

VMware Capacity Testing

Bill Gleeson

Scapa Technologies

6th April 2006

Introduction

This report describes the process and outcome of a five day stress testing and benchmarking engagement carried out by Scapa Technologies at

The purpose of the engagement was to determine by means of stress testing, the capacity and performance of a blade server hosting 32 Windows XP VMWare instances. Each instance would host a user running the SAP and Siebel applications in session. The requirement was to ensure that 32 simultaneous sessions would be supported without any degradation in performance, either at the server end – measured by an increase in resource utilisation, or the client end – measured in response time.

The Scapa Test and Performance Platform is an innovative and powerful application performance testing, diagnosis and monitoring product applicable across a wide range of commercial software technologies at multiple points in the application life cycle. In this case, it is being used to drive RDP sessions (one to each of the 32 VMWare images) that replay scripted user activity, while response times and performance across the architecture are monitored and analysed.

Approach

The approach to testing in part followed the Scapa Expedite methodology, where, usually, an initial set of simple scripts is used to determine ballpark figures for system performance and capacity. However, in this case, the processes followed by the scripts were not given much consideration, as the desired usage profile could be met by scripts that simply ran the applications, without too much focus on what the simulated users actually did.

The focus of the testing was on total system (where system refers to the single blade) capacity and performance, and any point of failure at that capacity. The important question being whether a single blade server would support 32 concurrent sessions running sample applications. Therefore, assumptions regarding application functionality were required, although the test sessions were monitored for any application failures.

Infrastructure and Settings

Testing took place on a test environment, consisting of 1 blade (000106) running VMWare ESX server, hosting 32 VMWare instances of Windows XP (V XXXXXXX001086). The Scapa Server agent and the Wintask 3.0d scripting runtime were installed on each XP image under the C:\Program Files directory. A shortcut to the Scapa Server MetaAgent (metagent.exe RDP) was placed on each image's Startup folder. This agent is required to execute initially for every automated test session – the shortcut can be disabled when testing is not in operation.

The Scapa Test and Performance Platform 3.2.1 was installed in its default directory on a provided desktop machine (\ 100189093). This machine was also used as a test client, from where the test RDP sessions were launched. The version of Scapa was latterly upgraded to 3.3, primarily to take

advantage of the improved reporting capability of that version.

A single test user name (testlin) and password was provided for the UK domain. A list of 16 logins was provided for SAP (each login to be used twice), and a single user for Siebel. Passwords for the SAP and Siebel users were reset. An initial running of each application was required to finalise the installation.

Applications

There were two hosted applications to be considered for simulating a user activity, primarily:

- SAP Utilities (using the SAPGUI client)
- Siebel eEnergy

Scripts

Initially, separate scripts were created for each application. These were eventually amalgamated into a combined script (combo.src) that would simulate the following user activity:

| | |
|----------------|---|
| Initialisation | Open an RDP session and log in with credentials |
| | Open SAP and login |
| | Open Siebel and login |
| Execution | In SAP, open a customer account and make an amendment to the record. Close and save the record. |
| Finalisation | Close SAP |
| | Close Siebel |

The scripts were stored on a share (\\stainex01\log_apps\scripts) that was accessible to each of the XP instances – each instance had this share mapped to its local Z: drive.

Results and Observations

Initial Findings - Baselineing

Following the Scapa Expedite methodology, the first task, once the infrastructure had been set up and the scripts created and edited, was a sanity check; would the infrastructure support 32 logged in users, without any of those users undertaking further activity?

The Execution phase of the script, referred to in the previous section, was altered to merely pause for 60 seconds – the simulated users would then initialise both applications, and merely keep them open, with a 60 second loop being repeated as many times as required. Any deviation in response times from 60 seconds would show a performance issue.

The following observations were made. Note that response time refers to the total time taken to perform the loop (in this case something that is known to take 60 seconds under optimal conditions) **as measured from the client side.**

| <i>Usercount</i> | <i>Response time (not including first iteration)</i> | <i>Notes</i> |
|------------------|--|---|
| 1 | 60 | |
| 10 | 60 | |
| 30 | 60 | No machine indicating CPU use > 20% or memory use >30%. Backend ESX server layer usage averaged 0.57 of optimum |

The steady response time, even at the maximum population (one user failed, and one image did not have SAP installed) is excellent news – the only deviation in behaviour came during the login phase.

Logins were handled sequentially; as one login (start RDP session to an image, send username and password twice, open desktop, open SAP, pass username and password, open Siebel, do likewise) completed, another began. This approach was necessary due to the requirement to repeat the passing of initial credentials. The first 12 sessions all logged in in a fairly standard time – between 35 and 40 seconds. However from the 13th session onwards, large deviations in login times were noted. It became apparent that large CPU and memory use was being reported by Virtual Centre at that time.

Subsequent running of the same test (the following day) did not show this discrepancy in login times, and it was further noted that some background activity had been present on the system at that time – the most likely cause of the lengthy login times.

Subsequent Findings

Once baselined, the target script was amended back to the original version – opening both applications and processing customer records. A list of 999 customer records was provided.

A detailed record of one of the test runs in this configuration is found in the accompanying “32 users 20060406-154501.pdf” attachment. Logins completed in an average of **54** seconds, which as considered acceptable as this time included the initialisation phase outlined above.

Once in the execution phase, high CPU utilisation at the ESX layer of **5.4** (times optimal capacity) was noted, and the client side response time began to vary widely. However individual VM Image metrics did not indicate high per-machine processor, network, disk or memory usage.

From a VMWare technical note:

A load average of 1.00 means that the ESX Server machine's physical CPUs are fully utilized, and a load average of 0.5 means they are half utilized. On the other hand, a load average of 2.00 means that you either need to increase the number of CPUs or decrease the number of virtual machines running on the ESX Server machine because the system as a whole is overloaded.

A reading of > 5 would then initially lead to the conclusion that the ESX server is drastically overloaded, but before this conclusion is taken, some points on the nature of this test must be considered:

- The goal of the test was to provide 32 simultaneous sessions, not the loading therein
- The tests were run constantly, with little think time, at a high throughput
- No consideration was made of application functionality, other than response times

Conclusions

In summary, the ESX Server could support 32 simultaneous sessions, each of which had two core applications open. Utilisation of resources on the server varied greatly between the two observed points after that of idleness and full activity, but given that typical observed usage patterns are usually closer to the former, this should not be a cause of concern.

Further testing should concentrate on mimicking more accurately expected usage patterns, perhaps by including some additional applications into the testing mix.

One area of concern is the adverse effect that some types of management jobs (Alteris, rebuilding images etc.) appear to have on resource consumption and response times. During several periods of the engagement, a sluggish system was attributed to such activities – this should be verified by retesting when a known job is occurring.