KICS
The Korean Institute of
Communications and
Information Sciences

Available online at www.sciencedirect.com
ScienceDirect

ICT Express ▮ (▮▮▮▮) ▮▮▮–▮▮▮

www.elsevier.com/locate/icte

Cloud computing with single server threshold and double congestion thresholds

Shun-Ping Chung, Yu-Ju Lu, Yu-Chen Lai*

Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

Received 4 January 2017; received in revised form 6 March 2017; accepted 13 March 2017
Available online xxxx

Abstract

In this work, we study how to minimize the average system delay in a cloud computing center with heterogeneous servers, where each server may have a different average service rate. We consider $M/M_i/C/K$ with a single server threshold and double congestion thresholds. The analytical models and performance measures are derived for the systems considered. The effect of the average arrival rate on performance measures is studied. It is shown that $M/M_i/C/K$ with a single server threshold and double congestion thresholds outperforms $M/M_i/C/K$ in terms of the average system delay. Finally, a computer simulation is written to verify the accuracy of the analytical results.

Keywords: Average arrival rate; Average system delay; Cloud computing center; Heterogeneous server

1. Introduction

The requirements for cloud computing are increasing, owing to its high efficiency in time and energy, flexibility in data access, simplicity and ease of use, and ease of maintenance [1]. Obviously, when demand increases, the number of servers in a cloud computing center must increase. The computing power of a new server may be very different from that of the existing servers. That is, cloud computing centers may have heterogeneous servers, where the computing power of each server may be different, and thus the service time of one job at different servers may be different [2]–[3].

In [4], in contrast to a traditional M/M/C/K system, where the average service rate of each server is the same, a cloud computing center is modeled as $M/M_i/C/K$, where the average service rate of each server may be different. This is characterized as follows: (1) if there is more than one idle server, the job will enter the server with the fastest service rate; (2) the job in service with the shortest service time among the users in service will leave the system first; (3) once there is an idle server, that server will take one job in the queue and start the service, as long as the queue is not empty; and (4) once service is provided by a server, the job will stay at that server until the service is finished.

The drawback of $M/M_i/C/K$ is that if servers with a slower service rate are used, it may take longer to finish the service. Instead of entering one idle server with a slower service rate, the job waiting in the queue may spend less time in the system if it chooses to wait for a busy server with a faster service rate to become idle. In Lin and Kumar [5], the authors proposed the adoption of a threshold-based scheme to improve system performance for $M/M_i/C/K$ with two heterogeneous servers, i.e., one fast server and one slow server. Specifically, when becoming idle, the fast server is always used if there is at least one user in the queue, whereas the slow server is used only when the number of users in the queue is above one threshold.

The authors in H.P. Luh and I. Viniotis [6] determined a policy that minimizes the average number of customers in the

* Corresponding author.
E-mail addresses: spchung@mail.ntust.edu.tw (S.-P. Chung), m10307603@mail.ntust.edu.tw (Y. Lu), m10407613@mail.ntust.edu.tw (Y. Lai).

Peer review under responsibility of The Korean Institute of Communications Information Sciences.

http://dx.doi.org/10.1016/j.icte.2017.03.002

system for $M/M_i/C/K$ with more than two heterogeneous servers via Linear Programming (LP). In [7], the authors proposed the adoption of an improved Random Early Discard (IRED) algorithm for queue management. Specifically, the new arrival packet is dropped based on two congestion thresholds; minimum and maximum. No packet will be dropped when the number of packets in the queue is smaller than the minimum threshold. If the number of packets in the queue is greater than or equal to the maximum threshold, the packets will be dropped with a certain probability. When the number of packets in the queue is between the minimum and maximum thresholds, the packets will be dropped with a probability that is a function of the queue occupancy. We apply the server threshold and/or congestion threshold to improve the performance of cloud computing with heterogeneous servers.

The rest of this work is organized as follows. In Section 2, we describe the system models and consider them in detail. In Section 3, the analytical models of the considered systems are derived. In Section 4, the analytical and simulation results are presented and discussed. Finally, we make a brief conclusion in Section 5.

2. System model

We consider a cloud computing center with C heterogeneous servers, where each of the heterogeneous servers may have a different average service rate. The service time of any job at each server is exponentially distributed. The average service rate of the ith server is μ_i, $i = 1, 2, \ldots, C$. We consider cases with $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_C$, i.e., the first server has the fastest service rate, the second server has the second fastest service rate, and the Cth server has the slowest service rate. The job arrivals follow a Poisson process with average arrival rate λ. The queue size is assumed to be finite and can accommodate at most N jobs. The performance measures of interest are the average number in the system (L), average queue length (L_q), loss probability (P_L), throughput (TH), average system delay (W), and average queueing delay (W_q).

Specifically, we study $M/M_i/C/K$ with a single server threshold and double congestion thresholds, described as follows. C servers are divided into two classes: class-1 and class-2. The number of class-1 servers is denoted by C_1, and the number of class-2 servers is denoted by $C_2 = C - C_1$. The class-2 server has a single server threshold t and two congestion thresholds: minimum (th_{min}) and maximum (th_{max}), where $t \leq th_{min} \leq th_{max}$.

For $M/M_i/C/K$ with server threshold t and two congestion thresholds, when becoming idle, the class-1 server is engaged as long as there is one job waiting in queue, whereas the class-2 server is not used until the number of jobs in the queue is greater than or equal to the threshold t. Furthermore, no job will be dropped when the number of jobs in the queue is smaller than the minimum threshold (th_{min}). If the number of jobs in the queue is greater than or equal to the maximum threshold (th_{max}), the jobs will be dropped with probability max_p.

When the number of jobs in the queue is between the minimum and maximum thresholds, the jobs will be dropped with a probability that increases nonlinearly as a function $f(x)$ of the number of jobs in queue x, where $f(x)$ is given as follows:

$$f\,(x) = \begin{cases} 0, & 0 \leq x < th_{min} \\ \dfrac{(x - th_{min})^2}{(th_{max} - th_{min})^2} max_p, & th_{min} \leq \mathrm{x} < th_{max} \\ max_p, & th_{max} \leq x < N. \end{cases}$$

3. Analytical model

In this section we consider $M/M_i/C/K$ with a single server threshold and double congestion thresholds. The associated state balance equations and performance measures are as follows.

3.1. State balance equations

$M/M_i/C/K$ with a single server threshold and double congestion thresholds can be described as a (C+ 1)-dimensional Markov chain with state $(i, j_1, j_2, \ldots, j_C)$, where i represents the number of jobs in the queue, j_k represents whether the kth server is busy [$j_k = 1(0)$ if busy (idle)]. The state space is represented as follows:

$$S_1 = \{(i, j_1, j, \ldots, j_C) | 0 \leq i \leq N,\ j_k = 0\, or\, 1, \\ k = 1, 2, \ldots, C\}.$$

To simplify the notation, $M/M_i/C/K$ with a single server threshold and double congestion thresholds is reduced to a two-dimensional Markov chain with state (i, j), where i represents the number of jobs in the queue and $j = \sum_{k=1}^{k=C} j_k 2^{k-1}$. The state space is represented as follows:

$$S_2 = \{(i, j) | 0 \leq i \leq N,\ 0 \leq j \leq 2^C - 1\}.$$

The steady state probability of state (i, j) is denoted as $\pi_{i,j}$, and the balance equations are divided into three groups and are given in the following.

(1) The queue is empty, and not all class-1 servers are busy.

For $0 \leq j \leq 2^C - 1$ and $\prod_{n=1}^{C_1} j_n = 0$,

$$\left(\lambda + A_j\right) \pi_{0,j} = B_j + F_j \tag{1}$$

where $A_j = \sum_{k=1}^{C} \mu_k j_k$, $B_j = \sum_{k=1}^{C} \mu_k \pi_{0,j+2^{k-1}} (1 - j_k)$, $F_j = \sum_{k=1}^{C_1} \lambda \pi_{0,j-2^{k-1}} (\prod_{n=1}^{k} j_n)$.

(2) The queue is empty, and all class-1 servers are busy.

For $0 \leq j \leq 2^C - 1$ and $\prod_{n=1}^{C_1} j_n = 1$,

$$\left(\lambda + A_j\right) \pi_{0,j} = B_j + F_j + I_j \pi_{1,j} \tag{2}$$

where $I_j = \sum_{k=1}^{C_1} \mu_k$.