USING SIMULATION TO PREDICT PERFORMANCE CHARACTERISTICS OF MIRRORED WWW HOSTS

Dr. Dennis Guster, St. Cloud State University, MN, guster@stcloudstate.edu
Dr. Paul Safonov, St. Cloud State University, MN, safonov@stcloudstate.edu
Charles Hall, St. Cloud State University, MN, chall@bcrl.stcloudstate.edu
Dr. Richard Sundheim, St. Cloud State University, MN, rasundheim@stcloudstate.edu

ABSTRACT

Distributed processing has provided a number of advantages to network designers. These service advantages include better performance, backup and availability potential. This paper uses simulation to predict network performance of three host models: single host, dual hosts and dual hosts with mirrored file systems. The results indicate that the dual models do offer better performance potential and that mirroring file systems is practical given the level of processor overhead it generates.

Keywords: Webservice, WAN, distributed processing, mirrored file systems.

INTRODUCTION

The success of the internet and the effectiveness of browser based delivery have necessitated more sophisticated service delivery schemes. It is no longer enough to bring up a web site and hope for excellent reliability and performance. Rather careful planning and testing need to be used to ensure that the site will handle the expected load, have high availability and be easily recoverable in case of disaster. Distributed processing has provided a number of advantages to network designers. The logic is relatively simple, instead of running the web service on a single computer system, run it on two or more systems. This strategy puts more processors on the job which should increase performance and if the file systems are mirrored provides distributed backup. Furthermore, if the servers are truly mirrored one machine can fail and the other one(s) can continue to provide the web service. These service advantages include better performance, backup and availability potential (1). Therefore, to meet the growth expected in web services some form of distributed processing merits review.

Providing fault tolerant computing on PC based platforms is not new. In fact, Novell Netware introduced that service over 10 years ago (2). The recent surge in Linux as a services platform has sparked interest in deploying fault tolerant Linux based clusters. Even though Linux is a very stable operating system, it is only as reliable as its hardware, which typically is the point of failure (3). In fact, some integrators claim that a Linux cluster can yield a 99% availability parameter (4). Furthermore, if load balancing is enabled, improved performance can be realized as well. Load balancing involves a mechanism to evenly distribute the workload among the hosts in the cluster. Some algorithms are relatively simple such as programming the DNS (domain name server) to alternate the requests between/among the hosts in the cluster (1). Others are more complex, such as topology and policy based redirection in which performance characteristics of the network are evaluated and a dynamic algorithm is devised (1).

Synchronization of the file system between/among the cluster nodes is an integral part of this process. However, the packet traffic between/among nodes can have a detrimental impact on both network and processor performance. One way to minimize its effect on the network is to carry the synchronization traffic on a separate physical network often termed the back channel. However, clustering and distributing an application generates overhead beyond what occurs in a single system. In fact, adding additional processors to solve a given problem will probably not increase performance linearly (5). In some cases, the communication overhead among the processors is so great that the problem can be solved more quickly in serial than parallel (5).

When designing networks it is difficult enough to predict performance on a single host. Clustered fault tolerant hosts with mirrored file systems are much more complex and offer even a bigger challenge. Too often network designers elect the "brute force hardware method" in which they purchase the most powerful hardware available and hope for the best. Sometimes the hardware is configured with weaknesses in network interfaces or disk I-O capabilities and the required performance is still not realized.

This paper will attack this design problem by using computer simulation to determine processor and network utilization under a variety of loads and host configurations. To illustrate this process WWW traffic coming from three WAN links in a mythical company will be used to predict performance under the following configurations: a single host, dual hosts, and dual mirrored hosts with a back channel.

NETWORK CONFIGURATIONS AND SIMULATION STRATEGY

The basic configuration of the mythical company is relatively simple. In the first model, there are three WANs each supporting LAN connected computer workgroups. The WANS are connected to a central high end router. The router in turn is connected to a DNS server, which in turn ensures the appropriate packets are routed to the web service which is housed on a single stand alone host. The second configuration model is the same as the first except dual hosts provide the processing power for the web service. The third and final configuration model is the same as model two except that the dual hosts' file system is mirrored via a back channel.

The simulation was programmed with *Comnet* software. This point and click tool allows the programmer to select various network devices, connect them with a variety of network architectures, and use standard queuing theory to determine network performance delay. The critical part of using this software is determining packet inter-arrival distributions and packet size distributions. This is difficult because some research has shown that packet-inter-arrival rates often do not follow any know distribution (6). It is critical that the correct distribution be utilized or the validity of the simulation could well be suspect (6). To ensure representative distributions are utilized in the simulation, samples of packet data were recorded on the authors' network for the different types of traffic required and converted to tabular distributions. Because the primary application is web service, the first distributions determined were the packet inter-arrival rates and packet size distributions of http traffic. A 360 packet sample was extracted from the authors' network during a peak period using TCPDUMP. Table 1 provides descriptive statistics for each of these samples.

Table 1
Descriptive Statistics for the Sample Used to Generate the WWW Traffic

	Mean	St. Dev	Min	Max
Packet inter-arrival time	1.37	11.35565	0.00001	179.68
Packet size	597	677.643	66	1514

To devise a valid simulation a second set of distributions had to be created. Those distributions needed to describe the packet inter-arrival rate and expected packet sizes for updating the mirrored file system. This was the traffic that would travel across the back channel. A sample of this type of data was obtained from the authors' network using a NAS (network attached storage device) with RAID capability. Table 2 describes these samples.

Table 2
Descriptive Statistics for the Sample Used to Generate the Back Channel Traffic

	Mean	St. Dev	Min	Max
Packet inter-arrival time	0.078963	0.353918	0.000041	3.05951
Packet size	288.509	153.8333	60	528

RESULTS OF THE SIMULATION

The main purpose for undertaking the simulation was to determine how processor utilization would scale from a single to dual and to mirrored dual processors. The initial simulation was run with 15 workstations per WAN accessing web services based on the distribution defined in Tables 1-2. The results of that simulation and subsequent runs using the same simulation logic, but involving WAN workstation groups of 30 and 50 devices, is depicted in Table 3.

Table 3
Processor Utilization (%)

	15WSperWAN		30WSperWAN		50WSperWAN	
	P1	P2	P1	P2	P1	P2
Single processor	18.13	N/A	29.70	N/A	56.29	N/A
Dual processors	8.44	8.43	18.04	18.21	27.50	28.02
Dual processors (Mirrored)	26.85	36.78	29.64	37.09	29.58	43.22

It is clear that the process of mirroring file servers does generate a certain amount of overhead. In all cases, the utilization rate is greater for the mirrored pair than the non-mirrored processor pair. However, that overhead percentage seems to decrease as the number of active workstations is increased. Also, note that workload for the non-mirrored processors is about half that of the single processor. This would be expected if the load balancing algorithm is operating as planed. However, it appears that the burden of the mirroring process is not distributed equally. In all cases, processor number 2 exhibits about 10% more utilization. The magnitude of the workload is dangerously high for the single processor when loaded with 50 workstations per WAN. Basic queuing theory states that when 80% utilization occurs the system tends to lock up and become unusable. The magnitude of 56.29% indicates the single server is at capacity and any increase in workload would necessitate some form of host upgrade. Both dual server models still have some room for expansion beyond the 50 workstation per WAN load level. However, the data collected through the simulation about load levels on each of the WANs clearly shows that growth would not be practical unless the WAN capacities are increased. Each WAN is based on the T1 architecture in which the line speed is about 1.5 Mbs. Table 4 provides link utilization rates for each of the WANs at the three different load levels.

Table 4
WAN Link Utilization (%)

	WAN 1	WAN 2	WAN 3
Single			
15WS	25.87	30.48	35.88
30WS	44.99	63.10	51.30
50WS	99.95	99.91	99.91
Dual			
15WS	26.96	30.68	22.20
30WS	63.14	51.12	80.45
50WS	99.97	99.90	93.00
Dual-Mirrored			
15WS	39.22	22.19	29.01
30WS	82.12	70.98	57.53
50WS	99.99	81.07	99.95

This data reveals that at the 50 workstation load level all models are beyond 80% capacity, and the WAN links will become the system bottleneck. Even at the 30 workstation load level some values are in the 80% magnitude and other values are beginning to approach that level. For the most part the values are remarkably similar from model to model and differences could be explained by the variation in generating the distributions from tabular format. It is interesting to note how these robust utilization rates affect the flow of data in other parts of each network model. Table 5 provides that link data.

Table 5
Utilization on Other Links

	Router to DNS	DNS to Host(s)	Back Channel
Single			
15WS	15.73	1.57	N/A
30WS	26.93	2.69	N/A
50WS	49.68	4.96	N/A
Dual			
15WS	13.94	1.39	N/A
30WS	34.20	3.42	N/A
50WS	49.00	4.89	N/A
Dual-Mirrored			
15WS	16.23	1.62	0.21
30WS	35.65	3.57	0.13
50WS	47.40	4.74	0.08

The router to DNS connection is an Ethernet 10Mbs link. Even at the 50WS load this link is able to handle all the traffic the three WAN links are able to feed it. Both the DNS to Host(s) and back channel links are Ethernet 100Mbs connections and have plenty of capacity for future expansion. The host models can effect response time on the workstation level. Table 6 provides the average roundtrip delay in milliseconds from each workstation to the host(s).

Table 6
Average Round-Trip Delay to the WorkStation in Milliseconds

	Work Group 1		Work Group 2		Work Group 3	
	Host 1	**Host 2**	**Host 1**	**Host 2**	**Host 1**	**Host 2**
Single						
15WS	312	N/A	425	N/A	485	N/A
30WS	935	N/A	1051	N/A	1056	N/A
50WS	1373	N/A	1759	N/A	1323	N/A
Dual						
15WS	0595	701	605	692	514	418
30WS	1054	1308	1097	1108	1465	1529
50WS	2286	2439	2089	1896	1740	1892
Dual-Mirrored						
15WS	558	634	576	474	400	456
30WS	1543	1309	1273	1281	1118	761
50WS	1563	1566	1171	1150	1494	1594

In most cases the single host model provides smaller roundtrip delays from the workstation to host and back than does the other two models. This would make sense because requests do not need to go through the load balancing routine on the DNS server. In both dual host models, the magnitudes of values within each category are similar which indicates the load balancing routine is working fairly well. One might expect that the dual-mirrored model would experience higher delays that the dual model. This would be expected primarily because of the added overhead of running the mirroring process. The data do not support that assumption. In fact, in many cases, the delays are greater for the dual model than the dual-mirrored model. There may be two explanations for this situation. There was plenty of CPU power available for the dual-mirrored model. The utilization values were below 50% and, hence, the additional mirroring workload did not affect performance like it would have if its addition caused utilization to migrate into the 80% range. Another explanation is the simulation was not run long enough so the true workload samples were not generated from the tabular distributions. Perhaps if the same simulation were run for a much longer period of time, the two dual models would experience delay values of a more similar magnitude.

VALIDATING THE SIMULATION

In an effort to validate the data obtained in the simulation, a test-bed was set up and data collected. This test-bed was configured for three trials, one trial for each of the three models described earlier. However, the workstations were connected via a switched LAN, and there were 45 of them. Forty five were selected to match the number of stations in the 15 per workstation group across the three WANs. Each workstation ran the same script file that generated the WWW traffic. The packet inter-arrival and packet size distributions were not controllable from the script so this data's representativeness to the distributions used in the simulation was not known. Table 7 depicts the processor utilization results.

Table 7
Test-bed Processor Utilization in %
(45 Workstations Total)

	P1	P2
Single proc.	24.68	NA
Dual proc.	15.92	15.927
Dual proc. Mir.	27.42	27.68

The results are quite similar to the results observed in Table 3, the 15 per workstation column. The single processor utilization was about 6% higher, each non-mirrored server about 7% higher and one of the mirrored server readings was almost identical while the other was about 9% higher. The actual data reveals that the load balancing routine worked well, which didn't occur with the mirrored hosts in the simulation. The values observed are encouraging and tend to validate the core simulation logic. The variance observed could be mainly attributed to differences in the distributions used in the simulation. In order to determine how close the packet inter-arrival rates observed in the actual data match the distributions used for both the WWW and mirroring traffic, a goodness of fit analysis was undertaken. As suspected none of the original distributions or the ones collected from the actual data fit any know distribution. Table 8 displays the descriptive statistics for the actual data inter-arrival distributions. These values are quite a bit different from the distributions used in the simulation depicted in Tables 1 and 2. The fact that the means are smaller can be attributed in part to the workload generator.

This software continuously executed a script that ran a WWW session which was repeated for the duration of the data collection time frame, so there was little pause between packets. Whereas, in the simulation, a log of real WWW traffic was sampled, which did contain pauses. It is not realistic to expect a continuous stream in a production setting, so the distributions generated during the experiment are probably a worst case scenario. It would be interesting to plug them into our existing model and see how they effect processor utilization.

Table 8
Actual Packet Inter-arrival Times

	Mean	St. Dev	Min	Max
WWW	.0005366	.0024454	.000003	.030912
Mirroring	.0001063	.0001628	.000002	.002672

Because the means were much smaller in the actual data, the traffic patterns were significantly more intense. In fact these distributions were so much more intense than the simulated values that were generated as tabular distributions that they locked up the simulation program. In the single host model the simulation ran for only 2.4 seconds before locking up. In both dual host models the simulation software locked up almost immediately. Data from a checkpoint file in the single model revealed that processor utilization in the simulation had reached 74%, up significantly from the 18% observed in the initial simulation.

CONCLUSIONS

Therefore, the results from the test bed only partly validated the simulation. This was due to the fact that the data generating script was not representative of production data. A better test would have been to setup the various models and allow end users to access their web services for a set time. However, it is clear that if the hosts can handle the scripts as well as they did, and stay well under the 80% utilization limit, all of the models have promise. The data support the concept of adding additional processors to increase performance. Also, it appears that the overhead associated with mirroring file servers is minimal and implementing that model would be supported by both the simulated and actual data obtained herein.

REFERENCES

1. Buyya, R. (1999). *High Performance Cluster Computing*. Prentice-Hall.
2. The Encyclopedia of Computer Security. (1998). http://www.itsecurity.com/tutor/novellsftiii.htm.
3. Lewis, P. (1999). “Strictly On-Line: A High-Availability Cluster for Linux”. *Linux Journal*. 64, August 1.
4. Uniware Systems Limited. (2003). http://www.uniware.net/faulttolerance.html .
5. Guster, et al. (2003). “Adaptation of a Parallel Processing Technique used to Solve Physics Problems to Computer Network Management Applications”. Proceedings of the Information Resource Management Association Conference, Philadelphia, PA.
6. Partridge, C. (1993). The End of Simple Traffic Models. (Editor’s Note), *IEEE Network*, 7(5).