Scalability of RULE on Workstation PCs

Clancy Malcolm
Centre for Advanced Internet Architectures. Technical Report 031218A
Swinburne University of Technology
Melbourne, Australia

***Abstract*-This technical report describes the scalability of the Remote Unix Lab Environment (RULE) system when deployed on workstation class PCs. Two different scenarios were tested: performing standard tasks using a shell and serving web pages with content sourced from a database . The RULE system was found to perform satisfactorily with up to 100 virtual hosts on a single workstation PC and RAM usage was found to be the main factor limiting scalability.**

Keywords- Virtual Hosts, Scalability

I. Introduction

The Remote Unix Lab Environment (RULE) was first deployed at Swinburne University in semester 1 2003, on low-power VIA main-boards with ESP 5000 (500 MHz Celeron-Equivalent) processors [1,2]. Each host ran 5 virtual hosts that were used by students developing Java-based network applications. The ESP 5000 series is considerably underpowered for compute-intensive applications [3]. However, these hosts were more than capable of performing the required tasks and this inspired us to experimentally measure the scalability of the RULE system on workstation class PCs.

The workstation model chosen for the experiments was one of the University's standard workstation models at the time of writing – an HP Evo D530 with a 2.66GHz Pentium 4 Processor and 512MB of RAM [4].

A number of enhancements were made to the software used to configure the RULE – the Jail Host Toolkit (JHT). Support for virtual node (VN) devices was added to overcome limitations imposed by using physical disk partitions and additional administrative tools were developed to make it easier to maintain a large number of virtual hosts. These enhancements will be included with the next release of JHT.

Two tests were used to assess the scalability of RULE. The first test modeled a student performing an introductory Unix exercise where they performed fundamental shell operations such as making a directory and editing a file. The second test was to request a database generated web page, as may be performed by a student testing their own web application or using a web application to administer their virtual host.

II. Scalability of Fundamental Shell Operations

The aim of the first test was to model a class of students completing an exercise where they performed basic operations at a shell prompt. To model this scenario a shell script was written with the following steps:

1. Make a directory
2. Change to the new directory
3. Create a “Hello World” shell script
4. Make the shell script executable
5. Run the shell script
6. Delete the shell script
7. Change to the parent directory
8. Delete the directory created in step 2

A student would typically issue these commands by typing them at a shell prompt with a small period of time between each step while they comprehended the results from the previous steps and read the instructions. To model this delay the script used the 'sleep' command before each of the above steps to suspend execution of the script for 20 seconds. A 20 second delay was also used at the end of the script to model the student examining the results of the commands before finally disconnecting. Hence there were 9 delays with a total duration of 180 seconds.

A test client was configured to run the shell script using SSH. The SSH client used public-key authentication so that it could be executed without user intervention. The client's test script connected to each of 100 virtual hosts running on RULE and measured the total amount of time from invoking SSH to the command completing. The 180 second delay was then subtracted from this time to give the net time consumed by the test.

The client script was multi-threaded to permit concurrent connections to the server. In a lab environment not all students would start the exercise at precisely the same time and hence a random delay of between 0 and 5 seconds was used between opening each connection.

Figures 1a and 1b show the cumulative histograms of the test times when using 30 and 100 hosts respectively. Figure 1a illustrates that with 30 virtual hosts per primary host all tests had a time of less than 2.5 seconds. With 100 virtual hosts per primary host (Figure 1b)

71% of tests had a time of less than 2.5 seconds and 9% had a time of more than 5 seconds. Although there is a noticeable degradation in performance, the performance is probably still adequate for most learning environments.

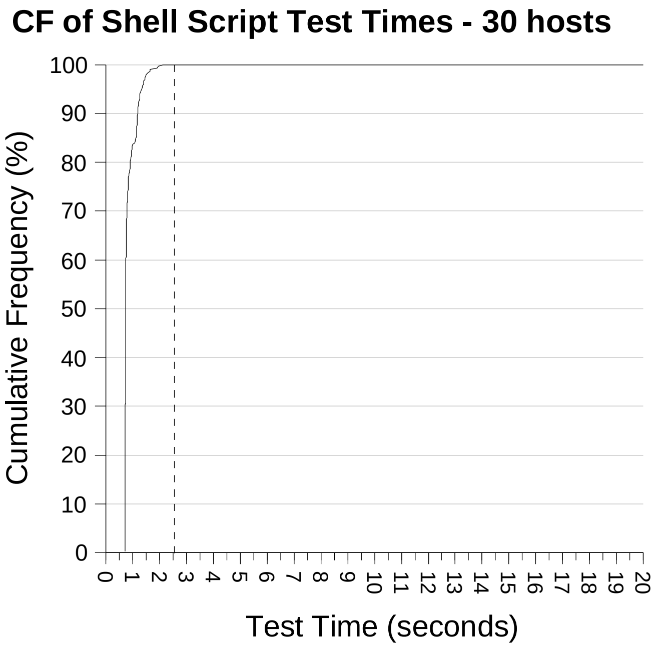

Figure 1a Cumulative Frequency of Shell Script Test Times with 30 virtual hosts

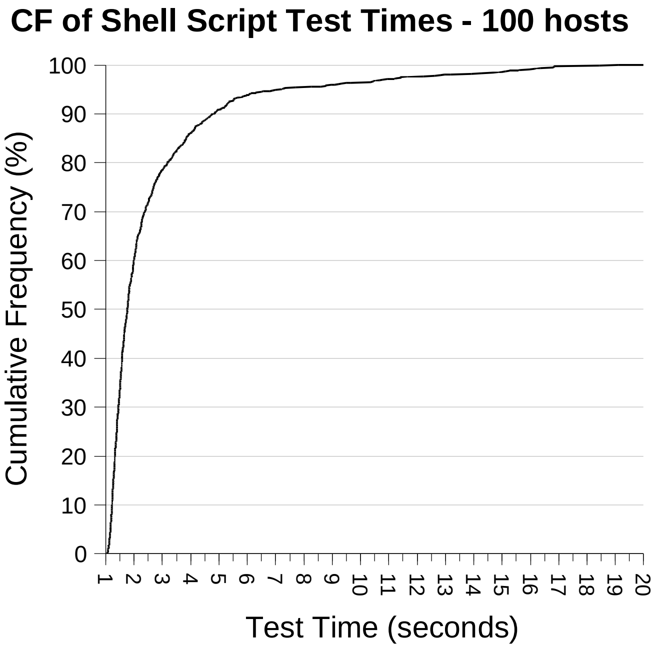

Figure 1b Cumulative Frequency of Shell Script Test Times with 100 virtual hosts.

III. Scalability of Database Generated Web Pages

Another typical usage scenario for RULE is to allow users to become familiar with installing and using network application servers. The memory usage caused by running servers such as web servers and database servers warranted running a separate experiment for this scenario.

To model this situation we installed the following software on each virtual host:

- Apache HTTPD 1.3.27 (web server)
- MySQL Server 3.23.55 and MySQL Client 3.23.55 (relational database engine)
- Mod PHP 4.3.1 (scripting language module for Apache HTTPD)
- phpMyAdmin 2.3.2 (web-based front-end for MySQL database administration)

This software is typical of that used in Unix-based web hosting solutions.

HTTP connections were used to request a list of database users from each virtual host and the time taken to load each page was measured. 10 concurrent connections were used and requests were distributed between virtual hosts in a random fashion. The experiment was repeated using between 1 and 45 virtual hosts in the pool and the average page load time was taken for each pool-size. These results are illustrated in Figure 2.

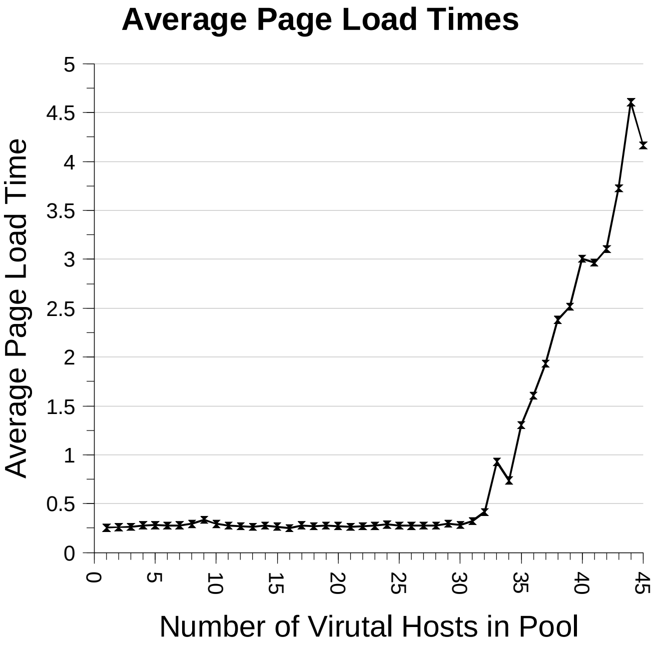

Figure 2. Average page load times for a database generated page for a range of virtual host pool sizes.

The results indicate that the average page load time was less than 0.5 seconds when 32 or less virtual hosts were used. When more virtual hosts were used the page load time increased rapidly. Running the 'top' utility on the primary host indicated that the performance degradation with more than 32 hosts was caused by having to constantly transfer data between physical RAM and hard drive swap files.

It was possible to have more than 32 virtual hosts 'booted' without significant performance degradation in this test, providing that no more than 32 virtual hosts were being used 'simultaneously'. Processes running on the unused hosts would be transferred to the swap file, allowing the active hosts to utilize the physical RAM. When one of the inactive hosts becomes active there is a noticeable delay of many seconds to retrieve its processes from the swap file, after which it continues to run at full speed for as long as it remains in physical RAM. This could be useful, for example, where a primary host had 60 virtual hosts but typical lab sizes were 30 students or less.

IV. Verifying That RAM is the Limiting Factor

To verify that the performance in these tests was primarily limited by the amount of RAM on the host we increased the RAM to a total of 1GB and repeated the database generated web page test. The results are shown in Figure 3.

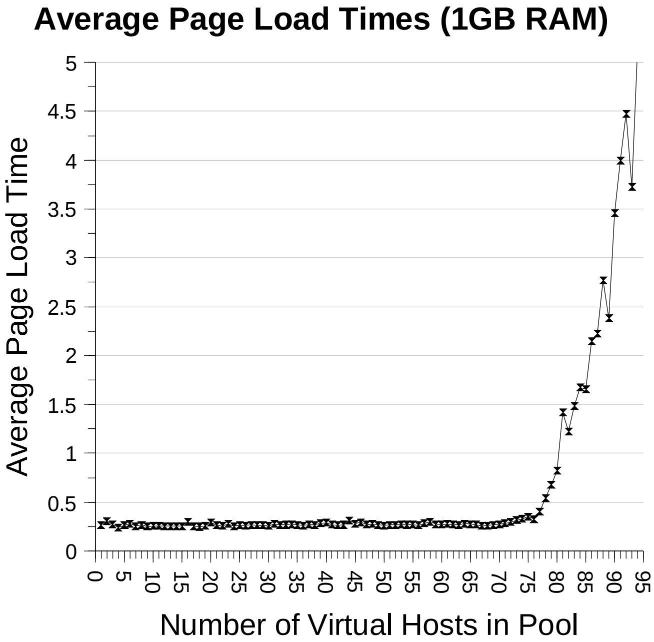

Figure 3. Average page load times for a database generated page when total RAM is increased to 1GB. Note the scale on the X axis is different to Figure 2.

The results show that by doubling the workstation's RAM there could be more than twice as many virtual hosts in the pool before the average page load time increased above 0.5 seconds (77 hosts compared to 32) .

V. Conclusion

The workstation class PC used in these experiments was capable of running 30 active virtual hosts being used concurrently for both shell operations and serving web applications. Running 100 virtual hosts on a single workstation is feasible, although it introduces some additional delays for shell operations and can become impractical if more than 32 virtual hosts are being used as web application servers at one time.

RAM was found to be the main factor limiting scalability, particularly for web applications, so increasing the amount of RAM could lead to significant improvements.

References

[1] GJ Armitage, "Maximising Student Exposure to Unix Networking Using FreeBSD Virtual Hosts," CAIA Technical Report 030320A, March 2003 (http://caia.swin.edu.au/reports/030320A/CAIA-TR-030320A.pdf)

[2] Via Technologies Inc, http://www.viapsd.com (as of December 2003)

[3] EPIACENTER.com, "Epia M10000 Review," http://www.epiacenter.com/modules.php?name=Content&pa=showpage&pid=21 (as of December 2003)