This is the published version:


Available from Deakin Research Online:

http://hdl.handle.net/10536/DRO/DU:30018138

©2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Copyright: 2008, IEEE
A Transformation Model for Heterogeneous Servers

Shui Yu, Robin Doss, Theerasak Thapngam  

School of Engineering and Information Technology, Deakin University,  
Burwood, VIC 3125, Australia  
{syu, rchell, tthap}@deakin.edu.au

David Qian  
Faculty of Higher Education Lilydale  
Swinburne University of Technology,  
Lilydale, VIC 3140, Australia  
dqian@swin.edu.au

Abstract

One of the characteristics of the current web services is that many clients request the same or similar service from a group of replicated servers, e.g., music or movie downloading in peer-to-peer networks. Most of the time, servers are heterogeneous ones in terms of service rate. Much of research has been done in the homogeneous environment. However, there has been little done on the heterogeneous scenario. It is important and urgent that we have models for heterogeneous server groups for the current Internet applications design and analysis. In this paper, we deploy an approximation method to transform heterogeneous systems into a group of homogeneous system. As a result, the previous results of homogeneous studies can be applied in heterogeneous cases. In order to test the approximation ratio of the proposed model to real applications, we conducted simulations to obtain the degree of similarity. We use two common strategies: random selection algorithm and First-Come-First-Serve (FCFS) algorithm to test the approximation ratio of the proposed model. The simulations indicate that the approximation model works well.

1. Introduction

Internet based applications are developing dramatically, and peer-to-peer applications are becoming the main platform for applications. One characteristic of the current Internet applications is that many clients request the same service from a group of replicated servers such as music and movies downloading, and news accessing. In order to improve the quality of service, distributed systems are established to alleviate the workload of the requests. An excellent architecture is the peer-to-peer network, such as Napster, Gnutella for music downloading, Skype for voice delivery over IP, and recently, the television over IP [14][8].

There are a lot of homogeneous analyses in parallel systems [13][12]. Homogeneous analysis for data distribution and search in peer-to-peer networks has been well done [6][15][7]. Under the condition of homogeneous environment, researchers have developed some fundamental conclusions in peer-to-peer networks.

Queuing theory has been applied to computer science for many years, however, most of the analyses have been based on homogeneous assumption. For example, since the proposed idea of parallel downloading, a number of papers focused on this area using the queuing theory [13][12][4], and all of them are conducted on the assumption of homogeneous environment. Game theory is also applied in this area [18], and it uses the homogeneous assumption as well.

However, the homogeneity of servers is not true in real situations, in fact most of the Internet based systems are heterogeneous, e.g. peer-to-peer systems. In order to depict an Internet based system, we have to explore its property of heterogeneity. Some works have been done in job scheduling in heterogeneous environments [5][3][11], Performance with block probability in heterogeneous environment has been explored using the queuing theory as well [2][1]. The real time heterogeneous traffic flow on the Internet has also been explored using the network calculus focusing on heterogeneous traffic control [9][10]. However, all these studies are not capable enough as a mathematical tool for the design, analysis for heterogeneous systems.

It is obvious and important for us to develop applicable models for heterogeneous systems. In this paper, we try to achieve this goal which aims to establish a mathematical model for the servers. We focus on a group of servers who offer the same service with different service rates. Therefore, we will be able
to conduct further works such as performance evaluations, system designs, and so on.

Our methodology is simple. We divide the heterogeneous servers into groups according to their service rates. We treat the servers in the same group as homogeneous servers. By this method, we transform a heterogeneous system into a group of homogeneous systems. Therefore, the previous conclusions on homogeneous systems can be applied. Our simulations demonstrated that the approximation model works well in random selection algorithms, and it works also well in the FCFS algorithm with conditions.

The rest of this paper is organized as follows: Section 2 introduces the related work for homogeneous and heterogeneous analysis and Section 3 proposes our modeling on heterogeneous servers. Two common algorithms are presented to evaluate the performance based on the proposed model. The simulations are conducted in Section 4 to demonstrate the effectiveness of the model through the two algorithms. Finally, Section 5 concludes the paper and points out future work.

2. Related work

Reference [19] has been researched on gathering servers into a service pool in a homogeneous environment to serve all the incoming requests. The aim was to reduce the reject rate for the clients. The paper analyzed the homogeneous cases using queuing theory. Furthermore, the paper touched the heterogeneous cases. However, no theoretical analysis was presented.

[13] proposed the paracasting model, an application layer anycasting protocol, for content parallel downloading. The queuing analysis was based on a set of N homogeneous servers. The paper progressed on two assumptions: the Markovian service rate for a request when there were finite servers and infinite servers with general service rate. Based on the homogeneous analysis, the authors found that paracasting is effective when a small number of servers, for example, up to four, are applied to serve a request concurrently.

In paper [17], the authors originally proposed a parallel accessing method to download popular content via the Internet. In contrast to downloading a file from a single server, the proposed strategy tried to access the desired file from different servers and reassemble them locally. Moreover, a file is divided into parts, and the faster servers deliver bigger portions while slower servers do smaller portions. The scheme aims on shortening the user’s experience and making the response time consistent. However, the paper does not give a theoretical analysis, even in the homogeneous cases. [4] proposed a multi-server, multi-installment solution approach to the delivery problem of continuous-media files over unreliable communication links. The proposed method can achieve a minimization of the client’s waiting time. Paper [12] pointed out that the previous research of parallel downloading focused on the performance of a given client, rather than the system of all clients using parallel downloading. They analyzed the performance of the two schemes, single downloading and parallel downloading using a birth-death process. The paper concluded that with proper admission control and dimensioning, single-server downloading can perform just as well as parallel downloading, without the complexity and overhead incurred by parallel downloading. Reference [18] modeled the parallel downloading scheme using Nash equilibrium. The authors obtained the conclusion that although the traffic configuration at Nash equilibrium is optimal from the viewpoint of the individual clients, it may be bad in terms of a whole system. All these researches have been done under the homogeneous assumption.

3. Modeling of heterogeneous servers group

In this section, we propose our approximation method to model the heterogeneous servers groups. Based on the proposed model, we analyzed the performance of two common scheduling algorithms, random algorithm and First-Come-First-Serve (FCFS) algorithm, which can reflect the approximation ratio of the proposed model through the comparison with the simulation results, which will be presented in the next section.

![Figure 1. Sample system of heterogeneous servers](image)

**Figure 1. Sample system of heterogeneous servers**

Our objective system is shown as Figure 1, a group of n heterogeneous servers, $S_i (i = 1, 2, ..., n)$, which
serve many requests from the Internet, where the $\mu_i$ is the service rate for server $S_i$. As a heterogeneous system, each server may have different service rates ($\mu_i \neq \mu_j$ where $i \neq j$ and $i, j \in \{1, 2, ..., n\}$) with high probability. We have a scheduling algorithm which is running on the system to decide which server to serve the incoming requests.

It is hard to find a mathematical expression for this kind of heterogeneous systems. Even though we can express the system in mathematics, for example, in terms of random processing, and stochastic processing, we can not apply the conclusions that researchers have achieved in the homogeneous scenarios. In this paper, we try to transform the mentioned heterogeneous system into a homogeneous system using approximation methods.

We make the following assumptions to make the discussion clear and simple in this paper:

- We assume there is no network delay for each request;
- The bandwidth for any server is infinite;
- The whole system is a M/M/1 system with request arrival rate $\lambda$, and system service rate $\mu$.

Let

$$\mu_{\text{min}} = \min(\mu_1, \mu_2, ..., \mu_n) = \mu$$
$$\mu_{\text{max}} = \max(\mu_1, \mu_2, ..., \mu_n)$$
$$k = \left\lfloor \frac{\mu_{\text{min}}}{\mu_{\text{max}}} \right\rfloor (1)$$

We assume $\frac{\mu_{\text{max}}}{\mu_{\text{min}}} \geq 2$, namely $k \geq 2$, otherwise, we can treat the system as a homogeneous system, which has been researched a lot in previous work.

Let $p(s_i)$ be the probability of server $S_i$ be accessed, then the average system service rate is

$$\mu_s = \sum_{i=1}^{n} p(s_i) \cdot \mu_i (2)$$

We divide the $n$ servers into $m$ levels based on their service rate. Each level consists of $z_m (m = 1, 2, ..., k)$ servers. We use $m \mu$ to represents the service rate of every server at level $m$. Therefore, we can transform the heterogeneous server’s system into a nearly homogeneous server’s system, as shown in Figure 2. After the transformation, we therefore can apply the conclusions that researchers have achieved in the queuing theory to analyze the heterogeneous systems. We will examine the effectiveness of the approximation in the next section.

![Figure 2. Simplified model for heterogeneous servers](image)

The arrival rate for the whole system is $\lambda$, and we use $\lambda_s$ to represent the actual arrival rate for one specific server in terms of statistic average, we then have

$$\lambda_s = \frac{1}{n} \cdot \lambda (3)$$

From [16], we have the following results for an M/M/1 system. The system utilization factor is as follow,

$$\rho = \frac{\lambda_s}{\mu_s} (4)$$

For any situation, we need $0 \leq \rho \leq 1$, namely $\lambda_s \leq \mu_s$. The average time that one request stayed in the system can be expressed as follow,

$$T_q = \frac{1}{\lambda_s} \cdot \rho = \frac{1}{\mu_s - \lambda_s} (5)$$

The average waiting time for one request is as follow,

$$T_w = \frac{\rho}{\mu_s \cdot (1 - \rho)} = \frac{1}{\mu_s} \cdot \frac{\lambda_s}{\mu_s - \lambda_s} = \frac{\lambda_s}{\mu_s} \cdot T_q (6)$$

In order to evaluate the approximation ratio of the proposed model with real systems, we take advantage of two common algorithms, random algorithm, and First-Come-First-Serve (FCFS) algorithm to test our approximations. The two algorithms are described as follows,

- Random algorithm: We take one server randomly to serve the incoming request. In
the other words, every server in the system could be chosen to serve the incoming request with equal probability.

- FCFS algorithm: In this scheduling algorithm, each server serves one request at a time when the job is done, the server returns back to the available queue. The scheduler always takes the first server in the available queue to serve the incoming request.

In the following discussion in this paper, we use the footnote \( r \) to represent the random algorithm, and \( f \) to represent the FCFS algorithm. We will infer \( T'_q \) and \( T'_w \) of each algorithm based on our approximation model in the rest of this section, which will be compared with the simulations results, respectively, in the next section.

### 3.1 Random algorithm performance analysis on the proposed model

For the random algorithm, the probability for one server be chosen to serve is given by

\[
P_r(S_i) = \frac{1}{n} = \frac{1}{\sum_{m=1}^{k} z_m}, \quad i = 1, 2, \ldots, n
\]  

(7)

Hence, the system average service rate as a super server is given by

\[
\mu_{sr} = \sum_{i=1}^{n} P_r(S_i) \cdot \mu_i = \frac{1}{n} \cdot \sum_{m=1}^{k} \left[ (m \mu) \cdot z_m \right]
\]  

(8)

\[
\rho_r = \frac{\lambda_s}{\mu_{sr}} = \frac{\frac{1}{n} \cdot \sum_{m=1}^{k} \left[ (m \mu) \cdot z_m \right]}{\mu \cdot \sum_{m=1}^{k} \left[ m \cdot z_m \right]}
\]  

(9)

Combining equation (9) with equation (5) and (6), we obtain

\[
T'_q = \frac{1}{\frac{1}{n} \sum_{m=1}^{k} \left[ (m \mu) \cdot z_m \right] - \lambda_s} = \frac{n}{\mu \sum_{m=1}^{k} \left[ m \cdot z_m \right] - n \lambda_s}
\]  

(10)

\[
T'_w = \frac{\lambda_s}{\mu_{sr}} \cdot T'_q
\]

(11)

In the special case, when \( z_m = C \) and \( m = 1, 2, \ldots, k \), we use \( \mu_{sr} \) and \( T'_q \) to represent the super server’s service rate, the average response time and average waiting time, respectively, hence, we obtain

\[
\mu_{sr} = \sum_{i=1}^{n} P_r(S_i) \cdot \mu_i = \frac{1}{2} \mu (k + 1)
\]  

(12)

\[
T'_q = \frac{n}{\mu \sum_{m=1}^{k} \left[ m \cdot z_m \right] - n \lambda_s} = \frac{1}{\frac{1}{2} \mu (k + 1) - \lambda_s}
\]  

(13)

\[
T'_w = \frac{\lambda_s}{\mu_{sr}} \cdot T'_q = \frac{\lambda_s}{\frac{1}{2} \mu (k + 1) - \lambda_s}
\]

(14)

### 3.2 FCSC Algorithm performance analysis on the proposed model

For the FCFS algorithm, the probability for one server be chosen to serve is given by

\[
P_f(S_i) = \frac{\mu_i}{\sum_{m=1}^{k} \sum_{j=1}^{z_m} m \mu}, \quad i = 1, 2, \ldots, n
\]

(15)

Hence, the system service rate as a super server is given by

\[
\mu_{sf} = \sum_{i=1}^{n} P_f(S_i) \cdot \mu_i = \sum_{m=1}^{k} \sum_{j=1}^{z_m} \left[ (m \mu)^2 \right] \sum_{m=1}^{k} \sum_{j=1}^{z_m} m \mu
\]

(16)

\[
\rho_f = \frac{\lambda_s}{\mu_{sf}} = \frac{\lambda_s}{\frac{\sum_{m=1}^{k} \sum_{j=1}^{z_m} (m \mu)^2}{\sum_{m=1}^{k} \sum_{j=1}^{z_m} m \mu}}
\]

(17)

Combining equation (17) with equation (5) and (6), we obtain
In the special case, when \( z_m = C \) and \( m = 1, 2, ..., k \), we use \( \mu_s' \), \( T_{qf}' \) and \( T_{wf}' \) to represent the super server’s service rate, the average response time and average waiting time, respectively, hence, we obtain

\[
\mu_s' = \sum_{i=1}^{k} P_f(S_i) \cdot \mu_i = \frac{1}{3} \mu(2k + 1)
\]

\[
T_{qf}' = \frac{1}{\mu_s' - \lambda_s} = \frac{1}{\frac{1}{3} \mu(2k + 1) - \lambda_s}
\]

\[
T_{wf}' = \frac{\lambda_s}{\mu_s'} \cdot T_{qf}' = \frac{\lambda_s}{\frac{1}{3} \mu(2k + 1)[\frac{1}{3} \mu(2k + 1) - \lambda_s]}
\]

4. Performance evaluations

In order to evaluate how close the proposed model is to the real applications, we conducted some simulations to examine it. We have 10 servers with different service rates in the simulation environment. We category the servers with similar service rate into one group, and use variable \( k \) to denote the number of groups, hence the variable \( k \) denotes the degree of heterogeneity of the servers. In our results, \( k \) represents the simulation results, whereas \( k' \) represents the theoretical results obtained by our proposed model.

We first investigated the approximation ratio of the proposed model under the random algorithm with respect to \( k \). We examined the service time against the arrival intervals and requests intervals, respectively. The results are shown in Figure 3 and Figure 4.

Figure 3 indicates that our proposed model works quite well to the applications simulations for all the heterogeneous service rates distributions.

Figure 4 shows that there is a fixed difference between our theoretical results and the simulation results when \( k \) is large. However, we can adjust the difference by adjusting the theoretical result by adding a constant.

We also investigated the approximation ratio of the proposed model under the FCFS algorithm with respect to \( k \). We examined the service time against the arrival intervals and requests intervals, respectively. The results are shown in Figure 5 and Figure 6.
5. Summary and future work

We noticed the lack of a mathematical model for heterogeneous servers groups in the Internet applications. We used an approximation method to transform heterogeneous systems into a number of homogeneous groups. Therefore, the previous conclusions about homogeneous analysis can be applied. We divided the heterogeneous servers into groups according to their service rate, and we treated the servers in the same group as homogeneous servers. The method is simple, however, it is quite important to represent the heterogeneous service rates in server groups as it offers us a foundation to analyze complicated scenarios. The simulations demonstrated that the approximation works well in some conditions, especially for the random selection algorithm, which is quite popular in unstructured peer-to-peer networks.

We have a high interest to investigate in this research further. For example, there is a need to refine the model by reducing the granularity. We are confident that the approximation will be improved once the granularity is decreased.

References


