Modeling and Analyzing Distributed Computing Environment with Petri nets
Case Study : Windows NT*Domain based environment

Jongwook Kim Young Hee Lim

Computer, Information and Automation Team
hyundai Electronics Industries Co., Ltd.
Ichon, Kyungki 12180
Korea

Abstract

Microsoft's Windows NT based systems are replacing the role of lower and middle range UNIX based systems in various fields. There have been many studies on the modeling and analysis of Unix based distributed computing environment for the resource allocation problem. Although, the basic principles may be similar to those of Windows NT based environment, the whole scheme is somewhat different from the traditional approaches. Extensive experiments are conducted varying the number of servers in a domain from one to ten, changing the number of clients from one to two hundred, processing number of transactions and transferring big chunks of data. The authors proposed a simple time Petri net model for the Windows NT client server environment based on the experiment results. The simle analysis can be done with this model, otherwise impossible.

1 Introduction

Downsizing and server client architecture seemed to be the theme of the eighties. But the growth of the internet in terms of number of participants and size spurred the server client architecture to invade almost everyday life even to a small office. Windows NT was introduced years ago, but as they adopted more popular TCP/IP protocol more and more, it is replacing many higher stand alone computing environments and even lower mid range server client environments. As the computing environments distributed wider and wider, the need for appropriate resource allocation became more critical. In this paper, a simple time Petri net model based on both architecture and experimental results were proposed in order to solve the resource allocation problem. Section 2 defines the Petri net and time Petri net model briefly. In section 3, the experiments and their results are described. Section 4 introduces the simple time Petri net model. Section 5 concludes the paper with brief summary and future direction.

*All trade marks and registered marks belong to the appropriate companies

2 Petri nets and time Petri net models

Definition 2.1 *A* **Petri net** *[4], denoted PN, is defined as a five-tuple (P,T,I,O,H) , where P, T, I, O and H are the set of places, transitions*

Definition 2.2 *A* **Time Petri net model** *[7] denoted $TPNM$ is a quadruple (PN, m_0, Λ, EP) where*

- *PN is the underlying Petri net.*
- *m_0 is the initial marking.*
- *Λ is the firing time function which assigns a positive real valued firing time to a transition defined as $\Lambda : T \rightarrow (0, \infty)$*
- *EP is the execution policy of the transitions.*

For a detailed discussion of the behavior of a time Petri net model, refer to [7].

In this paper, an arc, a place, an immediate transition and a timed transition will be represented by an arrowed line, a circle, a thin bar, a thick bar, res

0-7803-4053-1/97/$10.00 © 1997 IEEE

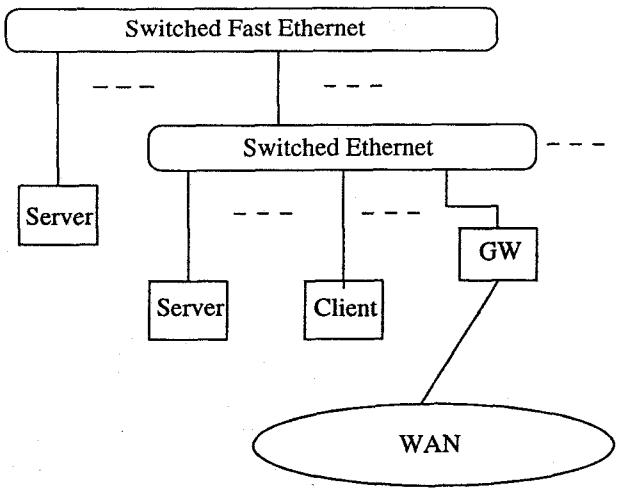

Figure 1: The Network Structure of SYSTEM IC Lab.

3 Experiments

The objective of the experiments was to monitor the server performance while changing the workload for the servers by varying the number of clients and servers.

3.1 Methods

The experiments were conducted for PC clients in System IC R&D Center of Hyundai Electronics Industires Co., Ltd. There are about 260 PC clients running on Microsoft Windows 95 OS Korean version. About half of the PC's are Pentium based and the other half are 486 based. They participate in a Microsoft Window NT domain. In addition to the PC's, about 150 Axil Workstations are connected to the same network. These are the mix of Sparc, Super Sparc and Ultra Sparc processors based workstations running on SunOS 4.1.4K and SunOS 5.5K. As illustrated in Figure 1, all the PC's and workstations are connected to a switched ethernet, some of the servers to a switched fast ethernet using TCP/IP protocols only. They participate in an internet domain, *sysic.com*. The local network is connected to the internet, PSTN and Hyundai Electronics' network through several gateways. The network is protected and hidden from outer world using firewalls and ip translation. In internet terminology, the local network is a single network in a single segment with no gateways, routers nor bridges. The (fast) ethernet switches perform *instant segmentization* of the network. The structured cable system was installed in November 1996. There have been no runt, giant packet and other electric signal failures which is caused by poor wiring after the installment including the duration of the expeirments. The experiments were conducted overnight, on holidays to insulate from other factors. Unix based workstations were working on their batched simulation jobs which requires quite large computational power and network bandwidth while the experiments were conducted. But even in the same network, UNIX workstations and PCs do not share much of the resources except for domain name service and mail service, which did not play a large role in the experiments.

Case	# DC	# File Server	Server NIC
1	1	1	10Mbps
2	2	2	10Mbps
3	1	1	100Mbps

Table 1: Experiment Condition

For case 1, 2 and 3, the clients were forced to log in to the NT domain such that user authentification occurs at the domain controllers. Then, 10 files whose total size is 75MB were copied from server to the clients in order to give enough and continuous stress to the file server. After the file copy actions, the clients initiated several SQL queries to the server. At the end of each step log were to made to the file servers. Table 1 shows the configuration of the servers for each experiment. The servers are HP Netserver LS 2 with dual 133MHz Pentium processors, 128MB main memories, 1GB OS disks and 4GB data disks for experiments via fast SCSI bus. One processor for each server was disabled, to give more stress to the processor. The login was done manually walking from PC to PC to mimic real life situation.

The server activities such as process, memory usage, system usage, network interface usage were gathered.

Case 4, 5 performed the same file copy actions between a UNIX server and clients, when the server had 10Mbps and 100Mbps NIC, respectively. File systems are established by automount NFS. Unlike PC clients, all the file activities were initiated simultaneously from remote session.

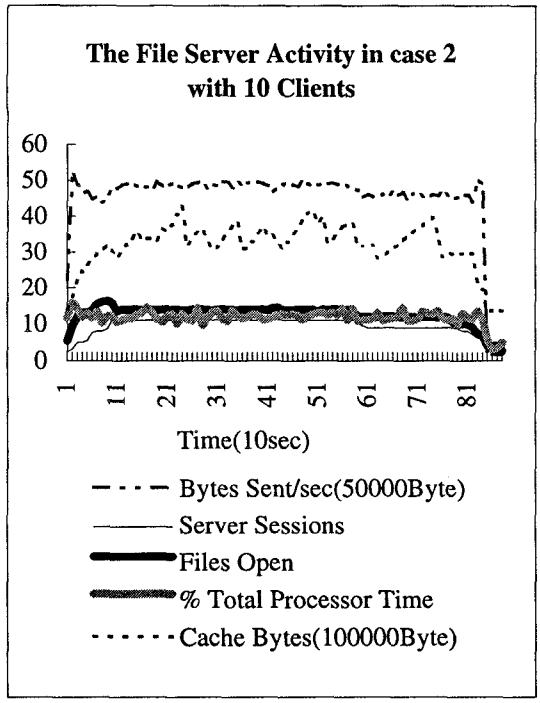

Figure 2: Typiecal File Server Activities

# clients	1	10	50	100	200
Case 1	87	878	3958	6387	
Case 2 server 1	88	482	1997	4126	
Case 2 server 2	88	422	1828	5025	
Case 3	88			825	1890
Case 4	86			6759	
Case 5	80			312	

Table 2: Time to Completion (seconds)

# clients	1	10	50	100	200
Case 1	70	834	3559	6662	
Case 2 server 1	69	378	2068	3592	
Case 2 server 2		433	1881	4653	
Case 3	70			7665	18452

Table 3: Data Put To Network (MB)

3.2 Results

Figure 2 shows activities at the file server for case 2 with 10 clients.

The main memory usage is almost flat (50MB) during the experiment for every case. It shows that neither processor (average processor time 17%) nor memory cache (average size 32MB) is the bottleneck resource. There was no noticeable activities in domain controller where SQL queries are processed. The results are almost same in cases 2 and 3 except for the items described below.

Table 2 shows the time for a file server to complete the service.

It shows that the time duration was increased proportional to the number of clients. For case 3, the magnitude of order of the duration was decreased by 1. It should be noted that in case 1, 2 and 3, if there was any untolerable network problem, the clients complainted for the problems and waited for user input. Therefore, we had to input user commands from client to client. This explains the fact that for small number of clients the time to complete the task for case 2 was half of that of case 1, but as the number of clients increased, the ratio was gradually decreased as the portion of the time to move from client to client was increased compared to the time for file transfer. In case of SunOS 4.1.4K workstations, they wait until the connection was re-established and finished the task. But SunOS 5.5K clinets dropped the whole connection and unmounted the file system, which resulted in that following file copy actions were wholly skipped.

Table 3 to Table 6 show the similar trends for each case.

# clients	1	10	50	100	200
Case 1	0	0	38	95	
Case 2	0	0	30	59	
Case 3	0			0	
Case 4				100	
Case 5				32	

Table 4: File Read Error (times)

# clients	1	10	50	100	200
Case 1	0	0	7.6	9.5	
Case 2	0	0	6.0	5.9	
Case 3	0			0	

Table 5: Data Not Transferred (%)

# clients	1	10	50	100	200
Case 1	0	0	8.0	10.7	
Case 2	0	0	5.7	5.4	
Case 3	0			0	

Table 6: Data Lost in Network (%)

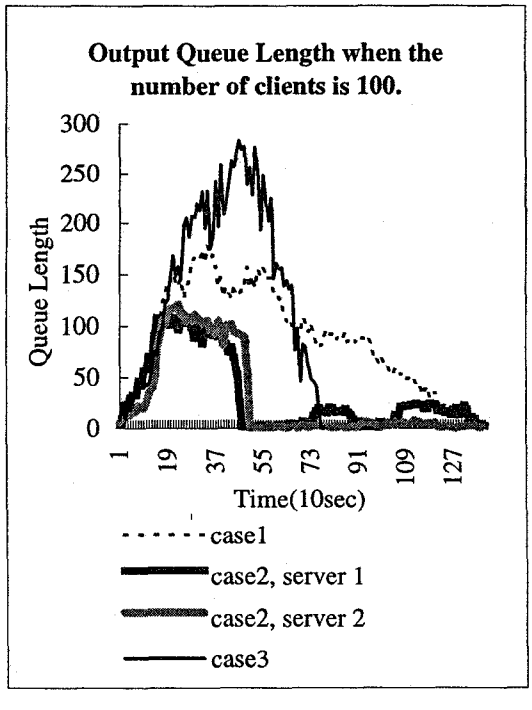

Figure 3: Output Queue Lengths of the File Servers

4 Petri net model

The network file read and write model for Windows NT environment are well described in [3, 5, 6, 9, 10, 11, 12]. The effect of having switched ethernet for file server whose service requested by many clients as in these experiments is to reduce the number of collision [17, 18]. The dramatic decrease in time to completion for case is explained by the fact that the switching equipment itself stores the data pumped out from the server in fast speed. This result in effective bandwidth growth for the server bandwidth by providing more opportunity for the clinets to access the data, even though the client side bandwidth still remains the same. There are models for lower layer protocols such as ethernet and TCP/IP stack. But it is unreasonable and impossible to analyze the lower layers since there are so many packets and stacks to model. Therefore, the authors proposed aggregated time Petri net model for this case. Figure 4 shows simple message connection model for client i.

- $p_{(i,0)}$: represents the state when client i requests message transfer to server.
- $p_{(i,1)}$: represents the state when client i waits for connection time out.
- $p_{(i,2)}$: represents the state when client i lost connection to the server.
- $p_{(i,3)}$: represents the state when client i waits for network availibility.
- $p_{(i,4)}$: represents the state when client i is notified time out.
- $p_{(i,5)}$: represents the state when client i succeed in sending message to server.
- $p_{(i,6)}$: represents the state when client i succeed in receiving data from server.
- $t_{(i,0)}$: represents the action when client i start sending requests message transfer to server.
- $t_{(i,1)}$: represents the action when client i connection to the network times out. $\Lambda(t_{i,2}))$ is given as the session time out key value in the registry.
- $t_{(i,2)}$: represents the action when the network sends time out message to client i.
- $t_{(i,3)}$: represents the action when client i obtain connection to the network.

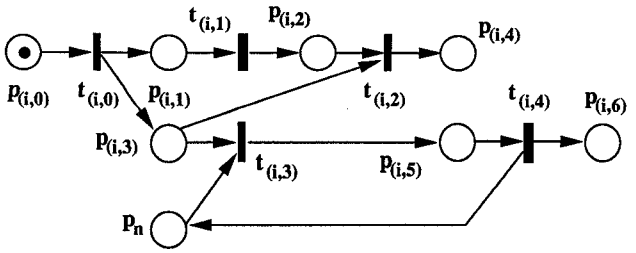

Figure 4: Message Connection Model for a Windows based client

- $t_{(i,4)}$: represents the action when client i start receiving data to the network. $\Lambda(t_{i,4}))$ is the time for the server to send the given data file via (fast) ethernet without contention.
- p_n: represents the state when the network is available.

This model is based on next facts:

- SMB (Server Message Block) time out is critical than the Frame Acknoledgement Time Out, since frame acknowledge time retries with binary exponential backoff algorithm while SMB time out is fixed [15].
- Server service of the server is always available compared to the network interface of the server.

These were confirmed by the experiments. It showed that once an SMB connection was made, the file transfer completed its mission.

Therefore, the final model of the Windows clients contending for a server attention is shown in Figure 5. In this case, we must set $m(p_n)$ to n_m, $\Lambda(t_{i,2}))$ to n_m times $\Lambda(t_{i,2}))$ and $\Lambda(t_{i,4}))$ to n_m times $\Lambda(t_{i,4}))$. Where n_m is the number of concurrent sessions succeed in connection to the server, which depends on the server network interface.

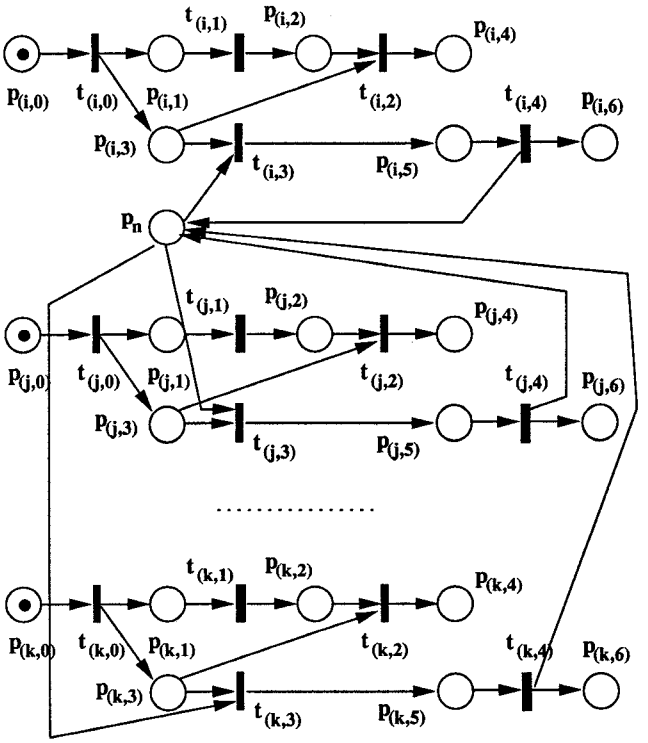

Figure 5: Windows Clients Contending for an NT Server

5 Conclusion

In this paper, the authors have conducted various experiments. For the experiments, there was no noticeable activity in processor time of the servers. But it was found that network is the most severe bottleneck which can result in entire job failure. Therefore, for a mission critical task, it is critical to solve the network problem. By upgrading the network interface card which costs only a few hundred U.S. dollars, the experiment was successful up to 200 clients, while

having one more a few ten thousand dollars still didn't solve the problem. The suggested model is not a direct one to one mapping from the lower layer protcols and server redirector behavior. But it faithfully reflect the functional aspect of the system, which is almost impossible using one to one mapping due to its large size. With this model, bottleneck problem is alleviated by pin pointing the cause of the performance degrade. The authors will conduct similar experiment for the severe dB queries problem, giving stress to the processor itself.

References

[1] M. Ajmon Marsan et al: "A class of Generalized stochastic Petri net for the performance analysis of multiprocessor systems", *ACM Tr. Comp. Sys.*, Vol. 2, No. 2, pp. 93-122, May 1984.

[2] Greg Bailey et al Ed.: *Microsoft Windows NT Resource Kit For Windows NT Workstation and Windows NT Server Version 3.51: Windows NT Resource Guide, Vol. 1*, Microsoft Press, pp. 520-521, 1995.

[3] Russ Blake : *Microsoft Windows NT Resource Kit For Windows NT Workstation and Windows NT Server Version 3.51: Optimizing Windows NT, Vol. 3*, Microsoft Press, pp. 207-249, 1995.

[4] A.A. Desrochers and R. Al-Jaar: *Applications of Petri Nets in Manufacturing Systems: Modeling, Control and Performance Analysis*, IEEE Press, Piscataway, N.J., 1995.

[5] Casey Doyle and Stuart J. Stuple Ed.: *Microsoft Windows NT server Resource Kit Version 4.0, Supplement One*, Microsoft Press, pp. 137-154, 1997.

[6] Editorial Boards: *Microsoft Windows 95 Resource Kit*, Microsoft Press, pp. 991-1031, 1995.

[7] J. Kim: *The Modeling, Analysis and Simulation of a Discrete Event Dynamic System using Time Petri Net Models*, Ph. D. dissertation, Dept. Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, New York, 1995.

[8] Mike Loukides : *System Performance Tuning*, pp. 176-205, O'Reilly & Associates, Inc., Dec. 1992.

[9] Mark Minasi : "Tuning Windows NT File Servers" *http://www.winntmag.com/issues /Dec95/tuning.htm*, 1995.

[10] Microsoft Developers' Network: "Windows NT Networking Model", *http://premium.microsoft.com/msdn/library/ winresource/dnwinnt/f1d/d1f/s7546.htm*, 1996.

[11] Microsoft Developers' Network: "File and Print Sharing Components", *http://premium.microsoft.com/msdn/library /conf/f4f/f50/d57/sa27e.htm*, 1996.

[12] Microsoft Developers' Network: "Network Resource Access", *http://premium.microsoft.com /msdn/library/winresource/ntserv/fa/dc /s308e.htm*, 1996.

[13] Microsoft Developers' Network: "The Workstation Service", *http://premium.microsoft.com /msdn/library/conf/f4f/f50/d57/sa280.htm*, 1996.

[14] Microsoft Knowledge Base: "Overview : Protocol Drivers", *http://www.microsoft.com/kb/articles /q103/8/80/htm*, Sep. 18, 1996.

[15] Microsoft Knowledge Base: "Adjust Parameters to Conect to LAN Man Over Slow Link", *http://www.microsoft.com/kb/articles/q98/8/49.htm*, 1997.

[16] T. Murata: "Petri Nets : Properties, Analysis and Applications", *Proc. IEEE*, Vol. 77, No. 4, pp. 541-580, Apr. 1989.

[17] Marc Runkel : "Ethernet Network Questions and Answers, Ver. 2.12", In *newsgroup comp.dcom.eterhnet* , Dec. 13, 1994

[18] T. Socolofsky and C Kale : "A TCP/IP Tutorial", Network Working Group RFC 1180, Jan. 1991.