A report on the performance benefits of upgrading memory on servers running VMware ESX Server 4.0

Testing conducted and report compiled by

Binary Testing Ltd Newhaven Enterprise Centre
Denton Island Newhaven East Sussex BN9 9BA

t +44 (0)1273 615270 e info@binarytesting.com

Test Scenario

This report was commissioned by Kingston Technology to look at the performance benefits of upgrading from 2GB to 32GB of physical memory on servers running VMware ESX Server and presenting multiple virtual machines (VMs) all configured to function as web and database servers.

A suite of tests was run to demonstrate the benefits of memory upgrades for the following scenarios:

1) The amount of web traffic multiple VMs can handle before performance degrades. The tests were conducted using a Spirent TestCenter appliance configured to run multiple Avalanches. These generate user requests for web content from the VMs and measure the response times.

2) The performance of multiple VMs running database servers. The Spirent TestCenter was also used for this test where simulated user requests accessed a backend SQL Server database running on the same VM as the web server.

As requested by Kingston Technology, the test suites were run for each scenario where the server under test was fitted with 2GB of DDR-2 memory and then upgraded to 32GB comprising 4 x 8GB DDR-2 modules.

TEST SERVER HARDWARE SPECIFICATION	
Chassis: Supermicro 1U Rack Mount	CPU: 2 x 2.5GHz Intel Xeon L5420 quad-core
Memory: 667MHz DDR2 (2GB and 32GB)	Storage: 2 x 1TB WD drives in mirrored array
RAID: Intel ESB2 3Gbps SATA	Network: Intel 10GBaseSX 10GbE PCI-e

The server was configured with VMware ESX Server v4.0 and each VM configured with Microsoft Windows Server 2008 Enterprise x64 running IIS 7.0. For the backend database tests each VM was also configured with Microsoft SQL Server 2008 x64.

To avoid any network bottlenecks the server under test was connected via a fibre-optic 10-Gigabit uplink to a Netgear GSM7328S Ethernet switch. The Spirent TestCenter had all eight ports connected to the switch's copper Gigabit ports.

TESTED with SPIRENT

VMWare's ESX Server allows multiple virtual servers to run on one physical server. Although each virtual machine (VM) can be configured more or less as the user requires, there are some things that can be done to tune ESX's performance to obtain better results. Although tuning ESX Server for optimum performance is outside the remit of these tests it was possible to do some simple tuning to make better use of resources.

Ideally, the host server will have enough main memory to meet the total memory requirements of all the virtual machines as well as its own memory overhead. For 10 VMs each configured to use 2GB of memory the host system would need more than 20GB of RAM.

However since most systems do not use all the memory available to them, this would mean that the host system's RAM would be under used. When a real system is running a particular application its working set of memory is generally less than the memory available.

A VM therefore only needs to be configured with enough memory for this working set, which in turn reduces the amount of host memory needed to run it. We took advantage of this fact when we set up the VMs and configured each one to use 1GB of memory. We also installed the VMWare Tools so that the systems could take advantage of ESX Server's own optimising functions.

ESX Server has a number of techniques to make better use of host memory. The first technique is page sharing, which means that when a VM starts to load a piece of code or data into its memory ESX Server checks to see if it already has a copy of that page in memory from another VM.

If it does have a copy then instead of allocating real memory to that page it simply shares the one it already has. Where several VMs are running copies of the same operating system, this saving can be substantial.

The second technique is called "ballooning". Ballooning can only take place if the VMWare Tools are installed on the VM. It allows the guest operating system to communicate with the host in such a way that ESX Server is aware of the memory pages the guest operating system has freed up.

ESX Server can then release the real memory that it has provided for these pages and reallocate them as needed. Without this technique memory remains allocated to a virtual machine even though it is not being used.

The final technique is to swap pages to disk. ESX Server uses this when the host memory is over-committed. Although some swapping may occur when starting and stopping virtual machines, swapping while systems are running may indicate a need for more host memory. Extensive swapping will also have an impact on overall performance, which will eventually be reflected in poorer response times.

Determining when performance has degraded to unacceptable levels is to some extent a subjective process, but we chose some key indicators so that we could define when this point had been reached for these tests.

Indicators include high rates of transaction failure for whatever reason (transport errors, time outs, user aborts), high levels of swapping in ESX Server, excessive response times, failure to achieve the specified workload.

When we tested the web server application we configured the Spirent TestCenter with one virtual network for each virtual machine. This allowed us to segregate network activity and direct traffic to specific servers and ensured that we spread the load evenly across all the participating virtual machines.

We first ran some tests to determine how many concurrent users a single VM could handle before performance began to degrade. Performance degradation was indicated by transactions timing out and increasing response times.

We also monitored ESX Server's performance figures to ensure that the system did not need to swap memory to disk. With nearly 2GB of RAM available and a VM that needed only 1GB we did not expect to see any swapping occurring and this turned out to be the case.

We used this figure to plan loads on the tests with multiple virtual machines. We chose concurrent users rather than the more usual transactions per second because each user followed a set script that accessed various pages on the web site.

Each simulated user would spend time on each page before retrieving the next one, and a simulated user would abort the transaction if they received no response within thirty seconds. This activity stressed the server by tying up server memory for buffers and session information, which in turn increased demand on the host memory.

For example, when running three virtual machines with the host configured with 2GB of memory we found that swapping took place as soon as we activated the other two VMs. When we applied the same test load to all three systems the host memory became over committed and HTTP transactions began to fail.

In theory three VMs should have been able to handle three times the workload that a single VM could cope with, but in practice this proved to be only partly the case. Although the system handled the load imposed on it, the response times increased and some transactions failed to complete. Average response times increased from 103 ms to 1098 ms.

When we carried out the database test we used the same simulated user technique as before, but in this case the web pages were used to trigger various database operations such as retrieving and updating database rows and executing complex SQL queries. When the requested operation was completed the user received an acknowledgement page, and the elapsed time indicated how long the operation had taken.

These operations stressed memory in various ways, and the time taken to complete an operation increased as the demand on the memory increased. Because database servers tend to need more memory we would normally have reconfigured the virtual machines with a larger memory size for these tests.

We retained the same 1GB size that we used for the web site tests, partly so that we could have like for like comparisons between the two sets of tests, but mainly because we would have over-committed the server memory and caused swapping to occur almost immediately. This would have distorted the final results.

Since database operations are generally more memory-intensive than web sites this imposed a further strain on the system and it was no surprise to find that the system could not handle the numbers of simulated users that the web site tests could support. Once again, we determined the level of performance that could be obtained with a single VM in a similar way to that used for the web server test.

ESX Server performance figures indicate how much stress the system memory is under. A small sharing value relative to the number of VMs in use indicates a less efficient use of system memory, while a larger value indicates a more efficient use. Similarly, although the balloon figure ought ideally to be zero, a relatively large figure also indicates an efficient use of memory.

By contrast a low swap figure is much better than a large one since it reflects memory over-commitment caused by insufficient system memory available for the number of VMs in use. Shared and balloon memory is allocated in system memory while swapped memory is stored on disk and this has an impact on performance since memory access is much faster than disk.

Spirent performance figures are straightforward. Average response times indicate the kind of stress the memory is under. Long response times indicate that the system is busy while short response times indicate that the system is coping easily. Factors affecting response times include memory commitment and processor use, although some processor load is due to ESX using processor time to manipulate memory.

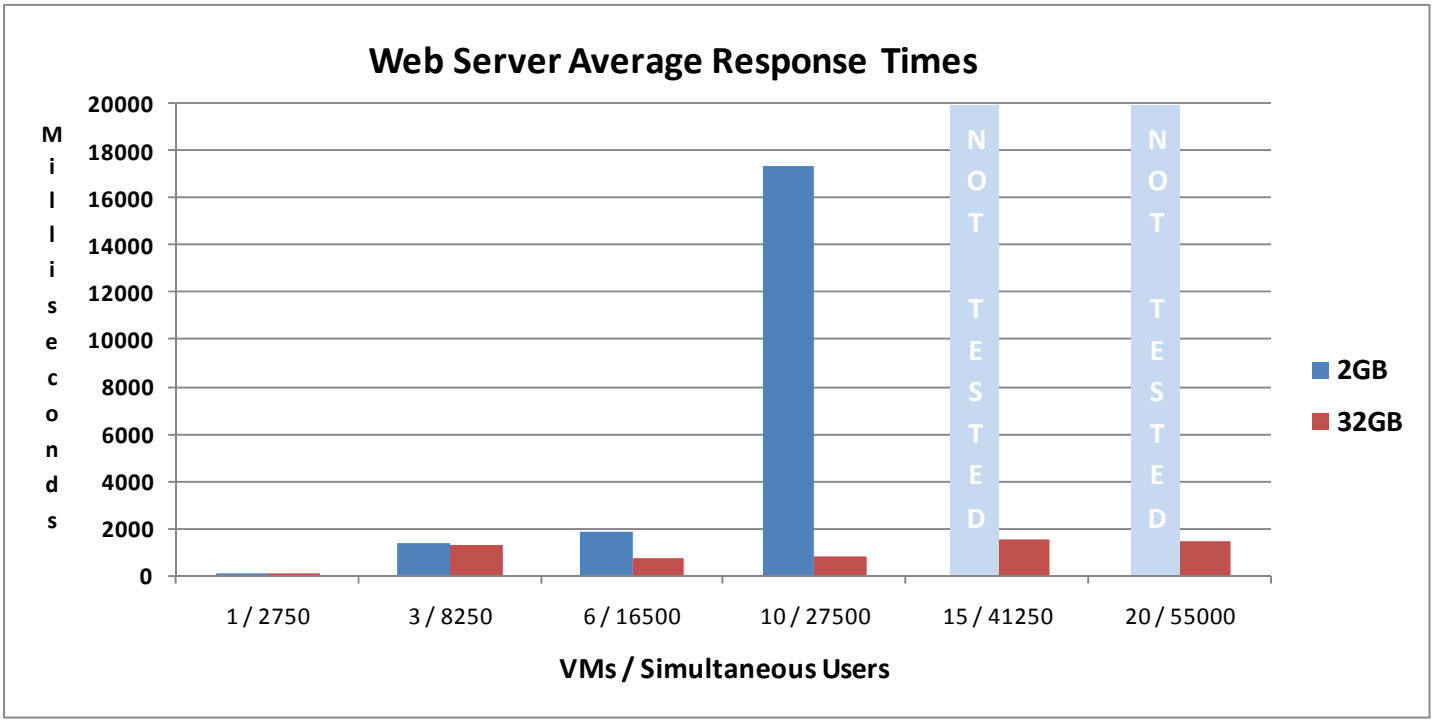
Failed transactions are an indication of a system becoming overloaded, and while small percentage figures are nothing to be concerned about, high figures indicate a system coming under stress. A transaction consists of a request and an acknowledgement that it has been received.

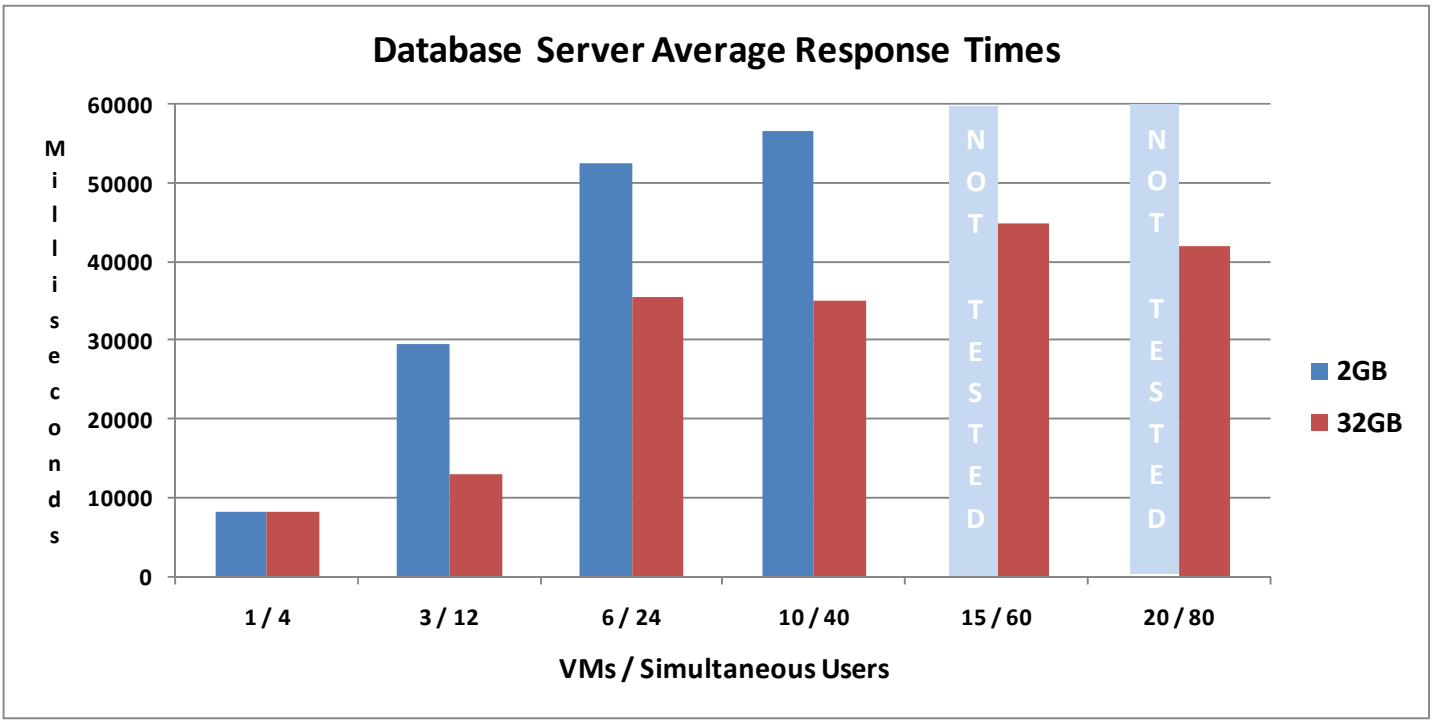
If the request is for some kind of data such as a web page then the data will be sent later as a separate transaction. Transactions may fail for a number of reasons but the effect is the same in that a request is sent to a server and no acknowledgement is received within the expected time limit.

Server Response Times

The test results show clearly the impact the amount of physical memory can have on the performance of VMware VMs running web and backend database servers. With the server fitted with 2GB of memory it was not possible to run tests with more than ten VMs as we were unable to load any more.

Web server responses to client requests across all combinations of VMs and simultaneous users were handled in a timely manner with 32GB of memory but response times degraded significantly when accessing the backend database.

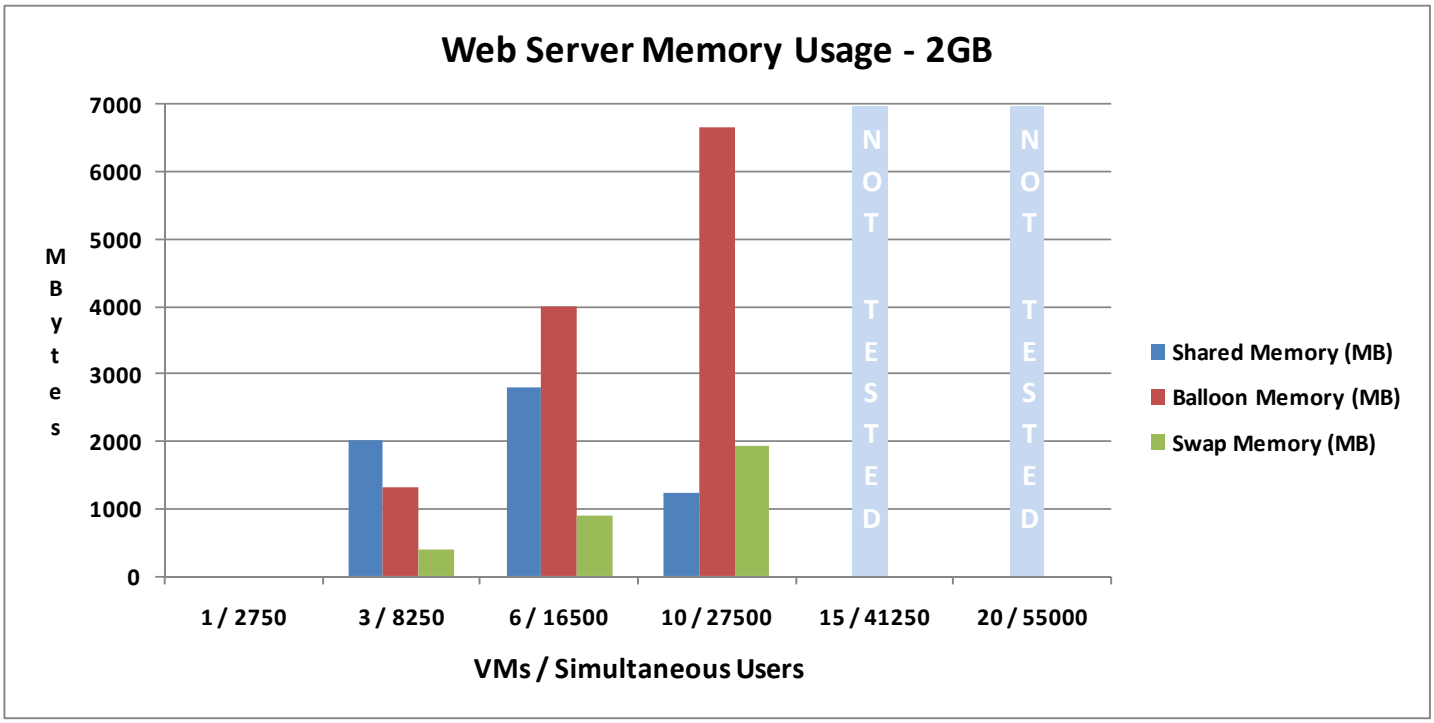


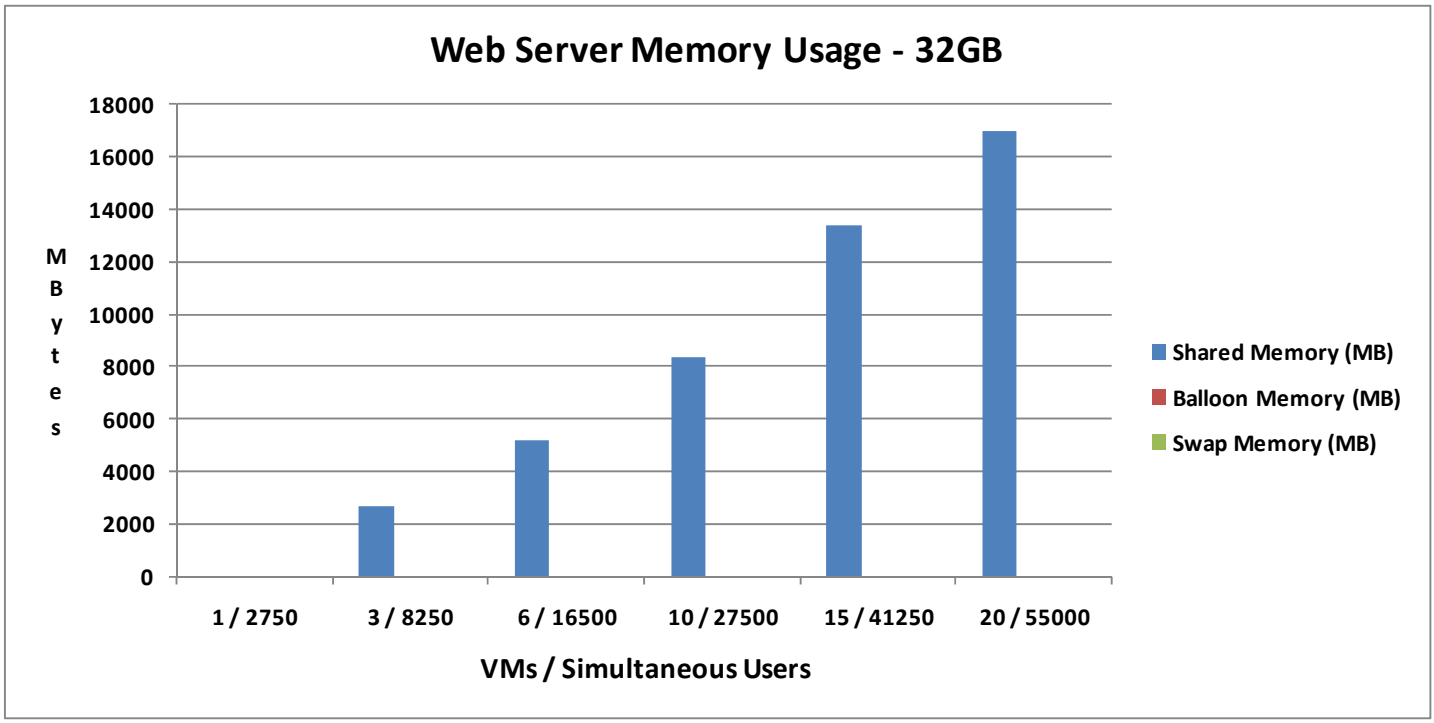


Web Server Memory Usage

For the tests with 2GB, it is clear that VMware's memory management had its work cut out. As the VM/user loads increased, ballooning became more prevalent as did page swapping to disk. This explains the rapidly deteriorating performance of the web and database servers when memory became overcommitted.

During all tests with 32GB of memory no ballooning or page swapping occurred and the graphs show a consistent rise in shared memory as the load was increased.

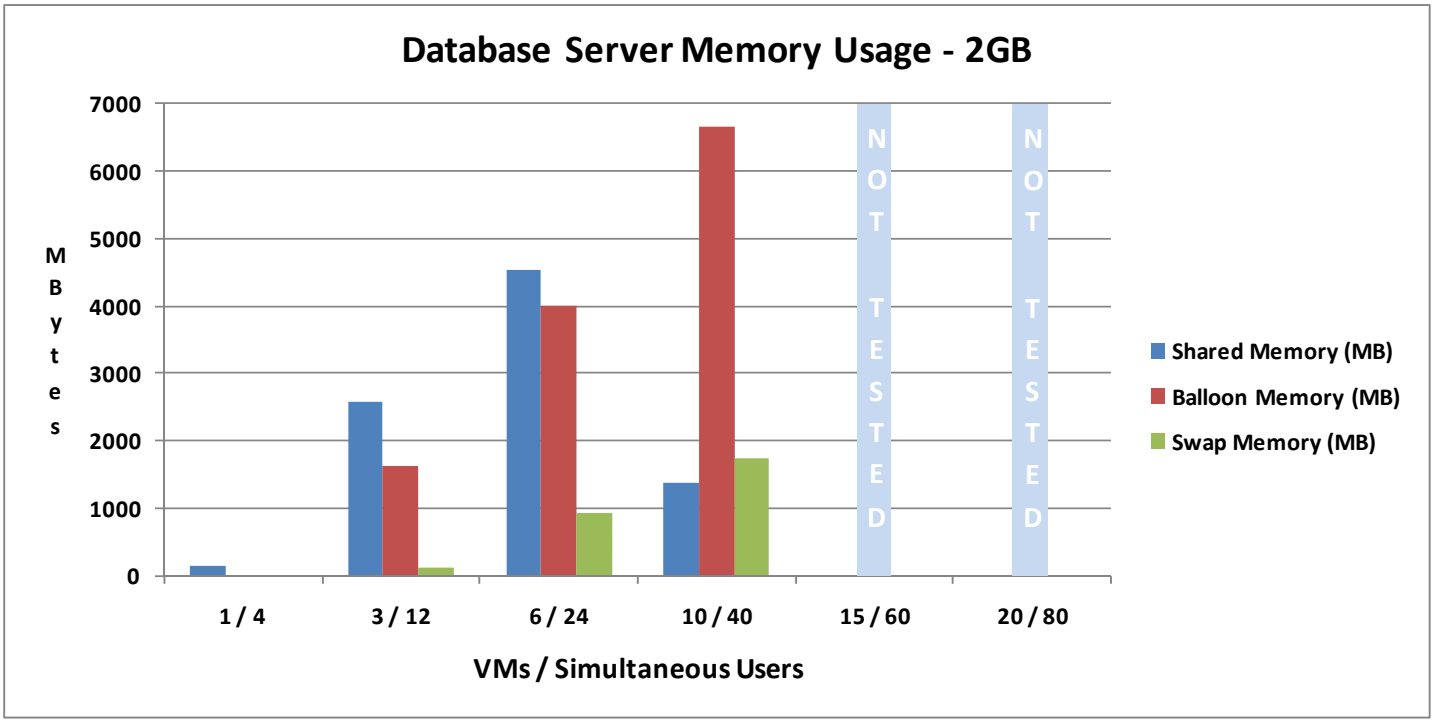


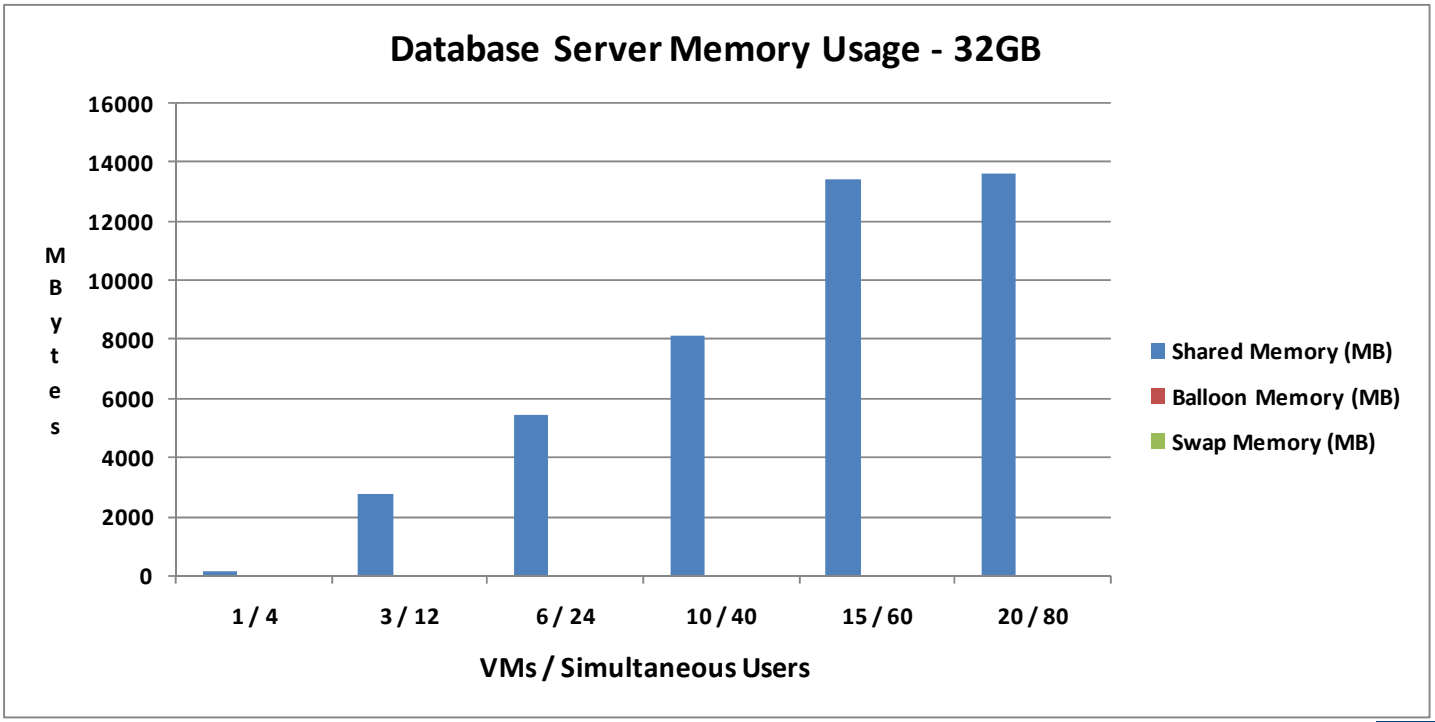


Database Server Memory Usage

The loads placed on the VMs during the database tests are much higher than for the web server tests. Each user being simulated by the TestCenter is requesting information from the web server which must, in turn, retrieve this from the associated SQL backend database.

As can be seen in the 2GB tests, memory sharing increase rapidly for the first two loads and then ballooning and page swapping to disk came into play as memory became seriously overcommitted. As with the web servers tests, 32GB of memory was sufficient at these loads to allow memory sharing only to be required.

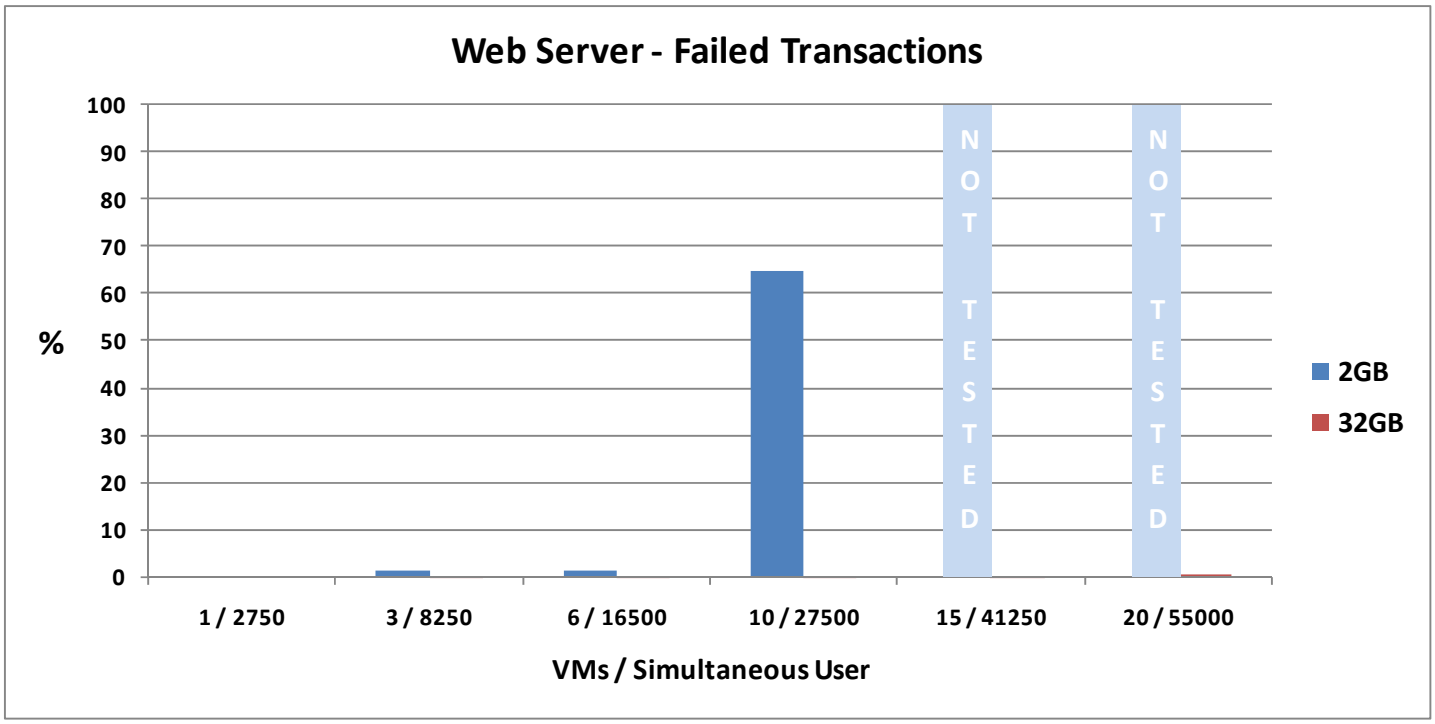


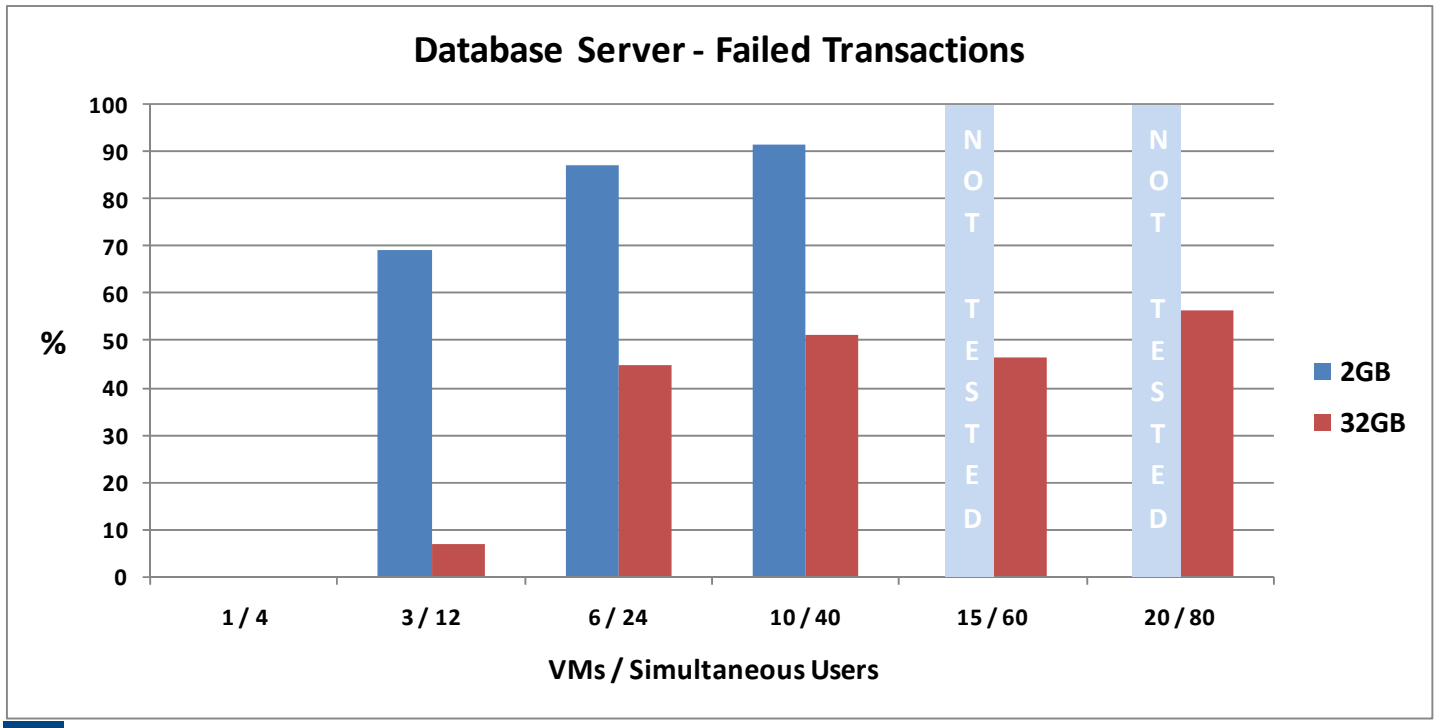


Server Failed Transactions

Web server tests with 2GB of memory show that failed transactions increased dramatically to 65 per cent. The server with 32GB of memory only failed to respond to less than one per cent at the highest load.

Of more interest is the failure rate for the database tests as with only 2GB of memory this reached an unacceptable 69 per cent with only three VMs loaded. However, even with 32GB of memory the VMs were increasingly unable to respond to user requests for data as the load ramped up. This shows clearly that considerably more physical memory would be required to service this number of VMs and users.





Conclusion

The results from these tests clearly demonstrate the effects the amount of physical memory can have on the performance of VM's running under VMware's ESX Server. With 2GB of physical memory as requested by Kingston Technology we immediately encountered problems as the server was unable to support more than ten VMs.

As we imported VMs we found the each subsequent job took longer and the eleventh import job failed to complete. Consequently, the jobs requiring fifteen and twenty VMs have been marked in the relevant graphs as not tested.

Web server response times for 2GB of memory became progressively worse and for ten VMs with a load of 27,500 users it actually reached a high of nearly eighteen seconds. For the database server tests with 2GB this was far worse with ten VM's and 40 simultaneous users having to put up with average response times of nearly one minute.

From the ESX Server memory usage statistics we concluded that for 2GB the only configuration the server could support without any memory management coming into play was a single VM. Anything beyond this and ballooning was initiated and, worse still, page swapping to disk.

Upgrading the server to 32GB of memory as requested by Kingston Technology allowed us to run our tests across the full range of VMs. However, although response times for the web servers never went above two seconds, the results for the database server tests were unacceptably high. With twenty VMs and eighty simultaneous users, the times for the servers to respond climbed to an average of over forty seconds.

ESX Server memory management had less to do with 32GB in the server with it reporting memory sharing as the only activity occurring. For all tests there was no evidence of ballooning or page swapping being required.

Although the 32GB of memory appeared to be handling the tests well the number of failed transactions as reported by the TestCenter was quite revealing. These are simulated user requests that do not receive any acknowledgement from the server they were directed to.

On the web server tests the percentage of failed transactions for 32GB never rose above 0.5 per cent. However, it failed to handle the higher demands of requests directed to the database servers with the final test using twenty VMs and eighty simultaneous users reporting an average number of failed transactions of 57 per cent.

Many businesses are moving to virtualisation to reduce operational costs but a failure to configure their servers correctly will have a negative impact on performance. Although testing was limited to server configurations with 2GB and 32GB of memory, the results clearly show the substantial performance improvements that can be made for VMs running web server and database services by upgrading memory.