

8-2010

# Scalability and performance of a virtualized SAP system

André Bögelsack

*Technische Universitaet Muenchen, boegelsa@in.tum.de*

Holger Wittges

*Technische Universitaet Muenchen, wittges@in.tum.de*

Helmut Krcmar

*Technische Universitaet Muenchen, krcmar@in.tum.de*

Follow this and additional works at: <http://aisel.aisnet.org/amcis2010>

---

## Recommended Citation

Bögelsack, André; Wittges, Holger; and Krcmar, Helmut, "Scalability and performance of a virtualized SAP system" (2010). *AMCIS 2010 Proceedings*. 13.

<http://aisel.aisnet.org/amcis2010/13>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2010 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# Scalability and performance of a virtualized SAP system

**André Bögelsack**

Technische Universität München  
boegelsa@in.tum.de

**Holger Wittges**

Technische Universität München  
wittges@in.tum.de

**Helmut Krömer**

Technische Universität München  
krcmar@in.tum.de

## ABSTRACT

Enterprise resource planning systems (ERP), like SAP systems, build the backbone of the business processes in today's large enterprises. This is why a weak performance of a SAP system tremendously decreases the performance of the user and thus of the enterprise. Today many SAP hosting providers make use of virtualization techniques, but disregard the impact of such solutions. In this paper we focus on the impact of virtualization solutions on the performance of SAP systems and follow a quantitative approach to ascertain several benchmark results. We make four contributions: 1) On the basis of a quantitative investigation we give a recommendation how to configure a SAP system for heavy workload. The recommendation helps to avoid hardware resource shortage. 2) We show that the average performance of a SAP system increases up to +2% if a container-based virtualization solution is used. 3) We show that the performance of a SAP system is decreased up to -33% if a Xen-based virtualization solution is used. 4) On the basis of the quantitative results we give recommendations for a new sizing process in order to meet the requirements for virtualized SAP systems.

## Keywords

SAP, ERP, performance, scalability, Xen, Sun Solaris zones

## INTRODUCTION

ERP systems build the backbone of large enterprises. They are vital for the execution of daily business processes and therefore critical for the success of the enterprise. Today, many ERP systems are hosted in environments utilizing virtualization technique (SAP, 2010). This has several advantages e.g. reduced administration and hardware costs due to consolidation. But it is common sense, that there is one main disadvantage: performance (Cherkasova et al. 2007). Due the usage of an additional software layer (virtualization solution) there is a performance overhead. A lot of research has been done so far to evaluate the performance overhead with different software products and benchmarks (Barham et al., 2003; Cherkasova et al., 2007; Huang et al., 2006; Mennon et al., 2005; Youseff et al., 2006). But so far we are not aware of any research, which deals with the performance of large ERP systems in virtualized environments. So this paper focuses on the performance and the scalability of an exemplary ERP system in a virtualized environment.

As an instance of an ERP system we choose a SAP system and as an instance for a virtualization solution we choose a Xen-based environment and a container-based solution namely Sun Solaris Zones. In order to evaluate the scalability and the performance of the SAP system we stress the SAP system with an internally running benchmark tool, called the Zachmann test, which is a standard memory test in the SAP environment. During the scalability and performance test we continuously increase the workload and pace several workload situations.

The rest of the paper is organized as follows: the second chapter gives an overview about the complexity of a virtualized SAP system. In the third chapter we describe the experimental design to illustrate how we gained the results. In chapter four we describe the scalability of the SAP system and chapter five describes the performance overheads of the SAP system in different environments. In chapter six we give recommendations how a new sizing process should look like. The last chapter summarizes the paper and provides an outlook.

## LAYERS OF A VIRTUALIZED SAP SYSTEM

In virtualized environments you have to deal with three major performance issues: CPU, memory and I/O overhead (Huang et al., 2006). In this paper we focus on memory operations and demonstrate the complexity by choosing a simple memory access pattern. The SAP system is a typical three-tier-application with an application server and a database server for any

data processing tasks and a presentation layer. In our evaluations we focus on the application server. An application server consists of several processes, which are used for different tasks in the SAP system. The most important process type is the dispatcher+work process. It is used for short time user requests. In order to service larger amount of user request, more than one dispatcher+work process is instantiated and they work closely together. Each dispatcher+work process is a process on the operating system level and requires system resources e.g. shared memory and semaphores. In a virtualized environment one SAP system runs on one operating system, which is located in a virtual machine. All virtual machines do not directly operate on physical but on virtual memory. Access to memory segments is controlled by the virtual machine monitor. In Figure 1 we illustrate the memory accesses in case of a user request for a virtualized and a non-virtualized SAP system. In case of the virtualized SAP system the way for a memory access is as follows: 1) user request is received by one dispatcher+work process. 2) The dispatcher+work process accesses a shared memory segment on the operating system level in order to service the user request. 3) The operating system itself uses dedicated virtual memory from the virtual machine level. 4) Memory on the virtual machine level is assigned by the virtual machine monitor. 5) The VMM has to manage the virtual-to-physical memory mapping.

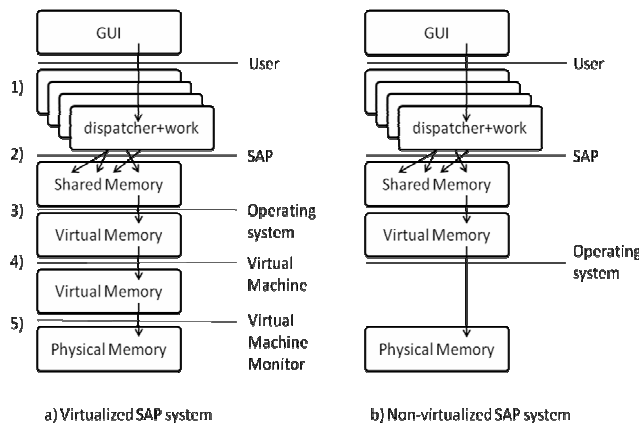


Figure 1: Memory access from SAP to physical memory

Figure 1 shows the way for a memory access in a setup with virtualization and without virtualization. Comparing both configurations shows, that the virtualized SAP system needs to pass through two additional memory access layers: the layer of the virtual machine, and the layer of the virtual machine monitor (VMM). In theory this may cause a performance overhead whenever intensive memory access is necessary.

**EXPERIMENTAL DESIGN**

**Workload situations**

In a typical daily workload scenario the SAP system encounters different workloads. For simplicity we distinguish between three situations: low, medium and high workload. In order to characterize the situations we use the average CPU utilization of the underlying hardware. Figure 2 illustrates each situation during the execution of different workload configurations in our experiment. Each diagram uses the time line as horizontal axis and the utilization (in percent) as the vertical axis. We monitored four utilization values: user time, system time, wait time and idle time. A high user time reflects that the CPU spends time for a user request. The calculation is as follows  $100\% - \text{percent of time spent for user requests} - \text{percent of time waited for I/O requests} - \text{percent of time for system activities} = \text{percent of idle time}$ .

The characterization for each workload situation is as follows: In a low workload situation (Figure 2a) only a small amount of time is spent for user requests (blue line) - there are enough resources available to complete any other requests which may occur. In a medium workload situation the system deals with a lot of user requests already but still has capacity for additional user requests whereas in a high workload situation almost the entire time is spent for user requests and there is no more capacity available.

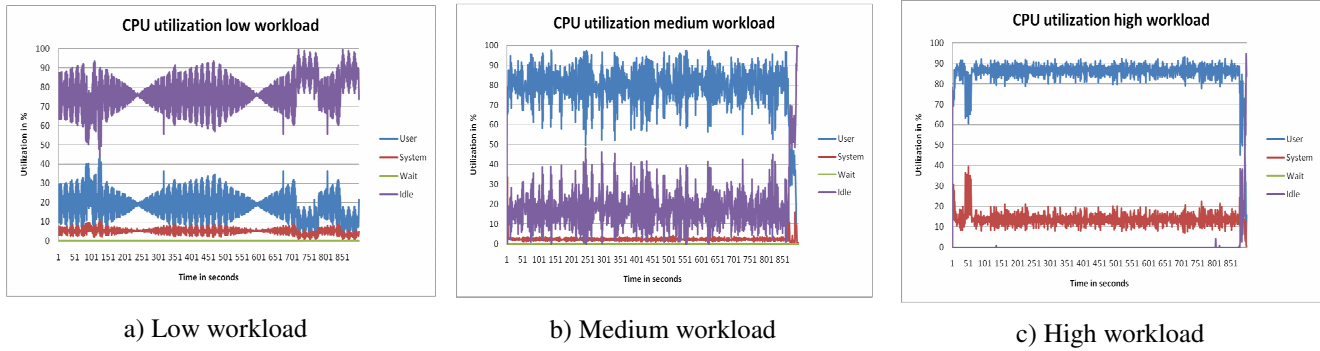


Figure 2: Workload situations

### Zachmantest

In order to reproduce such workload, we use a synthetic benchmark, the Zachmantest (Kühnemund, 2007). It is a well established test in the SAP context and is used for certification of new virtualization solutions. It aims on memory workload generation and runs inside the SAP system. Other third party testing tools might be suitable too, but here the focus lies on the workload generation from inside the SAP system and therefore the Zachmantest suits our needs best. The Zachmantest exploits the specific mechanisms of the dispatcher+work processes: intensive operations in the shared memory segments. The overall workload situation of the SAP system can be controlled easily by the numbers of Zachmantests running in parallel. Adapting the numbers of running tests in parallel we can increase and decrease the workload for the entire virtual machine and the virtual machine monitor.

The Zachmantest uses throughput as the performance metric. The test works as follows: 1) An internal table is created. Internal tables are representatives of database tables, but within the buffers of dispatcher+work processes. 2) The internal table is filled with random data. 3) The test randomly loops over the internal table and reads records. 4) The procedure terminates after 900 seconds. The throughput is the number of randomly accessed records in 900 seconds. The result is reported by each instance of the test. As more than one instance is used, the overall throughput is calculated as the average of each instance. This test is for testing the maximum possible throughput and hence simulates intensive SAP users. Dealing with heavy SAP user is important as such users are likely to slow down the entire SAP system.

### Testing procedure

In our testing procedure we have three workload parameters: 1) the number of parallel running virtual machines (and hence parallel running SAP systems) named  $w_{vm}$ , 2) the number of parallel running Zachmantests per SAP system  $w_z$  and 3) the number of configured dispatcher+work processes per SAP system named  $w_{dw}$ . The workload parameters range is pragmatically limited due to e.g. restrictions of available memory. We limit the number of parallel running virtual machines to 3, the number of parallel running Zachmantest to 16 and the number of dispatcher+work processes to 14. A test run looks like this:

- 1: set  $w_{dw}=1$
- 2: while  $w_{dw} \leq 14$
- 3:     set  $w_z = 1$
- 4:     while  $w_z \leq 16$  do
- 5:         perform Zachmantest
- 6:         wait 360 (for cooling down)
- 7:          $w_z++$
- 8:     endwhile
- 9:  $w_{dw}++$
- 10: endwhile

One entire test run finishes after 3.3 days and is repeated for three times. The overall result is calculated as the average of the three results. The test runs are then repeated for a varied number of virtual machines  $w_{vm}$ .

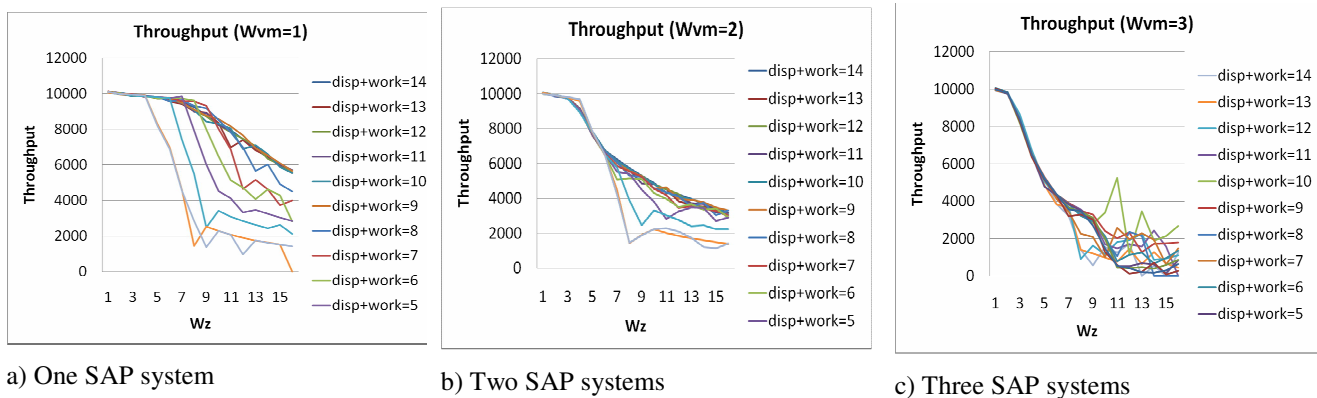
**Testing setups**

We use three different testing setups for the server. The first one is the native environment with Sun Solaris 10 as operating system without any kind of virtualization. The SAP systems are installed in a common way and operate in parallel. The second setup uses Sun’s Container technology (called zones) with Sun Solaris 10 as operating system and up to three zones as virtualization solution. Each zone contains one SAP system. The third setup is a Xen-based environment running a Sun Solaris Nevada as an operating system. The operating system is based on a Sun Solaris kernel with an integrated Xen VMM. All virtual machines are running as paravirtual machines. There was no memory page sharing mechanism enabled, so each virtual machine operates on its own memory segments in the hardware RAM.

The hardware configuration is as follows: the server is a X4200 server from Sun Microsystems with two 2.4GHz AMD Dual core Opteron 280 CPU. It is equipped with 16GB RAM (DDR2-667) and four internal 73GB 10,000RPM SAS disks.

**SCALABILITY OF A SAP SYSTEM**

The ability of the SAP system to concurrently handle multiple user requests is especially important in situations where a large number of users work in parallel. In this section we inspect the ability of the SAP system to utilize the underlying hardware effectively. In order to test the scalability of the SAP system we run several test configurations of the SAP system and increase the workload step by step as described in the previous section. Figure 3a to 3c shows the throughput of several SAP systems in different configurations. Figure 3a shows the throughput of one SAP system with a varying value of  $w_z$  and  $w_{dw}$ . The results for more than one SAP system per  $w_z/w_{dw}$ -configuration are displayed as the average throughput of all SAP systems.



**Figure 3: Scalability of one, two, three SAP systems with different configurations**

Figure 3a shows the results for one SAP system. As can be seen at a certain point the throughput decreases dramatically. For example: using only two dispatcher+work processes the throughput stays relatively stable until  $w_z = 4$  ( $w_{vm}=1$ ). But as soon as  $w_z \geq 5$  the throughput decreases in a significantly way. This behavior can be identified in every configuration. We see a link between 1) the number of dispatcher+work processes and 2) the number of parallel running Zachmannstests  $w_z$ . Interpreting  $w_z$  not as the number of synthetic benchmarks but as a number of power SAP users, we found that there is a ratio for the number of configured dispatcher+work processes and the expected power SAP user to gain the best throughput results.

The ratio should be between 1:1 and 1.5:1. A ratio of 1:1 means you handle 20 power SAP users with 20 dispatcher+work processes; a ratio of 1.5:1 means you handle 30 SAP power users with 20 dispatcher+work processes. If you choose a higher ratio like 2:1, you might face a performance problem if more power SAP users work on the system than expected. If you choose a lower ratio you might waste system resources, because every dispatcher+work process consumes a lot memory.

We determined the ratio also for two and three and in different testing setups. The ratio remains the same. Moreover we found that the ratio does not change when the SAP systems run inside a virtual machine or a zone. This leads us to the assumption that virtualization does not influence the scalability of a SAP system.

### NATIVE VERSUS VIRTUALIZED PERFORMANCE

Because virtualization is state-of-the-art in today’s computing centers, we investigate it’s influence on an ERP system’s performance when running in high workload situations. In our test environment we use two virtualization solutions: container-based Sun Solaris zones and a Xen-based environment. We compared the gained results against the native performance.

#### Performance in native environment versus zones

Zones are described as lightweight, container-based virtualization where only about 1% overhead should be incurred (Sun, 2009). Providing nearly native performance with all the benefits of virtualization, zones are used widely in SAP hosting scenarios which is why we did several performance measurements with zones. Zones are implemented on basis of a labeling technique (Sun, 2006) and so they do not require an additional software layer. But we expect to see a slight performance overhead because of this labeling technique. In the following, whenever a SAP system runs inside a zone, we refer to it as a zoned SAP system.

Figure 4 shows the overall throughput of one SAP system ( $w_{vm}=1$ ) installed in a native environment compared to the zoned SAP system. For each configuration you see the cumulated throughput. For one configuration (e.g.  $w_{dw}=2$ ) we run 16 separate stress test with a varying number of parallel Zachmanntests  $w_z$ . After that we cumulated the results from each stress test for this configuration. For low values of  $w_{dw}=2, 3, 4$  a native SAP system has a higher overall throughput than the zoned SAP system, which is the expected result.

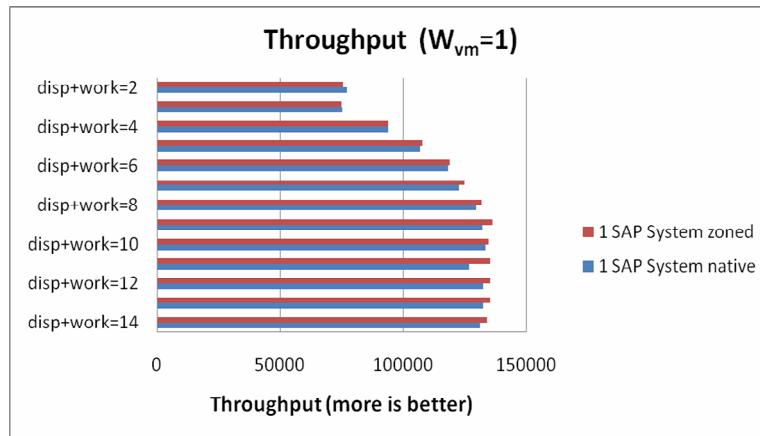


Figure 4: One SAP system zoned compared to native

For higher values of  $w_{dw}$  the result is the opposite. Starting with values of  $w_{dw} \geq 5$ , the zoned SAP system outperforms the native SAP system. As we did not expected to see this behavior, we repeated the tests several time with different configurations but came to the same results; we always gained a positive effect. A possible cause for this can be found in caching mechanisms, but yet we are still in the process of analyzing the results together with the operating system vendor. Table 1 summarizes the results of our tests. As we cannot compare the overall throughput between each configuration we use the individual throughput. Table 1 shows the ‘global’ best, average and worst result from all configurations for native and zoned SAP system. The unit is “records per Zachmanntest”. We calculated the difference between the native and zoned environment in order to demonstrate the impact of the zones.

|              | Native  | Zoned    | Difference |
|--------------|---------|----------|------------|
| Best result  | 10184.7 | 10140.97 | - 0.43 %   |
| Average      | 7278.4  | 7403.53  | + 1.70 %   |
| Worst result | 1169.11 | 1375.93  | + 15.03 %  |

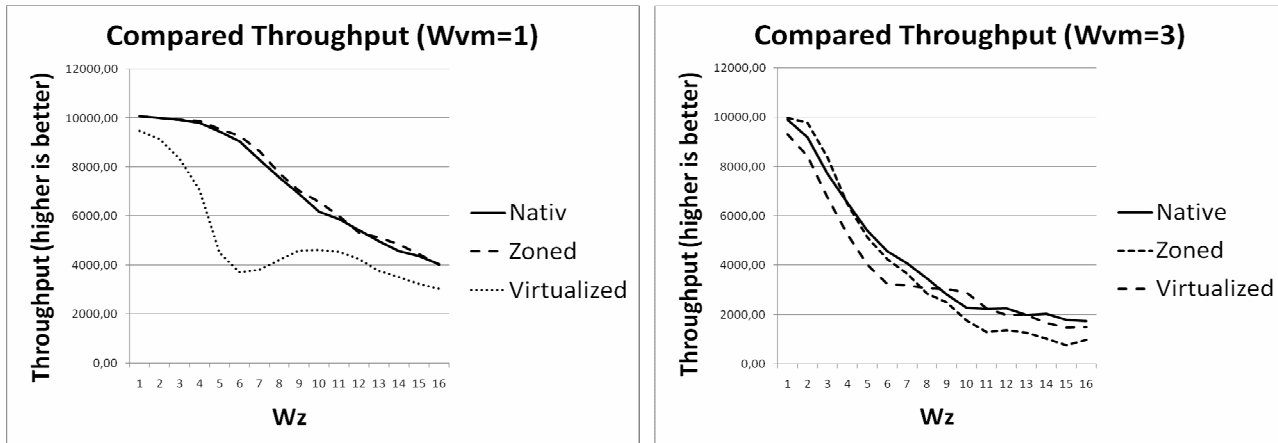
Table 1: Differences in throughput (records per Zachmanntest) between native and zoned performance

Although these differences appear quite small, the highest difference between the native and zoned SAP system is about +15.03%. In a real world SAP hosting center this means, that by using Sun zones the performance of a SAP system can be increased.

**Native versus virtual performance**

As virtual machines (compared to zones) require an additional software layer the virtual machine monitor, it is obvious that there should be a performance overhead. We expect to see an overhead because the additional software layer requires its own CPU resources. The VMM schedules, maintains, and manages the virtual machines on the host. In the following, whenever a SAP system runs inside a virtual machine, we refer to it as a virtualized SAP system.

Figure 5a shows the results of our measurements. We use the individual throughput of one SAP system ( $w_{vm}=1$ ) with different configurations. It runs inside a virtual machine (red line), then in a native environment (blue line) and also inside a zone (green line). For each configuration we iterated over the workload parameter  $w_{dw}$  from 2 to 12.



a) One SAP system virtualized compared to native and zoned

b) Three SAP systems virtualized compared to native and zoned

**Figure 5: Throughput of one and three SAP systems in native and virtualized environment**

The performance of the virtualized SAP system is significantly worse compared to both other configurations. The throughput is always below the results of the two other configurations. Considering our central assumption that virtualization has a negative impact on the performance, these test results can be seen as support. But it encourages us to do some more experiments with more than one virtual machine running in parallel.

Figure 5b summarizes the results of three virtualized and three native SAP systems. The blue line indicates the performance of three native SAP systems and the red line indicates the performance of three virtualized SAP systems. In order to display the results for three SAP systems, we use the average throughput. The results meet our expectations and the virtualized performance of three SAP systems is about 33% below the average throughput of native SAP systems. The highest loss for the virtualized SAP systems is about 44%. Table 2 summarizes the results.

|              | Native   | Virtualized | Difference |
|--------------|----------|-------------|------------|
| Best result  | 10324.48 | 7930.29     | - 23.18 %  |
| Average      | 7516.01  | 5001.48     | - 33.45 %  |
| Worst result | 4746.56  | 2651.72     | - 44.13 %  |

**Table 2: Differences throughput (records per Zachmanntest) between native and virtualized environment for  $w_{vm}=3$**

In average SAP systems loose 33% throughput in case of intensive memory operations. The result seems to be high, but related work shows up to 50% performance loss (Cherkasova, 2007). Our results are in the range of other research works and Table 3 puts our results in context to results from other research so far:

| Source     | Win/Loss    | Testing method                      |
|------------|-------------|-------------------------------------|
| This paper | Up to +1.7% | Zachmanntest with Sun Solaris Zones |

|                |              |  |
|----------------|--------------|--|
| Zhang [10]     | Up to -15%   | Kernel Build                           |
| Barham [1]     | Up to -16.3% | SPEC WEB99                             |
| Huang [3]      | Up to -17%   | NAS Parallel Benchmarks                |
| Youseff [9]    | Up to -30%   | MPI microbenchmark                     |
| Mennon [5]     | Up to -33%   | Httpperf                               |
| This paper     | Up to -33%   | Zachmanntest with Xen virtual machines |
| Cherkasova [2] | Up to -50%   | Httpperf                               |

**Table 3: Comparison of performance results**

Although there seems to be much related work available, to our knowledge there is no research work so far available, which deals with the performance of SAP system either in virtualized, zoned or native environment. Moreover our work does not use a third party tool in order to benchmark a computer system (like the SPEC benchmark tools) but uses a SAP specific program.

### SIZING RECOMMENDATION

Sizing is the process of estimating the necessary hardware for a new system. In order to run a SAP system in a virtualized environment with a sufficient performance, it is necessary to size the underlying hardware in new ways. In its infancy sizing was more an estimate than a calculation. The goal of a sizing process was the estimation of the required hardware resources. In the light of this paper's result existing sizing algorithms should be adapted, especially for the topics CPU and RAM.

The main factor for our sizing recommendation is the expected average number of SAP users. But the paper's results show that especially workload peak situations are critical. Typically such workload peaks occur twice a day: in the morning when everyone logs on to the SAP system and after the lunch break. In such situations the overall performance of the SAP systems decreases significantly and the underlying hardware does not meet the requirements any more. This situation gets worse when virtualization is used and two or more SAP systems on one server run into a workload peak. Facing two workload peaks at the same time makes the situation worse, especially in a virtualized environment. Hence, the sizing should consider if virtualization is used at all or not.

Usually the average number of SAP users is used to estimate the required hardware. We have seen that some SAP power users can lead to a heavy workload situation where no hardware resources are available any more. From our perspective, the number of SAP power users is even as important as the average number of SAP user. When planning the sizing of the SAP system itself, the ratio of expected SAP power user to the number of dispatcher+work processes should be taken into consideration.

The required CPU resources for new virtualized SAP systems should be calculated in traditional way, but with an extra-charge of at least 33% if power users are expected to be seen on the SAP system later on. This extra-charge stems from our measurement results for power users and should be taken into account in order to avoid memory throughput shortage.

### CONCLUSION

Virtualization vendors mention a wide variety of advantages that go along with their solutions: hardware cost savings, lower administrations cost and improved hosting processes. In this paper we look on the other side of the coin and investigated the performance of a SAP system in virtualized environments. We found that the scalability of a SAP system does not change due to the usage of virtualization. Using a quantitative approach, a ratio is determined. It helps to find the appropriate number of dispatcher+work processes for an expected number of power users in order to performance needs. Usually SAP system sizing is done by guessing or preliminary estimates. We offer an empirical determined ratio for the configuration of a SAP system for the first time. SAP system administrators can use this ratio as a rule for system configuration.

The core of this work focuses on the performance difference of a SAP system in a native, zoned and virtualized environment. We used a synthetic benchmark called the Zachmanntest to evaluate the overhead. The zone solution of Sun Microsystems is intended to cause nearly no overhead. We found, that it causes an average performance increase of about +1.70%. Causing an increase is quite unusual and thus we are in the state of analyzing the software architecture of the solution to explain the



cause of the increase. The Xen environment met our expectations. We measured an average decrease in performance of about -33.45%, which is in the range of other research results.

As a result we can give recommendations for a new sizing rule in order to meet the requirements of virtualized SAP systems under intensive workload, which helps to avoid performance degradation.

In this paper we focused on intensive memory operations of a SAP system. Beside memory intensive operations, I/O operations of the underlying database also play a significant role. It is our future research agenda to include I/O operations in our considerations.

## REFERENCES

1. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A. (2003) Xen and the Art of Virtualization, in *SOSP '03 - Proceeding of the nineteenth ACM symposium on Operating systems principles*, 164-177.
2. Cherkasova, L. Gupta, D., Vahdat, A. (2007) Comparison of the Three CPU Schedulers in Xen, in *ACM SIGMETRICS Performance Evaluation Review*, Volume 35, Issue 2, 42-51.
3. Huang, W., Jiuxing, L., Abali, B., Dhabaleswar, K. P. (2006) A Case for High Performance Computing with Virtual Machines, in *ICS '06: Proceedings of the 20th annual international conference on Supercomputing*, 125-134.
4. Kühnemund, H. (2007) Documentation for SLCS v2.3. SAP AG, Walldorf
5. Mennon, A., Santos, J. R., turner, Y., Janakiraman, G. J., Zwaenepoel, W (2005) Diagnosing performance overheads in the Xen virtual machine environment, in *Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments*, 13-23.
6. SAP AG (2010) <http://www.sap.com/solutions/benchmark/sd2tier.epx>, accessed on 1/19/2010
7. Sun Microsystems (2006) Virtualization and Namespace Isolation in the Solaris Operating System (PSARC/2002/174)
8. Sun Microsystems (2009) Systemverwaltungshandbuch: Solaris Container - Ressourcenverwaltung und Solaris Zones
9. Youseff L., Wolski R., Gorda B., Krintz C. (2006) Evaluating the Performance Impact of Xen on MPI and Process Execution For HPC Systems, in *Proceedings of Second International Workshop on Virtualization Technology in Distributed Computing*
10. Zhang, X., Dong, Y. (2008) Optimizing Xen VMM Based on Intel® Virtualization Technology, in *Proceedings of the 2008 International Conference on Internet Computing in Science and Engineering*, 267-274.