server there is a gap.

4 BENCHMARK REQUIREMENTS

Synthetic benchmarks try to solve the problems, arose with the application benchmarks. A SAP-specific, application server focused, synthetic benchmark must meet the following high-level requirements:

- comparable benchmarks between different SAP systems and hardware platforms
- valid results for real world SAP installations
- valid results for the performance of the entire SAP system

The development of a new synthetic benchmark for the application server of SAP ERP systems is quite difficult, as the performance is not only affected by one factor, but by many. Overall performance is influenced by the performance of the following three components: CPU, I/O and main memory (Huang et al. 2006).

When creating a synthetic benchmark for the application server, the workload must be analyzed in order to be clear on which performance factor to focus. In a large-scale SAP installation, we analyzed the workload of the application servers. All activities of the application servers are not I/O-bounded. This is because the application servers do not request data from disk or store data to disk. Usually, this is the task of the database management system. Therefore the factor I/O can be neglected. All activities of the application server consume CPU cycles. But in a real world SAP system, the consumed CPU time of the application server is 3x less than of the database (Schneider 2005). In fact, very often the application servers do not consume more than 10% CPU time. Hence, CPU as a performance factor can be neglected too. At least there is the performance of the main memory. The application server frequently stores often used data into the main memory of the server. Moreover, it consumes shared memory segments for storing user context, inter-user objects and temporary used data (Bögelsack et al. 2010b). Very often it can be seen, that main memory performance is the key performance factor for the overall performance of the application server. Therefore, a synthetic benchmark for the application server should mainly focus on the main memory performance.

5 ZACHMANNTEST

We developed a synthetic benchmark, called the

Zachmanntest. The functionality of the test includes the performance measurement of main memory activities of the SAP application server.

5.1 Zachmanntest Architecture

The Zachmanntest consists of two programs. The first one is an easy to use entry mask to specify some parameters of the Zachmanntest. The second one is the workhorse of the test, which produces a lot of main memory operations in the application server. In fact, those main memory operations are operations on so called internal tables. Those internal tables are two dimensional arrays and representatives of real existing tables from the database. In real life operations, internal tables are used to access often used data sets. That way, database accesses and traffic is reduced. Those internal tables store the same data as the corresponding database table, but are created and maintained by the application server. Each program of the SAP ERP system, which is somehow interacting with the database management system and stores/reads data from it, uses this concept. So from our point of view, this operation is a universal one and fits to the approach of a synthetic benchmark. A synthetic benchmark requires a specific sequence of operations/programs to be executed during runtime (Curnow, Wichmann 1976). This is achieved by specifying the following steps during the execution. Please note, that we used pseudo-code instead of ABAP statements:

```
While time < max_run
 Create internal table
 Fill internal table with data
 While iteration < loop_cnt
        Randomly select data set
     Read selected data set
     Increase throughput counter
 Endwhile
 Delete internal table
Endwhile
Print throughput counter
```

The value max_run defines the run time after which the execution of the Zachmanntest is aborted. The value loop_cnt defines a numerical value for how often the internal table should be cycled.

By executing the entire Zachmanntest, one instance of the workhorse is instantiated and handled by one work process (see section 1). The Zachmanntest produces a heavy load on the main memory of the application server. So, one Zachmanntest can be interpreted as one power user in the SAP ERP system (stressing the application server with heavy main memory activities).

5.2 Zachmanntest Performance Metric

The Zachmanntest has to quantify the performance of the underlying main memory from a SAP perspective in some way. Generally, there are several performance metrics available, e.g. response time metrics or throughput metrics. However, none of them fully fits to the Zachmanntest. For example, response time is not an adequate metric for quantifying how many table entries were accessed and how fast the internal tables were created.

The performance metric of the Zachmanntest is throughput, measured in rows per seconds. For example, after finishing one run of one Zachmanntest, you get the throughput result for the SAP ERP system about 9,000 rows per second. The meaning of this metric is the following: for the case of one power user in the SAP ERP system, approx. 9,000 rows per second can be accessed by one work process for this dedicated power user. When handling two power users in the SAP ERP system (two Zachmanntests) the throughput might be less or equal. This is because the maximum available throughput will be shared between both power users.

5.3 Zachmanntest Parameters

Running one instance of the Zachmanntest, results in one performance value. This is usually only the beginning of a larger performance measurement run where not only the peak performance of interest, but also the scalability etc. Therefore the Zachmanntest can be parameterized. The parameters are:

- Group (group)
- Number of instances of Zachmanntest (number)
- Runtime (time)
- Size of the memory to be allocated (size)
- Loop count (loop_cnt)
- Wait time (wait_time)

The parameters were introduced in order to emulate different workload situations. The meaning of the parameters is as follows. The parameter "group" can be used to group several instances of Zachmanntests together. This is useful, when emulating a power user with several connections to the SAP system. If Zachmanntest instances are not grouped together, several power users, each with a single connection to the SAP system, are emulated. By changing the parameter "number", many instances of the Zachmanntest can be created at once. Increasing the number of Zachmanntests results in a higher workload on the server. In order to eliminate temporary effects on the execution of the Zachmanntest, the runtime (parameter "time") of each benchmark run can be specified. On this way, temporary effects can be eliminated with a longer runtime, but short measurements are also possible with smaller runtimes. Moreover, when using this parameter it is possible to emulate short term and long term user requests. The size of the allocated memory is a quite important parameter ("size"). Each instance of the Zachmanntest creates an internal table with this size. So each instance can allocate only a small amount of memory (KB) or a large amount of memory (MB). By increasing the number of Zachmanntest instances the overall amount of allocated memory can be easily increased and hit the machine limit. The parameter "loop_cnt" defines a numerical value for how often the internal table should be cycled. The more cycles, the more time is spent in the loops. Less cycles result in less time spent in the loop. By varying this parameter different load patterns can be realized: short time accesses or long time accesses. By using the parameter "wait_time" a small wait time can be set e.g. in order to have a cool-down-phase. Especially during long term workload run times, this wait time may be suitable to vary the workload on the server.

5.4 Zachmann Workload Characteristics

The Zachmanntest aims on quantifying the performance of the underlying hardware by accessing the shared memory segments extensively. So the Zachmanntest can be characterized as a maximum throughput benchmark for the shared memory speed of the underlying server. However, characterizing the workload in a more accurate way is necessary. When starting the Zachmanntest, the resulting workload strongly depends on the underlying hardware and operating system. It can be discovered, that there are quite big differences between the operating systems like UNIX derivates or Microsoft Windows. The reason for this is either the implementation of the inter process communication (IPC) facilities or the implementation of the memory access for the SAP ERP system.

Moreover the workload depends on the underlying hardware. There are several hardware vendors in the market and the most of them offer servers based on Intel and AMD CPUs. Some vendors offer alternatives, like IBM with the Power CPU, Oracle with the SPARC CPU or Hewlett-Packard with the PA-RISC CPU. Besides, we see great differences between cache-coherent non-uniformed memory architecture (ccNUMA) servers and uniformed memory architecture (UMA) servers: the higher the workload (~40 Zachmanntests in parallel) the better is NUMA compared to UMA.

In summary, each instance of the Zachmanntest aims on a throughput maximization with a high CPU workload.

6 RESULTS

We argued that a synthetic benchmark can estimate the performance of the SAP ERP system in a more adequate way. By measuring the performance of several different SAP ERP systems, we are able to proof this statement.

6.1 Comparison between Several Servers

We have done several performance benchmarks with different hardware platforms and servers. Measuring the performance did not only include the peak performance, but also the scalability of the underlying hardware and software configuration. It turns out, that utilizing the Zachmanntest enables us to compare the performance of different hardware platforms and different servers. In Figure 1 the throughput of one instance of the Zachmanntest on different hardware platforms is shown, whereas the X-axis shows the hardware platforms and the Y-axis shows the throughput in rows/second. The concrete values for the performance testing can be found in Table 1. It can be seen, that some of the results are somehow surprising. We tested four different servers with different hardware configurations and CPU architecture. The server range covered small x86-based servers and server with PA-RISC processors for large-scale SAP installations. For each of the server configuration we noted the clock speed of the processor and the architecture of it. It is surprising to see, that for the Zachmanntest results are closely together. A PA-RISC processor with only 1.0GHz (server C) nearly reaches the same performance result as an AMD Opteron@2.8GHz (server A) and or an Intel Itanium CPU@1.6GHz (server B). This is surprising since the difference between both servers is only a slight one with ~1.5%.

Taking into account that the PA-RISC processor from server C has a clock speed of 1.0GHz and server A has a processor with a clock speed of 2.8GHz this result is surprising. Everyone would expect to see a significant better performance on server A than on server C. Looking on the results

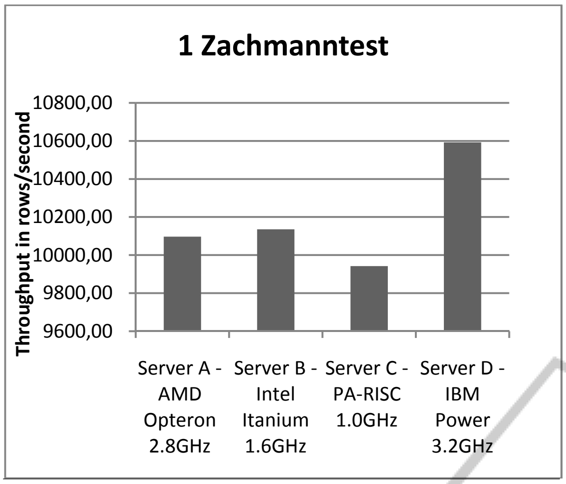

Figure 1: Throughput results for different hardware platforms.

Table 1: Results of performance measurement.

	1 Zachmanntest	Change in percent	Start of production year
Server A - AMD Opteron 2.8GHz	10096,90	+1,56	2006
Server B - Intel Itanium 1.6GHz	10135,70	+1,95	2009
Server C - PA-RISC 1.0GHz	9941,97	baseline	2003
Server D - IBM Power 3.2GHz	10592,01	+6,54	2010

from server D it is not surprising at all, that the 3.2GHz fast IBM Power CPU wins this benchmark. It is one of the newest CPU on the market and the CPU with the highest clock speed.

The reasons for the surprising results lie in the architecture of the CPUs. All tested CPUs use direct connected memory instead of bus-connected memory. On such memory the CPU can operate 1) directly without any inter-bus system and 2) with high speed. Moreover, the connection throughput between the CPU and the connected memory of a PA-RISC and an AMD Opteron CPU does not differ in such a great way. As the measurement with one Zachmanntest represents a peak performance test, there is any easy rule to describe the results: the higher the clock speed, the better the result.

In summary we have shown, how to compare the performance of different hardware platforms and CPU generations by using the Zachmanntest. The results show, that comparing the clock speed of a CPU can be misleading for performance estimation of a SAP ERP system. The specifics of CPU architecture must be taken into account too, when dealing with SAP ERP systems performance. This especially true for peak performance measurements but we expect to see different results for scalability measurements.

6.2 Scalability

The scalability of a SAP ERP system is the ability to provide a better performance when more resources are added to it or to provide the same performance to all users, when more users log into the system with steady resources.

In this section we present test results from two servers: server A is based on 48 Intel Itanium CPUs@1.6GHz and server B, which is based on 32 IBM Power 750 CPUs@3.2GHz. Please note, that those two servers are different ones than those we tested in the previous subsection. Taking only the CPU clock speed and the number of CPUs into account, one may come to the conclusion that server B outperforms server A because of the clock speed, but server A may show a better scalability because of the large number of CPUs.

Results for this test are illustrated in Figure 2, whereas the X-axis shows the number of parallel Zachmanntests and the Y-axis shows the throughput in rows/second.

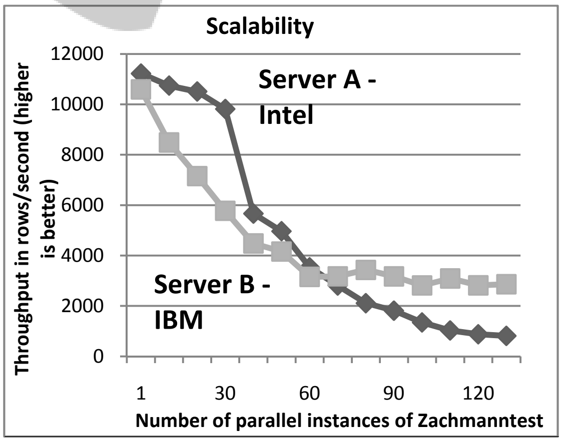

Figure 2: Scalability of two servers.

When dealing with the scalability of a SAP ERP system in both servers, one would guess to see a better scalability of the server B, because of the higher clock speed. But surprisingly this is not true. Instead, the peak performance of the Intel-based SAP ERP system is higher than for the IBM-based system. Of course, this situation is only true for a small part of the test. During the execution (at ~70 parallel Zachmanntests) both SAP ERP systems gain the same throughput. After that point in execution the IBM-based SAP ERP system shows a better throughput. Moreover, the throughput seems to be constant over the time. Please note, that the workload on the entire server is increased with the number of Zachmanntests. The performance of the Intel-based SAP ERP systems decreases dramatically. This scalability behavior is not relevant for small SAP ERP systems with 10-70 concurrent users, but for large scale SAP ERP systems installations with more than 1,000 users this become a big drawback

Without the ability of the Zachmanntest to run in multiple groups and to instantiate multiple times, the discovery of the bad scalability behavior of the Intel-based SAP ERP system would be still hidden.

6.3 Requirements Alignment

In chapter 4 we defined some high level requirements for a synthetic benchmark in the SAP environment:

- Comparable benchmarks between different SAP systems and hardware platforms
- Valid results for real world SAP installations
- Valid results for the performance of the entire SAP system

After presenting some testing results we are able to compare the abilities of the Zachmanntest with the requirements we defined.

First, a synthetic benchmark should deliver comparable results for different hardware and server vendors. By gathering a lot of results from different SAP ERP systems, installed on different platforms, we are able to verify this requirement. The Zachmanntest is able to quantify the performance in a simple way and make different hardware platforms comparable. As soon as a SAP ERP system is installed on the testing server, the Zachmanntest can be applied to the system and performance results can be gained.

Second, a synthetic benchmark should deliver valid results for real world SAP installations. We understand real world results as results, which can also be measured in SAP ERP systems outside the laboratory environment without any adoption to the benchmark specialties. In several projects outside of the scientific/university context the Zachmanntest was used to quantify and compare performance for several different servers. All gained performance results from section 6.1 and 6.2 were collected in real world projects. That way, we proof the real world appliance of our test.

Third, the Zachmanntest was intended to deliver performance values for the entire SAP ERP system. This is not true for the Zachmanntest as it only aims on the performance of the application server. In order to reach a comprehensive performance benchmark for the entire SAP ERP system a synthetic database benchmark must be applied, too.

In summary, the Zachmanntest only meets two of the three high level requirements. Nevertheless, it fulfills the two most important requirements in a way that we can easily utilize the synthetic benchmark for the performance measurement. Compared to the application benchmarks the Zachmanntest fulfills more high level requirements.

7 LIMITATIONS

The presented Zachmanntest is a synthetic, main memory oriented benchmark, which is able to measure the performance of a SAP ERP system, when working in the shared memory segments of the underlying operating system/hardware. Although the test shows a wide field of application, it has some limitations.

One major drawback is the fact that it is not used by the hardware or software vendors for quantifying the performance of their servers. There are no comparable results available and customers are forced to run their own benchmarks. This is not only a drawback of this specific benchmark it is also true for SPEC benchmarks.

Another drawback of this test is the focus on the shared memory operations. Besides these operations there are other operations, which might be important for the performance of a SAP ERP system. However, it is simple to extend the test with additional test methods/functions/workhorses.

8 CONCLUSIONS

In this paper we present the Zachmanntest, a synthetic benchmark, which quantifies the performance of a SAP ERP system when working on the shared memory. In the SAP ecosystem there are application and synthetic benchmarks available. Regarding the application benchmarks the implementation and interpretation of the performance results always has some drawbacks. Therefore we decided to implement a synthetic benchmark.

The Zachmanntest consists of a sequence of SAP internal functions. Those functions are broadly used in the SAP ERP system by a great number of SAP programs. In order to recreate different workload situations, the Zachmanntest can be parameterized. This ensures the special adoption of the workload situation for a real world scenario.

In a last step we present results from performance measurements. We used the Zachmanntest on two ways: first, to demonstrate the comparability of different hardware platforms and second, to demonstrate the scalability factor of two different servers in order to elaborate the entire performance behavior. By showing the results of the performance runs, we are able to proof the requirements alignment of the Zachmanntest to some high level requirements we derived from the discussion of the application benchmarks. It turns out, that two of the three requirements are met.

In order to fulfill all three requirements the Zachmanntest must be extended. Till today it is a synthetic benchmark for the application server only and neglects the performance of the database server. For measuring the performance of the entire SAP ERP system, the synthetic benchmark must be extended for the database with some synthetic database specific operations. This will be our next steps in making the benchmark more pleasant and useful for other performance engineers.

REFERENCES

Bögelsack, A.; Krcmar, H.; Wittges, H. (2010a): Performance Overhead of Paravirtualization on an exemplary ERP system. *Paper presented at the 12th International Conference on Enterprise Information Systems, Funchal, Portugal.*

Bögelsack, A.; Wittges, H.; Krcmar, H. (2010b): *Virtualisierung von SAP-Systemen. Galileo Press 2010b.*

Curnow, H.; Wichmann, B. (1976): A synthetic benchmark. In: *The Computer Journal, Vol. 19 (1976) No. 1, p 43.*

Doppelhammer, J.; Höppler, T.; Kemper, A.; Kossmann, D. (1997): Database performance in the real world: TPC-D and SAP R/3. Paper presented at the *Proceedings of the 1997 ACM SIGMOD international conference on Management of data, Tucson, Arizona, United States.*

Gradl, S.; Bögelsack, A.; Wittges, H.; Krcmar, H. (2009): Layered Queuing Networks for Simulating Enterprise Resource Planning Systems. *Paper presented at the 6th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems, Milano, Italy.*

Huang, W.; Liu, J.; Abali, B.; Panda, D.K. (2006): A case for high performance computing with virtual machines. Paper presented at the *Proceedings of the 20th annual international conference on Supercomputing, Cairns, Queensland, Australia.*

Kemper, A.; Kossmann, D.; Zeller, B. (1999): Performance Tuning for SAP R/3. In: *Bulletin of the Technical Committee on Data Engineering, Vol. 22 (1999) No. 2.*

Krcmar, H. (2009): Informationsmanagement. *Springer-Verlag New York, Inc. 2009.*

Lober, B.; Marquard, U. (2003): *Anwendungs- und Datenbank-Benchmarking im Hochleistungsbereich von ERP-Systemen and Beispiel von SAP. In: Datenbank-Spektrum, Vol. 7 (2003), pp. 6-12.*

Marquard, U.; Götz, C. (2008): *SAP Standard Application Benchmarks-IT Benchmarks with a Business Focus. In: Performance Evaluation: Metrics, Models and Benchmarks (2008), pp. 4-8.*

Poess, M.; Floyd, C. (2000): New TPC benchmarks for decision support and web commerce. In: *ACM Sigmod Record, Vol. 29 (2000) No. 4, pp. 64-71.*

SAP (2001): *SAP Memory Management (BC-CST-MM).*

SAP (2010): *SAP Standard Application Benchmarks. In: http://www.sap.com/solutions/benchmark/index.epx, accessed at 12.3.2010.*

Schneider, T. (2005): *SAP - Performanceoptimierung. Galileo Press, Bonn 2005.*