Computer Communications 128 (2018) 46–59

Contents lists available at ScienceDirect

Computer Communications

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

Modeling data backups as a batch-service queue with vacations and exhaustive policy

Apoorv Saxena*[,a], Dieter Claeys[1,a,b,c], Herwig Bruneel[a], Bo Zhang[d], Joris Walraevens[a]

[a] SMACS Research Group, Department TELIN, Ghent University, Sint-Pietersnieuwstraat 41, Gent B-9000, Belgium
[b] Department of Industrial Systems Engineering and Product Design, Ghent University, Technologiepark 903, Zwijnaarde B-9052, Belgium
[c] Industrial Systems Engineering (ISyE), Flanders Make, Belgium
[d] IBM Research AI, Yorktown Heights, NY 10598, USA

ARTICLE INFO

Keywords:
Data backup
Batch service
Queueing model
Vacation
Recovery point objective

ABSTRACT

For data backup processes to cloud infrastructure, there is a clean trade off between backing up frequently (improving data safety) and reducing resource usage (power consumption and communication cost). With rapid growth of data storage requirements in recent years, we need to find the right balance between both objectives. To explicitly address this trade off, we model a wide set of exhaustive data backup processes as a general batch service queueing model with multiple vacations and probabilistic restarts.

We study this queueing model and establish expressions for its performance measures such as system content and queue content distributions. This analysis aids in computing Quality of Service (QoS) measures of the data backup process such as the fraction of time the backup server is busy, the frequency of new connections and the age of the data at the beginning of a backup period. This enables us to quickly examine the dependence of QoS on the model parameters as well as to compute the optimal parameters in the backup process. We illustrate the latter by defining a particular cost function of a user and by framing an optimization problem.

1. Introduction

Processes which involve movement of data, including backup processes, consume much electricity and bandwidth. A significant part of this power is utilized to keep the system on in an idle state. Nguyen et al. [1] state, from the estimates of the US Environment Protection Agency, that 1.5% of the total power produced in US was used for Data Center computing. Further, they note that the ICT industry causes 2% of global CO_2 emissions. Therefore, in addition to higher electricity cost, under-utilization of computing resources leads to higher carbon footprint. Hence, new models are being proposed, such as in Guan et al. [2], to minimize the energy usage in data centers. Studies such as Chen et al. [3] have analyzed that the requirement of data storage technologies will grow exponentially and may grow to as high as 40 ZB by 2020 from 1.8 ZB in 2011. With this rapid increase machines need to schedule data backup processes efficiently and cost effectively.

Organizations have started moving from local disks towards cloud systems as primary storage nodes to run operations. Amazon [4] provides a list of major companies using its cloud services which includes some very well-known companies such as Netflix and Thomson Reuters. This move is driven by many remarkable benefits of the cloud infrastructure such as ease of setup, higher reliability, availability, security and protection from regional calamities or power failures (see Chang and Wills [5] for more details).

With the rapid increase in data storage requirements and migration towards cloud storage, the right balance between doing backups very frequently and reducing resource consumption has to be found. To help finding this balance, we model a wide set of data backup processes as a generic batch service queueing model with vacations, exhaustive service and probabilistic restarting conditions. As we will explain in Section 2, batch service queueing models with vacations provide a very natural way of modeling of data backups to cloud infrastructure. We are able to precisely compute the Quality of Service (QoS) measures of the data backup processes using this model. Nevertheless, to the best of our knowledge, data backup processes have been rarely modeled and analyzed as queueing processes. Yu et al. [6] propose a queueing model of cloud based streaming services to evaluate service quality. Van de Ven et al. [7] model data backups as a two dimensional Markov chain and

* Corresponding author.
E-mail addresses: saxena.apoorv@ugent.be (A. Saxena), Dieter.Claeys@UGent.be (D. Claeys), Herwig.Bruneel@UGent.be (H. Bruneel), zhangbo@us.ibm.com (B. Zhang), Joris.Walraevens@UGent.be (J. Walraevens).
[1] www.FlandersMake.be

https://doi.org/10.1016/j.comcom.2018.07.014
Received 23 March 2018; Received in revised form 6 June 2018; Accepted 6 July 2018

study optimization of the network bandwidth utilization. Xia et al. [8] present decision algorithms, utilizing Markov decision processes, which use constraints on recovery parameters to decide when to backup and which type of backup to perform.

Queueing models with vacations, in general, have been studied extensively in the literature (see Doshi [9] and Naishuo and George [10] for details about some applications). In particular, some relevant batch service queueing models with vacations have been studied. Sikdar and Gupta [11] study a server which goes into a vacation when the system is empty and restarts service on return if there is at least one customer in the system. Arumuganathan and Jeyakumar [12] study a batch service model with multiple vacations where the server requires a minimum threshold of queue size to keep the server on and another threshold to restart it after the vacation. The model also takes into account a setup and close down time for the server. Other models studied by Sikdar and Gupta [13] and Lee et al. [14] consider a single vacation case with a threshold where the system waits in idle condition after returning from a vacation if the threshold is not matched. The queueing models listed above have not been built and studied from the perspective of data backup processes. Our model, on the other hand, is specifically designed to model a wide range of exhaustive data backup processes.

The purpose of our paper is to model exhaustive data backup processes as a very general batch service queueing model with vacations and probabilistic restarting conditions. It is already well known that discrete-time models are more appropriate to model telecommunication processes (see Bruneel and Kim [15]). We thoroughly analyze the discrete time queueing model and obtain various performance measures. We then explicitly translate performance measures of the queueing model to QoS measures of data backup processes. Our model and the QoS measures are useful to find the right balance between doing cloud backups frequently enough (to increase data safety) and reducing resource usage (power consumption and communication costs). We study the dependence of QoS measures on the model parameters as well as compute the values of these parameters that minimize the user cost function. Note that, as stated in Chen and Trivedi [16], pure threshold policies may not be optimal. Therefore, we define parameters for probabilistic restarts to model the behavior of the server. Moreover, since the cost of connecting to cloud infrastructure is typically independent of the amount of data backed up, our work focuses on exhaustive policies (see eg the cost structure of Amazon Web Services [4] and Microsoft Azure Storage [17]).

This paper presents an extension of our previous work Claeys et al. [18]. The model studied by Claeys et al. [18] assumes single slot vacations and single slot service times. Our model extends the vacation length and service lengths to general distributions as well as introduces probabilistic restarting conditions. Making the model much more general implies that it models a wider set of exhaustive data backup policies. Additionally, this work looks at significantly more performance measures and a more thorough modeling of data backups.

We characterize our model using four types of parameters: l (threshold to begin service with probability 1), c (capacity of backup server), $\alpha_1, \alpha_2, \ldots, \alpha_{l-1}$ (starting probabilities) and T (a random variable which governs the distribution of vacation lengths). Moreover, both the arrival distribution (the number of arrivals in a slot) and service times (which depend on the batch size) can be chosen freely provided that the system remains stable. These flexible parameters empower us to set their values based on the traffic seen by the server and required system performance. Our aim is to answer questions such as: given the service level agreements, what are the right model parameters to maintain a stable and efficient system? Such questions can be answered accurately once we have computed the exact expressions of the main QoS measures in terms of the parameters. The main advantage of our analysis is that we can compute the QoS measures for a general set of parameters.

We start with a description of the model and justify its suitability for data backups in Section 2. We also discuss in detail the utility of each parameter of the model. It is then followed by the analysis of the model in Section 3. We also highlight some observations at the service/vacation epochs which give us some important relations to solve our model. Using the results of Section 3, we deduce the QoS measures of data backups in Section 4. We compute measures such as the age of data at the beginning of a backup period and the probability that the backup server is busy in a slot. We also evaluate the performance of the system for different system parameters in Section 5. Further, we construct an optimization problem to highlight how one can compute the optimal parameters to minimize user cost. In the last section we summarize our observations and results from previous sections and present possible directions for future work.

2. Backup process model

In data backup processes, packets that have not yet been backed up are part of the backlog. The backlog size then corresponds to the number of packets that have not yet been backed up. In case of exhaustive backup service processes, when the backup server initiates a backup, it continues backing up until the backlog size becomes zero.

We model the backlog as a queue where the customers represent data packets and the backup server by a batch server. As a result, packets arriving in the queue correspond to newly generated packets that have not yet been backed up.

The batch server has a capacity of maximum c packets and it employs an exhaustive policy i.e. the server keeps serving until the system content becomes zero. At that moment, the backup server immediately goes into energy saving mode which is modeled as a vacation of T slots with T a random variable. If upon returning from the vacation, the system content (i.e. the total number of packets in the queue and the server) is larger than or equal to l, the server immediately initiates the service (a new backup); otherwise it starts the service with a probability α_i, where i is the number of packets in queue. With probability $1-\alpha_i$ the server goes into another vacation i.e. stays in the energy saving mode. Lengths of vacation periods are independent and identically distributed.

2.1. Data storage on cloud infrastructures

A cloud data backup process in an organization involves several components as illustrated in Fig. 1. This setup is commonly used across the industry to perform their data backups such as services offered by company host-it [19]. Generally, the cloud infrastructure services are provided by an external organization such as Amazon [4], Microsoft [17]. The namenode drives and monitors all the data processes running on the cloud infrastructure while the data nodes store the data fragments of all the users. All the devices in the organization communicate over LAN and the data generated is kept in sync with the data packets queue (see Fig. 1). The local backup server coordinates the backup of the data packets in the data packets queue to the cloud infrastructure. It communicates with the namenode of the cloud infrastructure and drives the whole backup process. It is exactly the data packets queue and the backup server that are modeled by our queueing model. Fig. 2 highlights the components of our queueing model and connects it with the cloud data backup process of Fig. 1.

When the backup server wants to write data, it sends a request to the namenode. Namenode stores the metadata of the data to be stored and sends back the IP-addresses of the data nodes on which the backup server is allowed to write the data. A data packet completes its service when it has been written to the cloud infrastructure. Our work assumes that the backups performed are incremental in nature, i.e., only the changes in the system from the last copy are saved and backed up.

Now we discuss the model choices and the importance of the model parameters and characteristics.

Why is it a batch service model?

Data packets are segmented and transmitted in batches from the