Enterprise Applications in the Cloud:

Virtualized Deployment

Leonid Grinshpan, Oracle Corporation (www.oracle.com)

Subject

Enterprise applications (EA) can be deployed in the Cloud in two ways:

1. **Non-virtualized setup** hosts on the same physical servers different EAs without logical borders between them (no partitions, virtual machines or similar technologies in place).
2. **Virtualized arrangement** separates EAs from each other by employing the above-mentioned techniques.

In a recent article [*Leonid Grinshpan. Enterprise Applications in the Cloud: Non-virtualized Deployment; Measure IT, issue 10, 2011*] we analyzed interference among EAs when they are set up on physical servers without logical boundaries. We also outlined how to collect data on resource utilization by hosted EAs.

This current article studies the impact of hardware virtualization on EA performance. We use queuing models of EAs as scientific instruments for our research; methodological foundation for EA performance analysis based on queuing models can be found in the author's book [*Leonid Grinshpan. Solving Enterprise Applications Performance Puzzles: Queuing Models to the Rescue, Willey-IEEE Press, 2012, http://www.amazon.com/Solving-Enterprise-Applications-Performance-Puzzles/dp/1118061578/ref=ntt_at_ep_dpt_1*].

We begin by examining EA performance on a non-virtual platform that includes Web, Application, and Database servers. In the next step, we substitute the Database server with three virtual machines (VM) and compare EA performance on virtualized versus

non-virtualized platforms. Next, we evaluate how many CPUs should be assigned to VMs to ensure acceptable transaction response times. In the last step, we discuss why virtualization does not prevent applications from impacting each other's performance.

Model 1 Application Performance on Physical Platform

For analysis we are using the queuing model from the article [*Leonid Grinshpan. Enterprise Applications in the Cloud. Non-virtualized Deployment; Measure IT, issue 10, 2011*] with slightly different workload and transaction profiles. Model 1 (Figure 1) represents a simplified three-tiered Cloud with Web, Application, and Database servers. This cloud hosts three EAs (App A, App B, App C) serving three user groups, one group per EA. Each server corresponds to the model's node with a number of processing units equal to the number of CPUs in the server. The users of each EA as well as network are modeled by dedicated nodes. All servers are the physical ones without any partitioning among applications. Web and Application servers have 8 CPUs each; Database server has 16 CPUs.

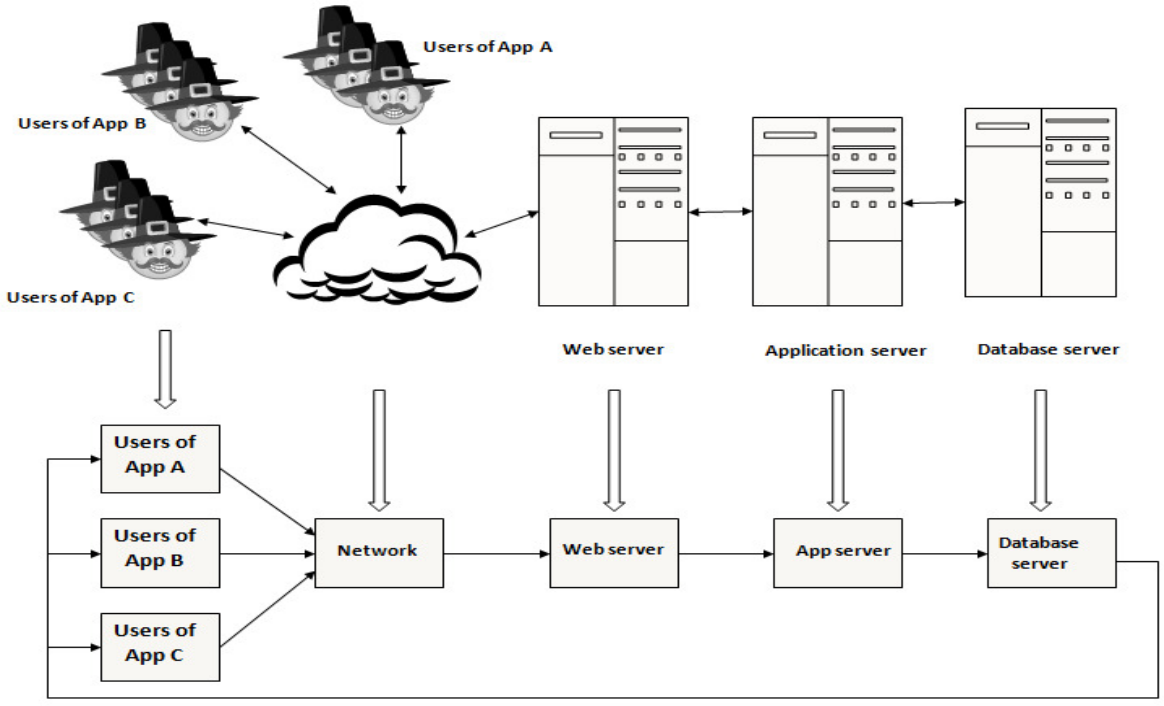

Figure 1 Model 1 of the Cloud hosting three enterprise applications

The models in this article were analyzed using TeamQuest solver [*http://teamquest.com/products/model/index.htm*]. Workload 1 for Model 1 is characterized in Table 1. For each application it is presented by a transaction identified by application name. For each transaction, each user initiates it a number of times indicated in column *"Number of transaction executions per user per hour."* We analyze the model for 100, 200, 300, 400, 500, 600, 700, and 800 users.

Table 1

Workload 1 for Model 1

Transaction name	**Number of users**									**Number of transaction executions per user per hour**
	Total 3	**Total 100**	**Total 200**	**Total 300**	**Total 400**	**Total 500**	**Total 600**	**Total 700**	**Total 800**	
App A transaction	1	50	100	150	200	300	400	350	400	10
App B transaction	1	25	50	75	100	150	200	175	200	20
App C transaction	1	25	50	75	100	150	200	175	200	5

To solve the model we have to specify the profile of each transaction (Table 2). Transaction profile is a set of time intervals (service demands) a transaction has spent in all processing units it has visited while served by the application.

Table 2

Transaction Profiles (seconds)

	Time in Network node	Time in Web server node	Time in App server node	Time in Database server node
App A transaction	0.001	0.4	2.0	10.0
App B transaction	0.0015	0.2	1.0	5.0
App C transaction	0.003	0.4	10.0	10.0

Transaction times for non-virtualized physical platform are presented in Figure 2.

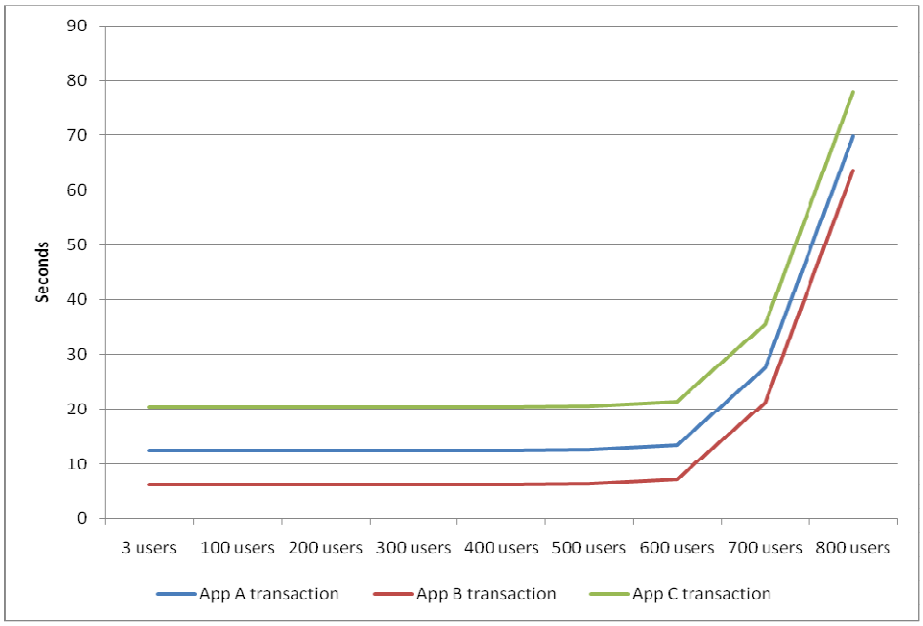

Figure 2 Transaction response times for three applications

Model 1 predicts that the system can serve 600 users with acceptable transaction times. An increase in a number of users beyond 600 causes steep transaction time growth because the Database server reaches the maximum of its CPU capacity (Figure 3).

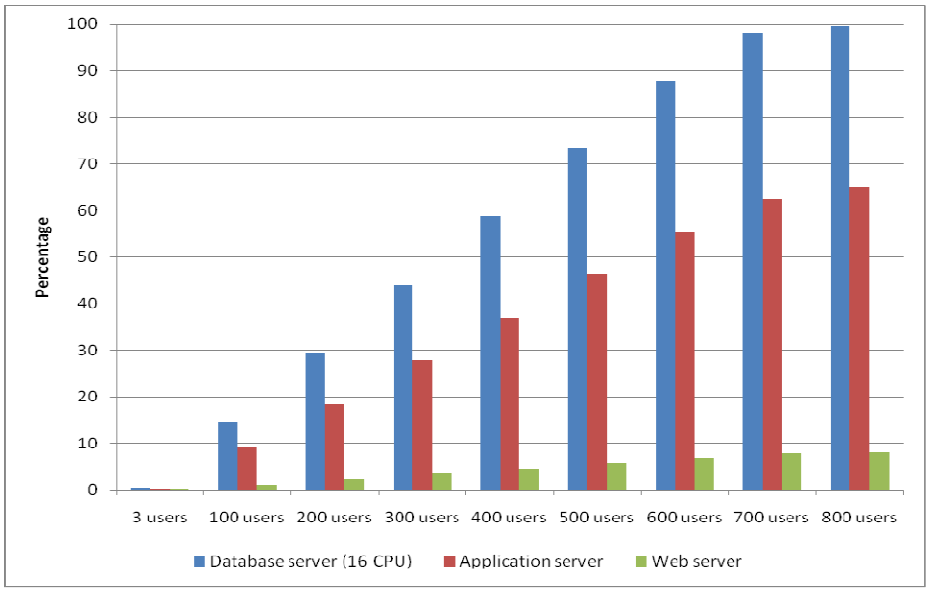

Figure 3 Utilization of Cloud's servers

Figure 4 shows breakdown (by percentage) of Database server utilization by different EAs.

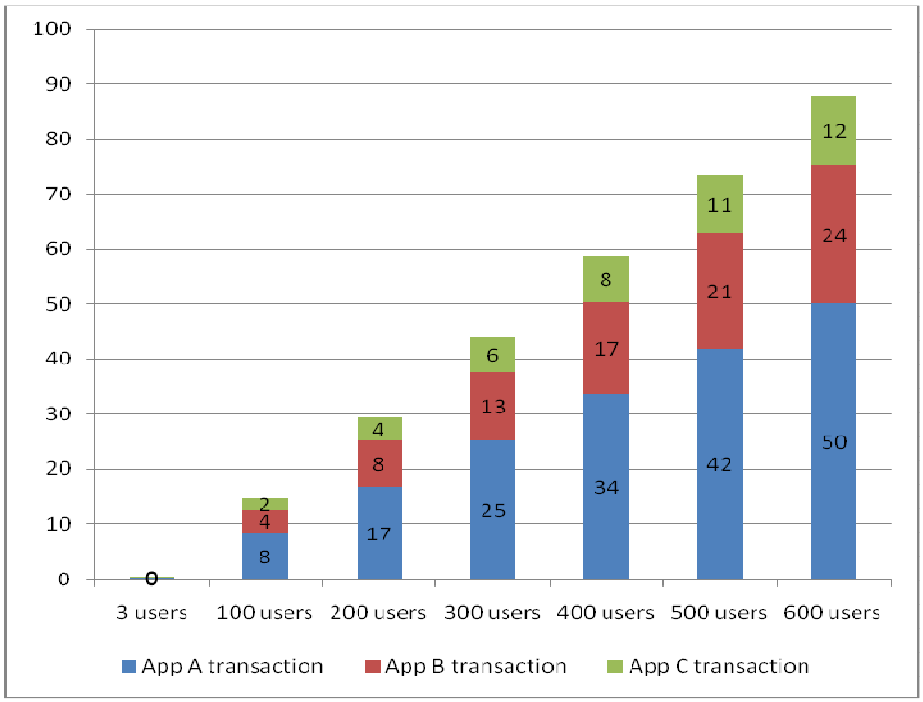

Figure 4 Breakdown of utilization of Database server by applications A, B, and C

Figure 4 suggests that the largest "consumer" of Database server CPU capacity is App A. We use this breakdown data on the next step where we divide Database server by three VMs (one VM per application) and study how Database server virtualization impacts performance.

Model 2 Application Performance on Virtual Platform

The virtualized Database server hosts three VMs: VM A with App A, VM B with App B, and VM C with App C. On the Database server with 16 CPUs, one CPU accounts for 6.25% of the server's total capacity (100% / 16 = 6.25%). A chart on Figure 4 shows the percentage of Database server utilization by each application for 600 users:

App A: 50%
App B: 24%
App C: 12%

Based on the above data, listed below are the numbers of CPUs we have to assign to each VM:

VM A: 50% / 6.25% = 8 CPUs

VM B: 24% / 6.25% = 4 CPUs

VM C: 12% / 6.25% = 2 CPU

Model 2 of a system with three VMs hosted on Database server is pictured in Figure 5.

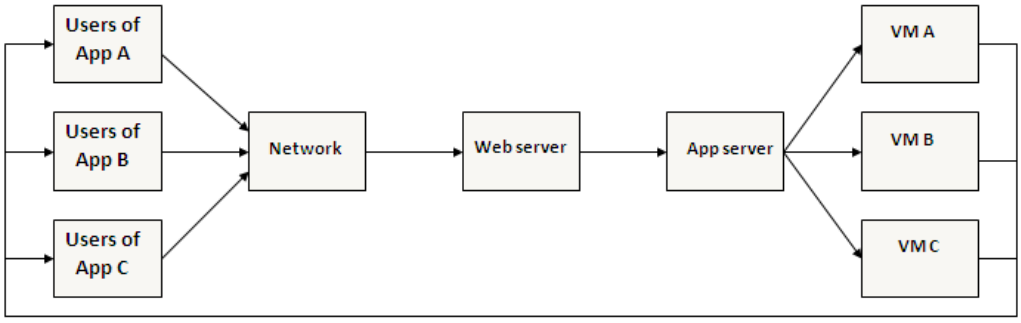

Figure 5 Model 2 with virtual machines

Transaction times delivered by Cloud with virtualized Database server are presented in Figure 6.

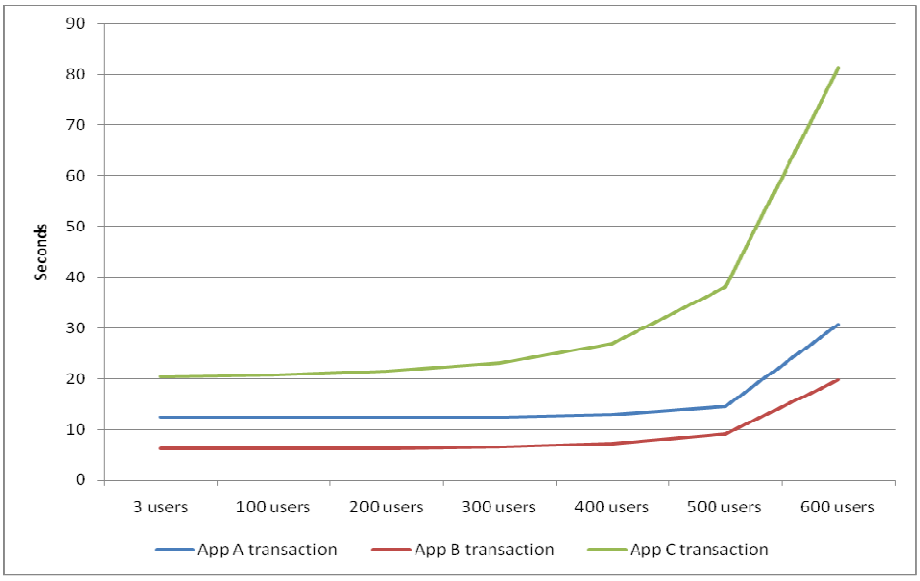

Figure 6 Transaction times in system with VMs having 8-4-2 CPUs

Model 2 predicts transaction time degradation for 600 users for all three EAs due to high utilization of all three VMs (utilization exceeds 90%, Figure 7).

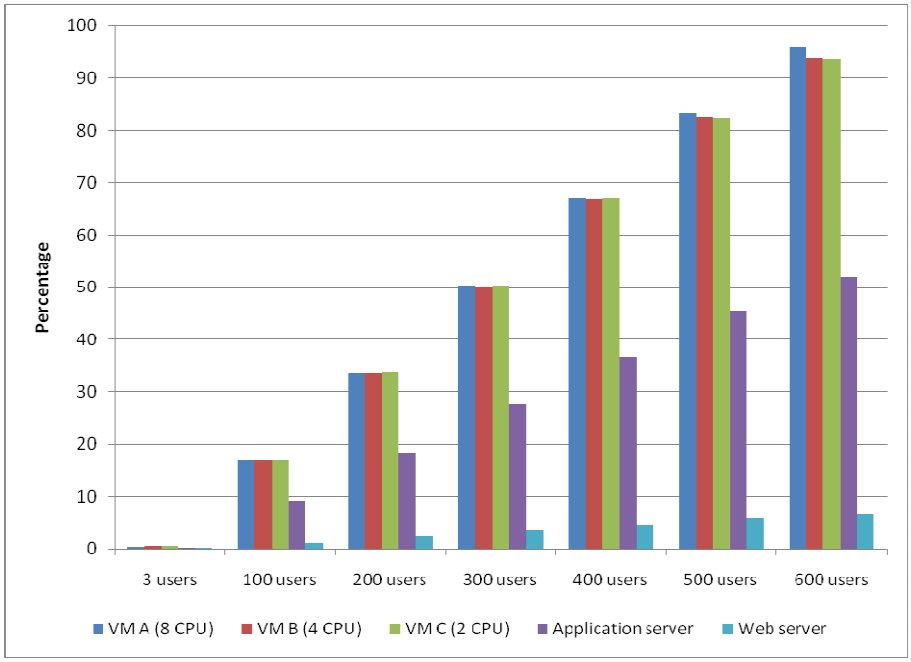

Figure 7 Utilization of servers in system with VMs having 8-4-2 CPUs

Model 3 Allocating additional CPUs

So far we assigned 8+4+2=14 CPUs out of 16 CPUs on the physical Database server to three VMs. Because the largest transaction time increase happened for App C and App A, we will distribute the remaining 2 CPUs as shown below:

VM A: 50% / 6.25% = 9 CPUs
VM B: 24% / 6.25% = 4 CPUs
VM C: 12% / 6.25% = 3 CPUs

Model 3 predicts noticeable improvement of transaction times for App C and App A (Figure 8). As expected, adding CPUs to VMs C and A did not impact time for App B.

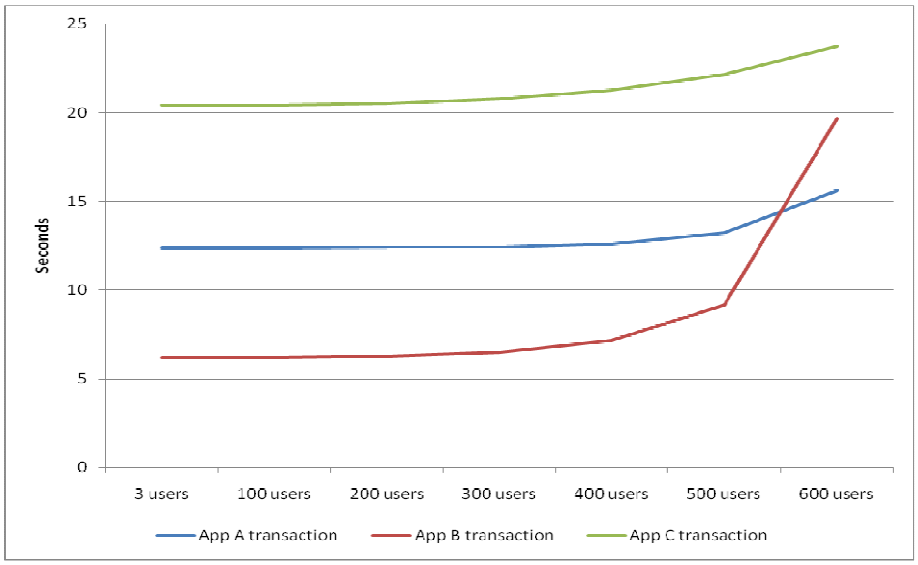

Figure 8 Transaction times in system with VMs having 9-4-3 CPUs

Time improvement for App C and App A was achieved by lowering CPU utilization of VMs C and A (Figure 9).

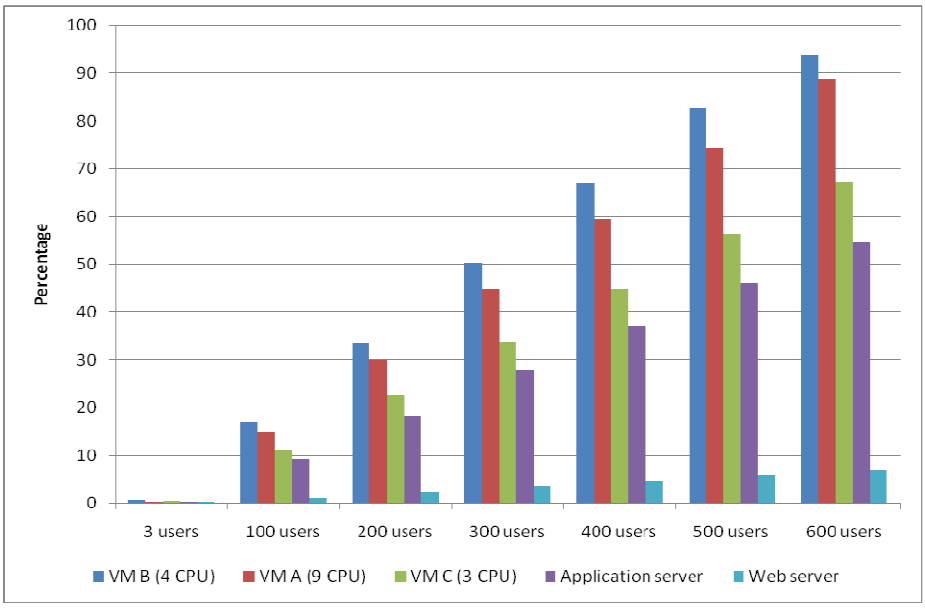

Figure 9 Utilization of servers in system with VMs having 9-4-3 CPUs

Comparison of non-virtualized and best performing virtual deployment with VMs having 9-4-3 CPUs is seen in Figure 10.

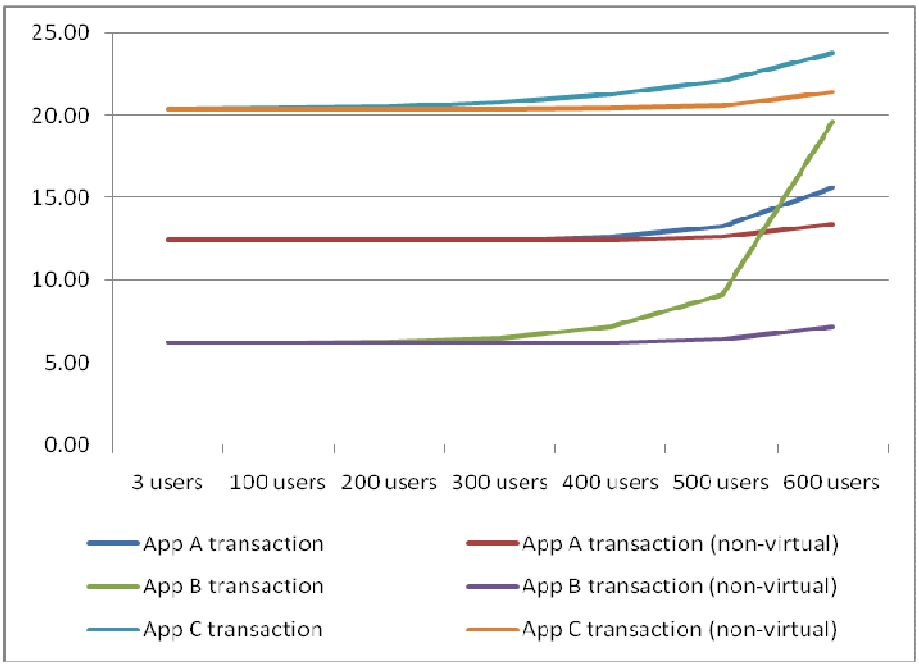

Figure 10 Transaction times on virtual and non-virtual platforms

Virtualized setup features longer transaction times than physical platform for all EAs starting from 300 to 400 users. That means the capacity of all VMs (having in total the same 16 CPUs as physical Database server) is lower than the capacity of the non-virtualized Database server. This surprising conclusion is in line with queuing theory analysis explaining longer transaction times in virtual systems by longer waiting times in queues.

Quantitative consideration of that effect can be found in [*Leonid Grinshpan. Solving Enterprise Application Performance Puzzles: Queuing Models to the Rescue, Willey-IEEE Press, 2012*]. Here we offer qualitative explanation that is based on the queues behavior in two real life environments – toll plaza and movie theater box office. A toll plaza with booths equally accessible by any cars has lower congestion than the same plaza with booths divided into two categories: ones serving only sedans and others serving only trucks. An intuitive justification is: in non-divided plaza, in the absence of the trucks a sedan can be processed by any booths and vice versa; in divided plaza the booths dedicated to trucks will stay idle even if there is a queue of sedans. The same behavior is exhibited by the queues in a box office – if any wicket serves any theatergoers, than

the waiting queue is not as long as in a case when some wickets provide only to particular customer categories.

Virtualization is not a Chinese Wall

Virtualization technology isolates EAs by creating an illusion that each EA is hosted on its own servers. But despite living in segregated environments, the applications will impact each other's performance when the physical capacity of a hosting platform is overprovisioned.

We show in this paragraph how system monotoring can help to identify overprovisioning of physical AIX platform with logical partitioning (LPAR) [http://www.redbooks.ibm.com/abstracts/tips0119.html?Open]. LPAR is comparable to a virtual machine in the VMware world and represents a subset of a physical server hardware resources. Because of these similarities, the logic of our analysis is applicable not only to AIX LPAR, but to other virtual platforms as well. To monitor a logical partition one can use ***vmstat*** command:

vmstat t N,

where ***t*** is a sampling time interval (seconds) and ***N*** is a number of sampling intervals. The command reports system configuration information as well as performance counters (Figure 11):

```
bash-3.00$ vmstat 2 10000

System configuration: lcpu=24 mem=31743MB ent=4.00

kthr    memory              page              faults           cpu
----- ----------- ------------------------ ------------ -----------------------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa   pc    ec
 1  0 1575647 2655175   0   0   0    0   0   0    3  258 268  0  0 99  0  0.01   0.3
 1  0 1575646 2655176   0   0   0    0   0   0   10  264 278  0  0 99  0  0.02   0.4
 1  0 1575644 2655178   0   0   0    0   0   0   20  448 293  0  0 99  0  0.02   0.5
 1  0 1575644 2655178   0   0   0    0   0   0    3  159 273  0  0 99  0  0.01   0.3
 0  0 1575642 2655180   0   0   0    0   0   0    3  329 267  0  0 99  0  0.01   0.4
```

Figure 11 System configuration and performance counters

In this example physical AIX server has 24 logical CPUs (parameter *lcpu*) and 31743 MB memory (parameter *mem*). The command was executed for a logical partition that is entitled to 4 CPUs (parameter *ent*). Entitlement value equals a number of physical CPUs available to a partition at any time. The total number of entitled CPUs for all partitions

cannot exceed the number of physical CPUs in a partitioned server. If the partition at any moment does not use all its entitlement, then unused capacity is transferred to the common shared pool of CPUs and becomes available to any other partition.
CPU activity is characterised by six parameters:

us - user time percentage
sy - system time percentage
id - idle time percentage
wa - time when system processes I/O requests (percentage)
pc - number of physical CPUs consumed
ec - percentage of entitled CPU consumed (these parameters can exceed 100%).

Figure 12 shows CPU utilization in LPAR when a hosted application consumed all four entitled CPUs. In such a case the sum of the parameters *us*, *sy*, *id* and *wa* is 100%; parameter *pc* informs that application is using almost 5 CPUs and parameter *ec* reports actually the same fact but only as a percentage of an entitlement. The sum of *pc* parameter readings across all LPARs is equal to the number of physical CPUs consumed by all applications. When this sum is equal to the number of physical CPUs, a host server is overprovisioned, affecting the performance of all hosted applications.

kthr		memory		page						faults			cpu					
r	b	avm	fre	re	pi	po	fr	sr	cy	in	sy	cs	us	sy	id	wa	pc	ec
5	0	6969496	1559003	0	0	0	0	0	0	176	28597	10831	90	3	6	0	4.77	119.3
5	0	6969493	1559006	0	0	0	0	0	0	222	26971	10871	91	3	6	0	4.78	119.6
5	0	6969497	1559002	0	0	0	0	0	0	175	27439	10874	90	3	6	0	4.92	123.0
5	0	6969471	1559028	0	0	0	0	0	0	150	26865	10647	90	3	6	0	4.80	119.9
5	0	6969473	1559022	0	0	0	0	0	0	208	26529	10967	91	3	6	0	4.57	114.3
5	0	6969471	1559024	0	0	0	0	0	0	115	26715	10766	91	3	6	0	4.90	122.4
5	0	6969468	1559027	0	0	0	0	0	0	193	27969	10699	90	3	7	0	4.85	121.3
5	0	6969506	1558970	0	0	0	0	0	0	163	34874	10656	90	4	6	0	4.89	122.2
5	0	6969542	1558934	0	0	0	0	0	0	345	34111	10703	89	4	7	0	4.86	121.5
5	0	6969544	1558891	0	0	0	0	0	0	314	33831	10569	89	4	7	0	4.82	120.4
6	0	6969540	1558895	0	0	0	0	0	0	260	33373	10810	89	4	7	0	4.86	121.4
5	0	6969555	1558879	0	0	0	0	0	0	219	33006	10187	89	4	7	0	4.77	119.2
5	0	6969652	1558781	0	0	0	0	0	0	209	34173	10709	89	4	7	0	4.89	122.3
5	0	6969568	1558860	0	0	0	0	0	0	294	33712	10424	89	4	7	0	4.89	122.2
5	0	6969569	1558859	0	0	0	0	0	0	152	32684	10267	90	4	7	0	4.86	121.4
6	0	6969558	1558869	0	0	0	0	0	0	217	33630	10960	89	4	7	0	4.83	120.7
4	0	6969552	1558875	0	0	0	0	0	0	209	33538	10879	89	4	7	0	4.88	122.0
5	1	6969551	1558873	0	0	0	0	0	0	205	33401	10662	89	4	7	0	4.90	122.5

Figure 12 LPAR monitoring data when application demand exceeded entitled number of CPUs

But what if the host is not overprovisioned, but a *pc* parameter for a particular LPAR exceeds its entitlements? Does this mean a performance degradation of an application in that LPAR? The answer is no; it just means that available CPUs are assigned to this LPAR. To avoid misunderstanding follow the rule: Even if the counter's readings look alarming, there are no EA performance degradations as long as transaction times are acceptable to business users.

Take aways from this article:

1. Queuing theory states that processing time in partitioned systems is longer than in non-partitioned ones. Hardware virtualization conforms to that rule and that is why we have to come to terms with the inconvenient truth that the total capacity of all guest virtual machines is lower than the total capacity of non-virtualized host servers for the same workloads and service demands.
2. Virtual environments inherently feature additional management overhead as well as competition for shared components (memory bus, I/O system controller, etc.) that slows down access to shared system resources.
3. Cloud service providers have to take into account a decrease in physical environment capacity, and in availability of their shared components while provisioning virtual environments for EAs.
4. Partitioning does not erect the Chinese wall among hosted applications - they collide when the total demand for resources exceeds available physical capacity.
5. Interpretation of performance counters in virtual environments is more sophisticated than it is in dedicated ones. To avoid misunderstanding follow the rule: Even if the counter's readings look alarming, there are no EA performance degradations as long as transaction times are acceptable to business users.

About the author

During his last fifteen years as an Oracle consultant, the author has engaged in hands-on performance tuning and sizing of enterprise applications for various corporations (Dell, Citibank, Verizon, Clorox, Bank of America, AT&T, Best Buy, Aetna, Halliburton, Pfizer, Astra Zeneca, Starbucks, etc.).