Performance under Failure of Multi-tier Web Services

Bing Xie, Yong Chen, Xian-He Sun, Hui Jin
Computer Science Department in
Illinois Insititute of Technology
Chicago, IL, U.S.A
bxie3, chenyon1, sun, hjin6@iit.edu

***Abstract*—Performance issues of multi-tier Web Services have been studied extensively in recent years. Performance modeling and prediction under failure of multi-tier architectures, however, is not well addressed yet. We propose a novel model named Performance under Failure of Multi-tier Architecture, or PerFAMA in short, to address this issue. We first show that the multi-tier architecture with failure considerations is a product-form network, and then analyze and model the failure impact. By applying the PerFAMA model, we are able to predict the end-to-end response time of multi-tier Web Services under failures. We have simulated two representative Web Services architectures and various failure scenarios to verify the proposed PerFAMA model. The experimental results show that the proposed model works well and the prediction accuracy is up to 98%.**

Keywords-Performance, Failure, Multi-tier Web Services

I. INTRODUCTION

Providing reliable Web Services is a timely issue. Modeling the performance under failure of multi-tier Web Services is a necessity and the first step to build reliable Web Services. Various behaviors, such as heavy burst requests, malicious intrusion attack and network errors, will lead to a service failure sooner or later, and different failures may have different impact to the end performance. We propose a novel two-step model named Performance under Failure of Multi-tier Architecture model, or PerFAMA model in short, and its associated prediction methodologies to analyze the behavior and performance of multi-tier Web Services under failures. The PerFAMA model is capable of capturing the essential features of Web Services and depicting the performance characteristics of the global multi-tier architecture and of each service tier respectively.

The rest of this paper is organized as follows. Section 2 reviews related work in performance modeling and prediction of Web Services. Section 3 introduces the proposed PerFAMA model for multi-tier architecture under failures. The experimental results are presented in Section 4, and finally we conclude this study and discuss the future work in Section 5.

II. RELATED WORK

Many studies in modeling the performance of multi-tier applications exist [1][2][5][12][16][17]. These existing studies can be classified into two categories, performance modeling on multi-tier applications and failure analysis of a single-tier application.

A. Related work on multi-tier architecture performance modeling

Jackson introduced the product-form network concept in [6]. Multi-tier systems which are product-form network have several nice properties [15]. The request sequence satisfies birth-death process [4] is one of the properties of product-form network.

Some efforts have been done in performance analysis of multi-tier Internet applications and services [1][16]. Urgaonkar et. al. proposed to use an enhanced Mean-Value Analysis (MVA) [13] algorithm for multi-tier architecture [16]. In [1], birth-death process was used for multi-tier architecture. Some other work focused on QoS study of multi-tier applications [9][10].

These existing studies demonstrate the importance and benefits of performance modeling of multi-tier architectures. However, they do not consider the failure factor.

B. Related work on failure impact analysis

Failure impact and reliability analysis of Internet and its applications are traditional research topics [3][8][14].

Many works have been done on understanding the failure [2][5][12][17]. Our model extends the single-tier failure analysis to the multi-tier architecture and considers the distinct Web Services performance characteristics. To our best knowledge, no one has done this work before.

III. PERFORMANCE UNDER FAILURE OF MULTI-TIER ARCHITECTURES

In this section, a new queuing model is introduced for modeling the performance of a multi-tier Web Service under failures.

A. Modeling performance under failure

We first show that the multi-tier architecture under failures is a product-form network under certain reasonable assumptions. The assumptions in the PerFAMA model are:

i) Both the request arrival rate and the failure arrival rate are independent of the state of nodes and satisfy the Poisson distribution. This assumption is widely used to analyze the failure impact on performance in existing studies [2][5][12][17]. Since the service start-up time is relatively short, it is acceptable to ignore the start-up time cost.

ii) Once a failure happens, the whole system will be down. This is generally true, where the product system will stop providing services until the failure is repaired.

1521-9097/09 $26.00 © 2009 IEEE
DOI 10.1109/ICPADS.2009.84

iii) There exists a queue in each tier with an infinite size to hold all requests.

iv) All service time distributions are exponential.

We show that under these assumptions a multi-tier architecture under failures satisfies the three conditions of the product-form network. In a multi-tier architecture under failures, the system has two states: service state and down state. In the service state, the conditions of product-form networks are satisfied following the definition. According to assumption i), the request arrival rate satisfies Poisson distribution. According to assumption iii), the arrived requests will be held in queues when failure occurs. Thus, the first condition of product-form is satisfied. For the second condition, according to assumption iv), the service time satisfies exponential distribution. According to assumption ii), if one request is processed in a service node and a failure happens during this time, the processing request will be held in the queue, then be restarted after failure is repaired. Thus, the service duration time is independent on the number of requests present at that service. Therefore, the second condition is satisfied. Finally for the third condition, according to assumption ii), when the system is in the down state, no request is served, and the sequence in the system will be held and its number is consequently fixed. Thus, the sequence is independent on the state of system, and the third condition is satisfied.

With the properties of product-form network, we model the composition of Web Services using a network of N +1queues, denoted as $Q_0, Q_1, ..., Q_N$. Each queue represents one Web Service, which is delivered by one service node (marked as *0,1,…,i…,N)*. The end-users' requests arrive with rate λ_0 satisfying the Poisson distribution and wait in queue *0* till node *0* is available. Queue *i* and node *i* are combined together and considered as tier *i,* marked as T_i, ($0 \le i \le N$). After finishing the work on node 0, the requests are delivered from T_1 to T_N step by step as shown in Figure 1. After the computation of the last tier T_N , the result is sent back to T_{N-1} and is forwarded to T_{N-2} and so on until it reaches service node *0*, where it is returned to the client. The system follows the "First Come First Served" rule to process requests according to the product-form network.

Since the multi-tier architecture under failures is a product-form network, the request processing in service node *i* ($1 \le i \le N$) can be modeled as a birth-death process, where the equilibrium equation exists: $\lambda \pi_{i-1} = \mu \pi_i$, where λ is the birth rate from tier T_{i-1}, μ is the death rate of node *i*, π_i is the transient probability ($0 \le i \le N$), and λ and μ are fixed.

The mean response time of multi-tier Web Services is thus,

$$E(T) = \sum_{i=0}^{i=N} E(T_i) \tag{1}$$

where $E(T_i)$ is the time consumption of the visit to tier T_i .

B. *Performance modeling of each tier under failure*

When a request is delivered to tier T_i $0 \le i \le N$, it waits in queue i until service node i is available. Consequently, the time consumption in tier T_i consists of two parts: waiting time in queue i and service duration time in node i. We analyze the service duration time firstly.

When a request arrives at node *i*, the request will be kept in queue *i*, and a copy of the request will be forwarded to node *i* for processing. If a failure happens during its processing, the intermediate results are lost and the service node *i* will be down for failure recovery. In real practice, the service is rolled back to the last checkpoint [4][7][11] or is directly restarted after the service is recovered. Because the checkpointing cost is relatively high and is usually used in high performance computing only, we take the approach that the service is restarted after failure recovery. After the node *i* is recovered, it fetches requests from queue *i* to continue processing. Let W_i denote the service time in service node *i*. Suppose failures occur S_i times before the request is finished, and the request has already been served a period of X_i time, where $X_i < W_i$ ($1 \le i \le S_i$), before the *i*th failure takes place. Suppose the *i*th failure requires l_i time to fix. After the service is recovered from the S_ith failure, the request will be completed without failure interruption in node *i* and

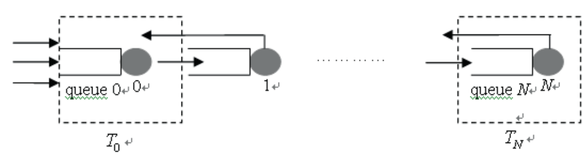


Figure 1. Modeling a multi-tier Web Service

be delivered to tier T_{i+1} . The process is shown in Figure 2.

Let t_{T_i} denote the service duration time in node *i,* which can be calculated as:

$$t_{T_i} = X_1 + l_1 + X_2 + l_2 ... + X_{S_i} + l_{S_i} + W_i \tag{2}$$

Formula (2) can be transferred as:

$$E(t_{T_i}) = W_i + \sum_{i=1}^{i=S_i} E(l_i) + \sum_{i=1}^{i=S_i} E(X_i) \tag{3}$$

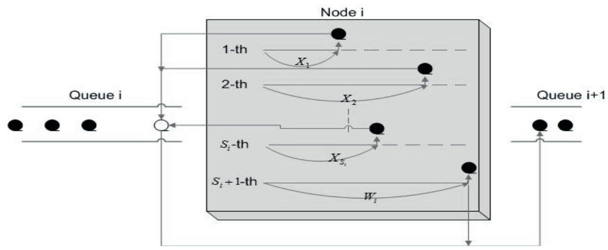


Figure 2. The serving process in node i

In addition to the service duration cost, a request has to wait in queue for a certain amount of time when the service node is busy or a failure happens. Let w_{T_i} denote the waiting time in T_i, where the service node is *i*. w_{T_i} is consisted of two parts, the waiting time during node *i*'s busy time (denoted as $w_{p,i}$) and the waiting time during failure recovery. According to assumption ii), "the waiting time during failure recovery" consists of the total waiting time for any failure in each tier (denoted as w_{S_0}, w_{S_1}, …, w_{S_N}). The probability of these two types of waiting time in tier T_i can be expressed as $\Pr(t = w_{p,i})$ and $\Pr(t = w_{S_i})$ ($0 \le i \le N$). The mean waiting time in tier T_i, thus, is:

$$E(w_{T_i}) = E(w_{p,i}) * \Pr(t = w_{p,i}) + \sum_{i=0}^{i=N} E(w_{S_i}) * \Pr(t = w_{S_i}) \quad (4)$$

Therefore, the total time cost in tier T_i is:

$$E(t_{T_i}) = W_i + \sum_{i=1}^{i=S_i} E(l_i) + \sum_{i=1}^{i=S_i} E(X_i) \quad (5)$$

C. Parameters calculation

There are two major parameters, $E(t_{T_i})$ and $E(w_{T_i})$, in the PerFAMA model. This section discusses the calculation of these two parameters respectively.

1) P $E(t_{T_i})$ calculationn

In tier T_i ($0 \le i \le N$), failure process arrives with rate $\lambda_{f,i}$, and the downtime of the failure follows a general distribution with mean $\mu_{f,i}$ and standard deviation $\sigma_{f,i}$. All these parameters are based on Poisson distribution. Notice that $\lambda_{f,i}$ is the reverse of MTBF (Mean Time Between Failure, by which the service time is interleaved), and $\mu_{f,i}$ represents MTTR (Mean Time To Recovery). From the service provider's viewpoint, failure process follows M/G/1 queuing model, where failure process is Poisson arrival, failure recovery time follows general distribution (exponential distribution belongs to general distribution), and one server serves in one tier. We can get the mean time consumption on failures in tier T_i.

$$E(l_i) = \frac{\mu_{f,i}}{1 - \lambda_{f,i} * \mu_{f,i}} \quad E(X_i) = \frac{1}{\lambda_{f,i}} - \mu_{f,i} \quad (6)$$

Taking these two results into formula (3), we obtain the duration of service time of tier T_i as

$$E(t_{T_i}) = W_i + S_i * \left(\frac{\mu_{f,i}}{1 - \lambda_{f,i}\mu_{f,i}} + \frac{1}{\lambda_{f,i}} - \mu_{f,i}\right) \quad (7)$$

2) P $E(t_{T_i})$ calculation

According to Formula (4), the values of $E(w_{p,i})$, $\Pr(t = w_{p,i})$, $E(w_{S_i})$ and $\Pr(t = w_{S_i})$ must be achieved first to calculate $E(w_{T_i})$.

In a product-form network, the waiting time for service node *i*, $E(w_{p,i})$ is given by [6].

$$E(w_{p,i}) = \frac{W_i}{1 - \lambda_i W_i} \quad (8)$$

As discussed in the previous section, λ_i is the birth rate of node *i* ($1 \le i \le N$). When *i*=0, $\lambda_0 = \lambda$.

Then we calculate the probability for waiting time for service ($\Pr(w_{p,i})$). With equilibrium equation, the probability of the delay for waiting service availability is given in [4] as:

$$\Pr(t = w_{p,i}) = \frac{\prod_{m=0}^{i-1} \frac{\lambda_m}{\mu_{m+1}}}{1 + \sum_{i=1}^{N} \prod_{m=0}^{i-1} \frac{\lambda_m}{\mu_{m+1}}}$$

where λ_m is the birth rate for service node *m*, μ_{m+1} is the death rate from node *m*+1, ($0 \le m \le i-1$). In a real system, it is difficult to measure all λ_m and μ_m for each tier. In general, the average value of λ and μ are used to simplify the calculation [7][11], where, $\lambda = \lambda_0 = \ldots = \lambda_i = .. = \lambda_N$ and μ_m is the reverse of the processing availability [1] of node *i*

($\frac{1}{W_i}$). Hence, $\Pr(t = w_{p,i})$ can be calculated as:

$$\Pr(t = w_{p,i}) = \frac{\prod_{m=0}^{i-1} \frac{\lambda}{W_{m+1}}}{1 + \sum_{i=1}^{N} \prod_{m=0}^{i-1} \frac{\lambda}{W_{m+1}}} \quad 9$$

$E(w_{s_i})$ is the MTTR (Mean Time To Recovery) of node i

$$E(w_{s_i}) = \mu_{f,i} \quad (10)$$

For $\Pr(t = w_{S_i})$ ($0 \le i \le N$), according to Poisson distribution, we have:

$$\Pr(t = w_{S_i}) = \frac{\lambda_{f,i}^{S_i} * e^{-\lambda_{f,i}}}{S_i!} \quad (11)$$

Combining Formula (7), (8), (9), (10) and (11), the end-to-end response time of a multi-tier architecture under failures can be decided. To make the whole calculating process clear, an algorithm is given in Figure 3.

Objective: calculating the mean end-to-end response time
Input:
// Input a set of service time of each tier
$W = \{W_0, W_1, W_2, ..., W_N\}$
// Input a set of failure arrival rate and mean in each tier
$\lambda_f = \{\lambda_{f,0}, \lambda_{f,1}, \lambda_{f,2}, ..., \lambda_{f,N}\}$
$\mu_f = \{\mu_{f,0}, \mu_{f,1}, \mu_{f,2}, ..., \mu_{f,N}\}$

λ // Input the request arrival rate
$N+1$ // Input the tier number
T_D ($W_i << T_D$) //Input the detection time
Output: $E(T)$
Begin
//Calculating the failure arrival times in each tier (S_i).
For (i=0,i≤N, i++)
$S_i = \lambda_i T_D$;
//Calculating the time consumption on tier T_i ($E(T_i)$)
For (i=0,i≤N,i++)

$$E(T_i) = \frac{\prod_{m=0}^{i-1} \frac{\lambda}{W_{m+1}}}{1 + \sum_{i=1}^{N} \prod_{m=0}^{i-1} \frac{\lambda}{W_{m+1}}} * \frac{W_i}{1 - \lambda W_i} + \sum_{i=0}^{N} \frac{\lambda_{f,i}^{S_i} * e^{-\lambda_{f,i}}}{S_i!} + W_i + S_i * \frac{\mu_{f,i}}{1 - \lambda_{f,i}} + \frac{S_i}{\lambda_{f,i}}$$

//Calculating the end to end response time ($E(T)$)

$$E(T) = \sum_{i=0}^{N} E(T_i)$$

End

Figure 3. Algorithm for calculating mean response

IV. MODEL VALIDATION

A. *Experimental facilities*

We have utilized the Colored Petri Nets (CPN) language and toolkit for the simulation experiments. Colored Petri Nets [18] is a graphical oriented language for design, specification, simulation and verification of systems. CPN tool [19] is widely used for editing, simulating and analyzing Colored Petri Nets.

B. *Model accuracy validation*

1) *Experimental setup*

We have built two representative multi-tier Web Services architectures, one three-tier and one five-tier, as shown in Figure 4 and Figure 5 respectively. We discuss the setup of the five-tier Web Service only since the setup of the three-tier case is similar.

Figure 5 shows the system page of the five-tier Web Service architecture and demonstrates how the simulation works. As shown in Figure 5, each service node has one associated queue to hold requests when the service node is busy or out-of-service due to a failure.

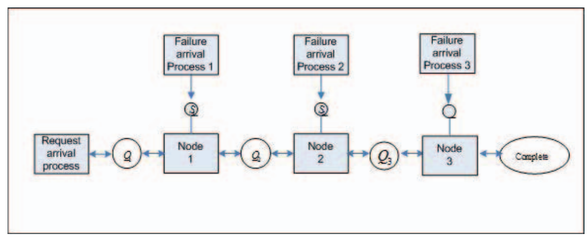


Figure 4. The system page of a three-tier Web Service architecture under failure

When a new request arrives, it will be enqueued in queue 0 first. If the service node 0 is available, the request will be delivered to node 0 immediately. If node 0 is busy or down to failure, the request will stay in the queue. After the request is processed by node 0, the request will be forwarded. After the failure is fixed and the service is restarted, the node will fetch requests from the previous queue. When a request is proceeded by the last tier successfully, the whole process for one request is completed.

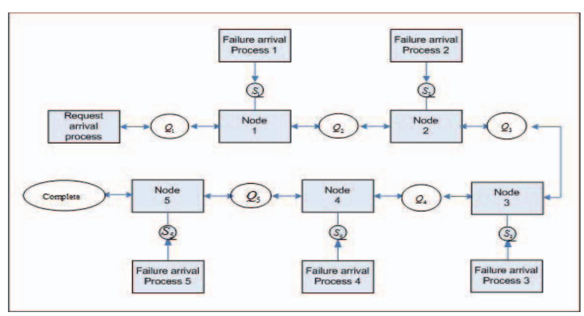


Figure 5. The system page of a five-tier Web Service architecture under failure

Failure are simulated with a separate process S_i. The mean stall time is $\mu_{f,i}$. Both request and failure arrival processes are generated as Poisson processes but with different parameters.

2) Experimental results

We have performed three series of experiments to verify the accuracy of the PerFAMA model. The first series of experiments aimed to study the accuracy of the PerFAMA model when the request arrival rate varied. The failure arrival rate and the failure recovery time were fixed as 0.01 per second and 10 seconds. The service time were set as [7, 9, 11] and [3, 5, 10, 7, 20] (unit: second) for the three-tier architecture and five-tier architecture, respectively. Figure 6 shows the results. The accuracy of the PerFAMA is up to 98% and the average accuracy is 94.3%.

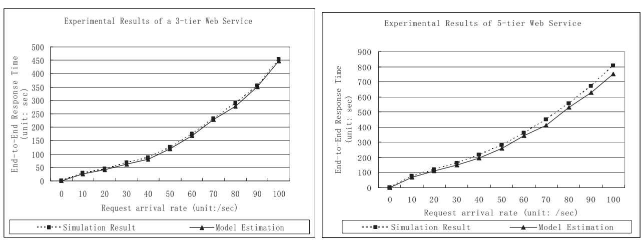


Figure 6. End-to-end response time on three-tier and five-tier web services when request arrival rate varies

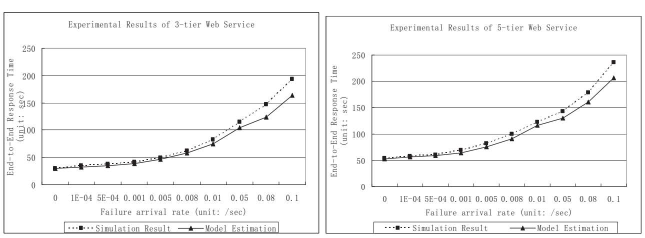


Figure 7. End-to-end response time of three-tier and five-tier web services when failure arrival rate varies

The second set of experiments was conducted to compare the prediction results of the model and the simulation results when the failure arrival rate varied. We fixed the request arrival rate as 30 per second, failure recovery time as 10 seconds and the service time as [7, 9, 11] and [3, 5, 10, 7, 20] (unit: second) on the three-tier and five-tier Web Service separately. We limited the ratio of the failure arrival rate and request arrival rate is less than 1:200. The results are shown in Figure 7. The prediction accuracy in this set of experiments is up to 97% and the average accuracy is 93.9%.

The third set of experiments was similar to the previous two sets of experiments, except that the failure recovery time was varied, while the request arrival rate was fixed as 30 per second, the failure arrival rate as 0.05 per second. The ratio between the failure recovery time and the system working time is less than 1:400. The results, as shown in Figure 8, verified that the PerFAMA model can predict the response time with up to 95% accuracy. The average accuracy is 91.4%.

These three sets of experiments have verified that the proposed PerFAMA model works well under various failure scenarios. The next section discusses the tier sensitivity and bottleneck analysis.

C. Tier sensitivity and bottleneck analysis

The service time of these two sets of experiments was configured as [3, 5, 10, 7, 20] (unit: second) for workload set A and [20, 5, 7, 10, 3] (unit: second) for workload set B separately. This configuration represents the vastly different service time of Web Services and variant tier sensitive workloads.

1) Tier sensitivity analysis

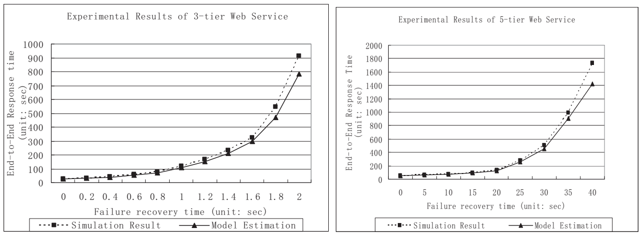
We varied the failure arrival rate on each tier for both workload set A and B to study the tier sensitivity. The results are shown in Figure 9.

The experimental results demonstrated that the failure impact on end-to-end response time is tier sensitive. The longer of the service time, the higher of the failure impact. The slope of each line in the figure represents the increasing speed of the end-to-end response time, and the steeper of the line, the greater of the increasing speed. This matches the PerFAMA model well since the frequency of the failure arrival rate directly affects the number of failures, S_i. The greater of S_i is, the bigger of $E(T_{Q_i})$ and $\Pr(t = w_{S_i})$ are. Consequently, $E(Q)$ is greater.

2) Bottleneck analysis

We conducted another set of experiments to confirm the capability of the PerFAMA model to detect potential bottlenecks by taking two examples, workload set A and B. We fixed the failure arrival rate as 0.001 per second and the recovery time of each failure as 10 seconds, but varied the request arrival rate. In order to find out the bottleneck series, we increased the capability of the corresponding node after a bottleneck is found to bigger enough to hide its effect and find out the next potential bottleneck. The results are shown in Figure 10.

The experimental results illustrated that the service time played an important role. The longer of the service time, the more likely the corresponding node becomes the bottleneck.



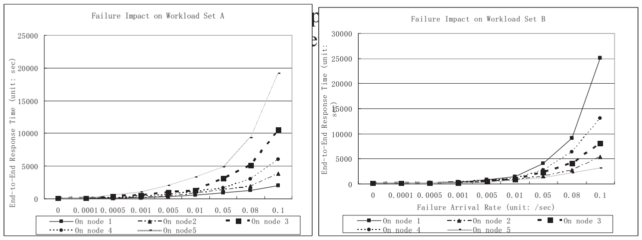


Figure 9. Failure impact on
end-to-end response time of tier-sensitive workloads

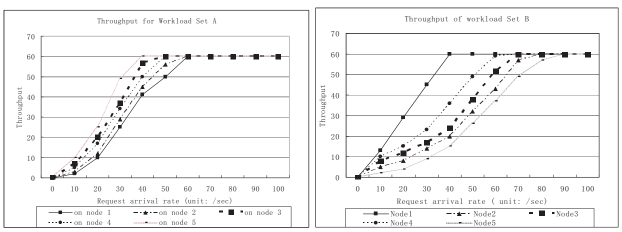


Figure 10. Throughput of workload set A and B

The order of the bottleneck also depends on the length of the service time of each tier. The later position of the node, the larger weight it has in deciding the distance. For instance, the distance between the first bottleneck node, node 5, and the second bottleneck node, node 3, is obviously smaller than the distance between node1 and node 4.

V. CONCLUSION AND FUTURE WORK

As Web Services tending to large-scale size and deeper hierarchy architecture, a failure is likely to occur when serving an end-user's request. The proposed model adopts a network of queues to represent how all tiers in a multi-tier Web Service cooperate with each other and process requests. We have shown that this queue network is a product-form network and analyzed the end-to-end response time of multitier Web Services considering failure impact. The experimental results show that the proposed model captures the performance under failure of Web Services well.

REFERENCES

[1] S. Bhulai, S. Sivasubramanian, R. van der Mei, M. Van Steen, "Modeling and Predicting End-to-End Response Times in Multi-Tier Internet Applications", *Managing Traffic Performance in Converged Network*s, ISSN, 0302-9743, pp. 519-532.

[2] A. Duda, "The Effects of Checkpointing on Program Execution Time", On Information Processing Letters, June 1983, vol.16, pp. 221-229.

[3] S. Floyd, V. Paxson, "Difficulties in Simulating the Internet", IEEE/ACM Transactions on Networking. Feb. 2001, Vol.9, pp. 392-403.

[4] P. J. Fortier, H. E. Michel, "Computer Systems Performance Evaluation and Prediction", 2003, Chapter7- Queuing Theory,

[5] S. Garg, Y. Huang, C. Kintala, K.S.Trivedi, "Minizing Completion Time of a Program by Checkpointing and Rejuvenation", In Proc. Of 1996 ACM SIGMETRICS Conference, Philadelphia, PA, May 1996, pp. 252-261.

[6] J. Jackson, "Networks of waiting lines", Operations Research 5(1957), pp. 518-521.

[7] D. J. Lilja, "Measuring Computer Performance: A Practitioner's Guide", C*ambridge University Press*, 2000, Chapter11.

[8] M. May, J. Bolot, A. Jean-Marie, C. Diot, "Simple Performance Models of Differentiated Services Schemes for the Internet". On *INFOCOM'99*, pp.1385-1394.

[9] R. D. van der Mei, H. B. Meeuwissen, "Modeling End-to-End Quality-of-Service for Transaction-Based Services in a Multi-Domain Environment", In *Proceedings IEEE International Conference on Web Services* (*ICWS*), Chicago, 2006, USA.

[10] W.Q. Lin, Z. Liu, C.H. Xia, L. Zhang, "Cost Minimization of Multi-Tiered e-Business Infrastructure with End-to-End Delay Guarantees". *ACM SIGMETRICS Performance Evaluation Review*, 2004, pp. 25-27.

[11] P. V. Mieghem, "Performance Analysis of Communications Networks and Systems", *Cambridge University Press*, Chapter 11.3.

[12] V. F. Nicola, V. G. Kulkarni, K. S.Trivedi, "Queueing Analysis of Fault-Tolerant Computer Systems", On *IEEE Trans. Software Engineering*, 1987, Vol. SE-13, No.3, pp.363-37.

[13] M.Reiser, S. S. Lavenberg, "Mean-Value Analysis of Closed Multichain Queuing Networks", In *Journal of the Association for Computing Machinery*, volume 27, 1980, pp. 313-322.

[14] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger. M. F. Kaashoek, F. Dabek, H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications", *IEEE/ACM Transactions on Networking*, 2003, Vol. 11, No.1 pp. 17-32.

[15] B. Pittel, "Closed Exponential Networks of Queues With Saturation: The Jackson-Type Stationary Distribution and Its Asymptotic Analysis", In *Mathematics of Operations Research*, pp. 357-378.

[16] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, A. Tantawi, "An Analytical Model for Multi-tier Internet Services and Its Applications", *In: Proc. Of the ACM SIGMETRICS conference,* 2005, pp. 291-302.

[17] M. Wu, X.-H. Sun, H.Jin "Performance under Failure of High-End Computing", on *Proc. Of the ACM/IEEE SuperComputing Conf. 2007(SC'07)*, Nov. 2007.

[18] http://www.daimi.au.dk/CPnets/intro/

[19] http://wiki.daimi.au.dk/cpntools/cpntools.wiki