Computers & Industrial Engineering 93 (2016) 227–235

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Performance analysis of a two-node computing cluster

Jinting Wang [a,*], Shan Gao [a,b], Tien V. Do [c]

[a] Department of Mathematics, Beijing Jiaotong University, Beijing 100044, China
[b] Department of Mathematics, Fuyang Normal College, Fuyang 236037, China
[c] Analysis, Design and Development of ICT Systems (AddICT) Laboratory, Budapest University of Technology and Economics, H-1117, Magyar tudósok körútja 2., Budapest, Hungary

ARTICLE INFO

Article history:
Received 27 October 2014
Received in revised form 28 September 2015
Accepted 8 January 2016
Available online 15 January 2016

Keywords:
Queueing system
Synchronous working vacation
Impatient customers
Generating function
Hypergeometric function

ABSTRACT

This paper proposes a queueing model for the performance evaluation of a computing cluster. Requests arrive to the configurations according to a Poisson process at rate λ. There is a load balancing hardware in front of two servers to route requests to a free server if any. The service is commenced according to the first-come-first-served (FCFS) principle. The system can be either in the normal state or in the working vacation state. The service time of each customer during the normal state is exponentially distributed. We develop the balance equations for the steady-state probabilities and solve the equations by using the generating function and hypergeometric function. We obtain some performance measures for the system, such as the steady-state probabilities of the servers, the mean system sizes, the mean sojourn time of a customer served as well as the proportion of customers served and the rate of abandonment due to impatience. Finally, some numerical results are presented to demonstrate effects of some parameters on these performance measures of the system.

1. Introduction

Nowadays, the information technology (IT) service is carried out with the help of different application servers (e.g.: Web and database servers, etc.) which are hosted in physical servers (often called machines or nodes). Therefore, the cost-efficient organization of a system providing services plays an important role in the success of service providers, which can be supported by a new queueing model developed in this paper. In the last decades, there are queues with working vacations which have been widely used for the performance analysis of communication systems, manufacturing systems, bank service systems and other public service systems (Do & Krieger, 2009; Servi & Finn, 2002). The working vacation term in queueing models has been motivated by the operation of practical systems where less resource may be available for certain periods than the normal operation periods.

The pioneering work on working vacation queue can go back to Servi and Finn (2002) for an M/M/1 queue with multiple working vacations. Later, the work of Servi and Finn (2002) has been extensively generalized. For example, Wu and Takagi (2006), Li, Tian, Zhang, and Luh (2009), Gao and Liu (2013) generalized results in Servi and Finn (2002) to the M/G/1 queue with working vacation. Baba (2005) studied a GI/M/1 queue with working vacation by using the matrix-analytic method. Tian, Ma, and Liu (2008) studied a discrete time Geom/Geom/1 queue with multiple working vacation, Tian, Zhao, and Wang (2008) studied an M/M/1 queue with single working vacation, Lin and Ke (2009) presented a multi-server system with single working vacation. Do and Krieger (2009) used a multi-server with vacations to model management activities in cloud computing environments. Recently, Gao, Wang, and Zhang (2013) presented the GI^X/Geo/1/N queue with negative customers and multiple working vacations, Li, Zhang, Xu, and Gao (2013) studied the GI/Geo/1 queue with Bernoulli-schedule-controlled vacation and vacation interruption. Some recent works on vacation queue model can be found in Ke, Wu, and Zhang (2010) as well. Do, Papp, Chakka, Sztrik, and Wang (2014) analyzed the M/M/1 retrial queue with working vacations and negative customer arrivals, which is an extension of a model introduced in Do (2010).

In principle, customers in waiting line are prone to leave the service system due to impatience and long time delays. In the queueing literature, Yechiali (2007) considered an M/M/c queue, where the system, considered as a whole, suffers occasionally a disastrous breakdown and immediately enters into the repair period, each individual customer who arrives during repair period activates a random-duration timer (impatient timer). If the timer expires before the system is repaired, the customer who abandons the queue will never return. Altman and Yechiali (2008) studied an M/M/1 queue with impatient customers. Yue, Yue, and Xu (2012) considered an M/M/1 queueing system with working vacations

* Corresponding author. Tel.: +86 13681373723.
E-mail address: jtwang@bjtu.edu.cn (J. Wang).

http://dx.doi.org/10.1016/j.cie.2016.01.006
0360-8352/

and impatient customers, where customers will abandon the system during working vacation if their service has not been completed or has not begun before the expiration of an impatient timer. Different from Yue et al. (2012), Selvaraju and Goswami (2013) considered an M/M/1 queue with impatient customers and two different types of working vacations in which the customers arriving at the system during a working vacation period become impatient and may abandon the system because of the expiration of their impatient timer.

In this paper, we propose a novel model to analyze a typical cluster configuration that is often applied in small enterprises, departments of universities and even big companies. In such environments, information technology services and mission-critical applications (e.g., mail servers, database service, web servers, etc.) are provisioned with the help of a physical server. To provide highly available services against the failure of hardware components and/or software, a clustered system is often applied with hardware and software redundancy. In a lot of cases, a cluster is built with two physical servers because of the cost reason. Each physical server runs a copy of an operating system, application software and an appropriate framework. The most well-known open and free software framework to build a Linux cluster using commodity hardware is developed in the framework of the Linux-HA project (http://www.linux-ha.org). The responsibility of the framework is to detecting node or daemon failures and reconfigure the system appropriately, so services can be provided without interruption if one component fails. In this paper, we consider an active-active load balancing cluster (see Kopper, 2005; van Vugt, 2014) where servers synchronously exchanges background information (e.g., internal state of service, file directories) so each server is backup of the other server. Requests arrive to the configurations according to a Poisson process and can be served by any of two physical servers according to the first-come first served (FCFS) principle. Due to the active-active load balancing configuration, two practical cases are taken into consideration when the system becomes empty:

- The two physical servers synchronously apply the Dynamic Voltage and Frequency Scaling (DVFS) technique (i.e., reduce the operating voltage/frequency of CPU) (see Weiser, Welch, Demers, & Shenker, 1994) to save the energy consumption.
- The two physical servers are synchronously brought to a maintenance phase. In this case, some software updates or hot-swap hardware upgrades are performed.

In the term of the queueing theory, the two servers synchronously start a single vacation with an exponentially distributed random period. The system goes back to the normal state after a vacation. The system can serve arriving requests with a lower service rate during a vacation than the normal case.

This work is different from that of Selvaraju and Goswami (2013) and the differences are listed as follow:

- We consider an M/M/2 queue model with synchronous single working vacation based on a practical two-node computing cluster.
- We derive some key performance measures explicitly such as sojourn time of a customer who finishes his service completely, the proportion of customers served, and the rate of abandonment of a customer due to impatience.
- Numerical experiments are presented for a cluster model that is built from Commercial Off-The-Shelf (COTS) servers by considering the average power consumption.

The rest of this paper is organized as follows. In Section 2, we give the details of a computing cluster and model it as an M/M/2 queue with synchronously working vacation and impatient customers. In Section 3, the steady-state analysis is presented. In Section 4 the sensitivity analysis on some performance measures is given through numerical examples. Section 5 concludes our paper.

2. A performance of a computing cluster

A cluster of two physical machines depicted in Fig. 1 is typically applied in the environment of IT infrastructure to provide a service (e.g., hosted services based on application server environments), see Marcus and Stern (2003) and Kopper (2005). Requests arrive to the configurations according to a Poisson process at rate λ. Before the two servers, there is a load balancing hardware with software that is used to route requests to a free server (see Fig. 1).

The service is commenced according to the first-come-first-served (FCFS) principle. The system can be either in the normal state or in the working vacation state. The service time of each customer during the normal state is exponentially distributed with mean $1/\mu_b$, where we assume that the stability condition $\rho = \lambda/(2\mu_b) < 1$ is fulfilled.

When the system becomes empty, the two physical servers synchronously apply the Dynamic Voltage and Frequency Scaling (DVFS) technique (i.e., reduce the operating voltage/frequency of CPU) (see Weiser et al., 1994) to save the energy consumption of the cluster. In the term of the queueing theory, the two servers synchronously start a single vacation with an exponentially distributed random period. The vacation time V is assumed to follow an exponential distribution with mean $1/\theta$. During a vacation, arriving requests can be served during the vacation periods of servers. The service times during the working vacation follow an exponential distribution with rate μ_v. Note that $0 < \mu_v \leqslant \mu_b$ because the CPUs of servers are operated with a reduced frequency. If the idle servers return from the vacation, they stay in the system waiting for arriving customers without taking another working vacation. Correspondingly, they change the service rate μ_v back to the regular rate μ_b.

Due to the servers' lower service rates during the working vacation, the customers waiting in the queue are assumed to be impatient. That is, whenever a customer arrives during a working vacation period and finds the two servers busy, he/she activates an impatience timer 'T', which follows an exponential distribution with parameter $\xi(\xi > 0)$ and is independent of the number of customers in the system. If the impatience timer expires before the end of the working vacation state and the customer's service has not begun, the customer leaves the system forever. Otherwise, once the customer's service begins before the expiration of the impatience timer, she will stay in the system.

Various stochastic processes involved in the system are independent of each other.

3. Steady-state analysis

In this section, we carry out a stationary analysis for the model given in Section 2. We first derive the balance equations for the steady-state probabilities and then solve the equations by using the generating function method.

The state of the system at time t is described by the following random variables:

(1) The number $N(t)$ of customers in the system including the ones in service.
(2) The state $J(t)$ of the server, where server may be in an SWV (Single working vacation) ($J(t) = 0$) or in a regular service period (idle or busy with normal service rate) ($J(t) = 1$).