Semi-markov model of processing requests to the cloud storage

Mikhail Zamoryonov ^{1,*}, Victoria Shevchenko ¹, Elena Vladimirova ¹, and Nikolay Sapozhnikov ²

¹Sevastopol State University, Department of Information Technologies and Computer Systems, Sevastopol, Russian Federation

²Federal state owned military educational institution of the Higher professional Education “the black sea higher navy order of the red star academy by P.S.Nakhimov” Russian Federation, 299028, Sevastopol

Abstract. This paper presents a semi-Markov model as an important part of the computer-aided manufacturing and a modular system of cloud storage, which affects the functioning of the whole process. The residence times of the system in states and the probability of system transitions are determined. There is a stationary distribution of the embedded Markov chain. The residence times of the system in states with allowance for repeated hits are determined by the theorem on the distribution functions of the residence times of the system in states with allowance for repeated hits. When using the trajectory method, the distribution function of the time for the complete processing of the read request by such a system is determined. A comparison of the expectation time complete processing the read request obtained in this study and obtained by the known formula for determining from the literature the expectation residence time in the subset of the system states.

1 Introduction

Computer-aided manufacturing today is a complex-structured multi-purpose hardware and software systems operating on a wide range of problems with the use of advanced information technology and advanced software and hardware tools, to effectively carry out operational planning and management of the whole enterprise as a whole. At the same trends in the development of such systems is largely due to the transfer of software geographically distributed industrial companies by means of Internet communications in the virtual space. The introduction of the concept of virtual enterprise (extended enterprises) allows to provide technological support for remoting professionals and individual units of mechanical assembly production at the stages of design, production, marketing and other stages life cycle of products.

One of the most promising information technologies in this direction is the technology of cloud computing. Cloud computing, according to [1], provide a model for convenient network access on demand to the total pool of configurable in computing resources (networks, servers, storage, applications and services) that can quickly identify and provide with minimal management effort or minimal intervention on the part of the supplier of cloud services [2]. The introduction of cloud computing technology has some significant advantages [3-6]: the enterprise by eliminating the need to invest substantial sums to machine tool making, purchase of expensive software (ERP, PDM, CAD / CAM / CAE-systems); Economy of computing resources; Possibility of mobility of workplaces; Scalability, which increases the efficiency of computing resources through the use of virtualization technologies; saving on maintenance of the

IT department, these functions are moving to outsourced cloud providers.

One common use for this cloud for industrial automation is the placement of the cloud of part or all SCADA-system (Supervisory Control and Data Acquisition). Cloud technologies are used by SCADA applications in two main ways: 1) the SCADA system is installed locally, directly connected to the control network, and also sends information to the cloud where it can be stored and distributed from there. 2) the SCADA system is fully deployed in the cloud and remotely connected to the control network. The data received from these systems in a network cloud storage for later use MES-systems (Manufacturing Execution System) and other industrial automation systems. It should be noted that the volumes of transmitted, processed and stored information are very significant and can be attributed to the Big Data. In this connection, the problems associated with determining the performance of these systems become particularly topical. We also note that mathematical models allow not only to determine productivity, but also to indicate ways for increasing it.

The primary characteristic processing systems and data storage is waiting for the response to the request or the response time. Given that the response time is a random variable that depends on a number of factors different for each particular implementation of the system, the definition of this time is not an easy task. As part of further studies, the inventors have focused on one of the technologies of data storage systems – block storage [7] as the most widely used for industrial automation systems.

*Corresponding author: zamoryonoff@gmail.com

The purpose of this paper is to determine the distribution function (FF) of the response time of a block cloud storage system to a data read request.

Consider a block system of cloud storage of general data. This system works as follows. Modular storage system (NAS) provides computer systems access to a block-level storage volumes. In this environment, the file system is created on the basis of computer systems, and access to the data is carried out through the network at the block level. A block storage system consists of one or more controllers and a storage system. In its turn, the controller storage system consists of three key components: a client interface (external ports and controllers), cache memory and the server part. The structure of the block-type storage intelligent system is shown in Fig. 1, a detailed description is given in [7].

The Input/Output request is transferred from the computer system via an external port and processed using the server's cache memory. Read request can be

served directly from the cache, if it found the necessary data to the user. In modern storage systems, the client interface, cache memory and server part are usually integrated into a single panel (which is called the storage processor or storage controller).

The client interface provides data exchange between the storage system and the computer system. It consists of two components: external ports and external controllers. Typically, the client interface has redundant controllers for high availability, and each controller contains several ports that allow multiple computer systems to be connected to an intelligent data storage system.

Fig. 2 shows the read operation using the cache memory subject to the availability (a) and unavailability of (b) in the data cache.

Fig. 1. Components intelligent data storage system.

Fig. 2. Read using cache operations

When the computer system sends a read request, the storage controller reads the RAM label to determine the availability of the necessary data in the cache. If the requested data is found in the cache, this is called a hit in the read cache or hit, while reading, and the data is sent directly to the computer system without performing operations in the internal storage. This provides for the computer system a short response time (about milliseconds). If the requested data is not found in the cache, then this is called a cache miss. In this situation, the data must be read from the storage system. The server part gains access to the appropriate storage device and retrieves the requested data. The data is placed in the cache memory, and then sent to the computer system using the client interface. Cache misses increase the response time for Input/Output operations.

The performance of reading is measured by the hit rate of read operations or the frequency of hits, which is usually expressed as a percentage. This ratio is the ratio of the number of read hits to the total number of read requests. The higher the hit rate of read operations is, the higher the performance of these operations is.

It should be noted that the response time also depends on the reliability of data transmission networks.

To simulate storage block type is proposed to use trajectories method given in [8]. This method allows one to obtain an exact solution of the Markov recovery equations for a system with a discrete phase space of states.

2 Formulation of the problem

We pose the problem in the following way: Distribution functions (DF) $F_1(t)$ and $G_1(t)$ of the random variables (RV) α_1 and γ_1 , which are sometimes processing read requests from the user to the cache memory and from the cache memory to the storage data server (SDS), respectively, and DF $F_1(t)$ and $G_1(t)$ of random variables α_2 and γ_2 – time between failures in the channels of communication between the user and the memory modules and between the cache and SDS. In addition, DF $R(t)$ RV β , being the time search data in the cache memory and the probability p of finding the desired information in the memory are known. Also the probability of finding the data in the cache memory of the system p is known P .

When building the model it is assumed that when a failure occurs in the data transmission channel of the user performing a read request is restarted.

In the current study, it is necessary to determine the time distribution function for the complete processing of the read request by the specified system.

3 Building a model

The functioning of the system is described by a semi-Markov process (SMP) $\xi(t)$ with a discrete-continuous phase space of states [9, 10, 11, 12, 13]. We introduce

the following set E of Semi-Markov states of the system:

$$E = \{S_0, S_1, S_2, S_3\}$$

Let us decipher the meaning of the state codes:

S_0 - the request is executed (instantaneous state);

S_1 - a request from the user is received (or when the query has failed in data transmission channel) the re-execution request is beginning;

S_2 - searches for the data in the cache memory modules;

S_3 - the data in the cache memory is not found, a request is made to SDS (or during a query to SDS, the failure occurred in the data channel), the process is beginning with the start request.

Count system states is shown in Fig. 3.

Fig. 3. System graph.

Times of stay $\theta_0, \theta_1, \theta_2, \theta_3$ of the system states S_0, S_1, S_2 and S_3 are equal to:

$$\theta_0 = 0; \theta_1 = (\alpha_1 \wedge \alpha_2); \theta_2 = \beta; \theta_3 = (\gamma_1 \wedge \gamma_2),$$

where $\wedge$ is a sign denoting a minimum of random variables.

Then the DF of the times of the system stay in the states are equal:

$$F_{S_0}(t) = 0;$$

$$\bar{F}_{S_1}(t) = \bar{F}_1(t) \cdot \bar{F}_2(t);$$

$$F_{S_2}(t) = R(t);$$

$$\bar{F}_{S_3}(t) = \bar{G}_1(t) \cdot \bar{G}_2(t).$$

We describe the probabilities of transitions of the embedded Markov chain (ECM) (1).

$$\begin{cases} P_0^1 = 1; \\ P_1^1 \{\alpha_1 > \alpha_2\} = \int_0^\infty \bar{F}_1(t) f_2(t) dt; \\ P_1^2 \{\alpha_1 < \alpha_2\} = \int_0^\infty f_1(t) \bar{F}_2(t) dt; \\ P_2^0 = p; \\ P_2^3 = 1 - p; \\ P_3^3 = \int_0^\infty \bar{G}_1(t) g_2(t) dt; \\ P_3^0 = \int_0^\infty g_1(t) \bar{G}_2(t) dt. \end{cases} \quad (1)$$

Using system (1), we write the system of equations for determining the stationary distribution of the embedded Markov chain $\rho(x)$ [14, 15, 16]:

$$\begin{cases} \rho_0 = \rho_2 \cdot P_2^0 + \rho_3 \cdot P_3^0 \\ \rho_1 = \rho_0 \cdot P_0^1 + \rho_1 \cdot P_1^1 \\ \rho_2 = \rho_1 \cdot P_1^2 \\ \rho_3 = \rho_2 \cdot P_2^3 + \rho_3 \cdot P_3^3 \end{cases} \quad (2)$$

Normalization condition:

$$\rho_0 + \rho_1 + \rho_2 + \rho_3 = 1. \quad (3)$$

Solving the system of equations (2) using condition (3), we obtain:

$$\begin{cases} \rho_0 = \frac{P_1^2 \cdot P_3^0}{2 \cdot P_1^2 \cdot P_3^0 + P_3^0 + P_1^2 \cdot P_2^3}; \\ \rho_1 = \frac{P_3^0}{2 \cdot P_1^2 \cdot P_3^0 + P_3^0 + P_1^2 \cdot P_2^3}; \\ \rho_2 = \frac{P_1^2 \cdot P_3^0}{2 \cdot P_1^2 \cdot P_3^0 + P_3^0 + P_1^2 \cdot P_2^3}; \\ \rho_3 = \frac{P_1^2 \cdot P_2^3}{2 \cdot P_1^2 \cdot P_3^0 + P_3^0 + P_1^2 \cdot P_2^3}. \end{cases}$$

We represent the set of all states in the form: $E = E_+ \cup E_-$, $E_+ = \{S_1, S_2, S_3\}$, $E_- = \{S_0\}$. A subset of states E_+ describes the process of servicing the read request.

To determine the DF complete request processing time for reading system it is used the method of trajectories [8]. For this we use the theorem on the DF time the system spends in the states considering re-entering [8].

Define the trajectories of the system's output into a subset $T_1 = \{S_1, S_2\}$, $T_2 = \{S_1, S_2, S_3\}$.

We define the residence time distribution function of the system in the states 21 and 20 and with the DF repeated hits by the formulas (1):

$$F_{S1}^\theta(s) = \frac{F_{S1}(s)}{c_1 - (c_1 - 1)f_{S1}(s)},$$

$$F_{S3}^\theta(s) = \frac{F_{S3}(s)}{c_3 - (c_3 - 1)f_{S3}(s)}.$$

$$\text{where } c_1 = \frac{\rho_1}{\rho_0 \cdot P_0^1} = \frac{1}{P_1^2}; \quad c_3 = \frac{\rho_3}{\rho_0 \cdot P_0^1} = \frac{1}{P_3^0}.$$

With sequential convolution DF time of system stay is defined in each path:

$$F_1^T(t) = F_{S1}^\theta(t) * F_{S2}(t);$$

$$F_2^T(t) = F_{S1}^\theta(t) * F_{S2}(t) * F_{S3}^\theta(t);$$

where $*$ is the sign of the convolution operation.

Let us find the probability of realizing each of the trajectories:

$$P_1^T = p; \quad P_2^T = 1 - p,$$

moreover, $P_1^T + P_2^T = 1$.

Hence it becomes possible to determine the DF $F_\Sigma^\theta(t)$ of the time of the system stay in the subset E_+ regardless of the initial state, as a weighted sum of (a mixture of) the DF system residence times in the subset each trajectory with coefficients equal to the probability of implementing these trajectories

$$F_\Sigma^\theta(t) = F_1^T(t) \cdot P_1^T + F_2^T(t) \cdot P_2^T. \quad (4)$$

The results of simulation by the formula (4) shown in Fig. 4.

Fig. 4. Type of DF of full read request processing time.

The initial data for modeling are DF $F_1(t)$, $F_2(t)$, $G_1(t)$, $G_2(t)$, $R(t)$, which are corresponded to the generalized second-order Erlang law with parameters λ_1 , λ_2 ; μ_1 , μ_2 ; ν_1 , ν_2 ; τ_1 , τ_2 , respectively. And:

$$f_1(t) = \frac{\lambda_1 \lambda_2 (e^{-\lambda_1 t} - e^{-\lambda_2 t})}{\lambda_2 - \lambda_1}, \quad f_2(t) = \frac{\mu_1 \mu_2 (e^{-\mu_1 t} - e^{-\mu_2 t})}{\mu_2 - \mu_1},$$

$$g_1(t) = \frac{\nu_1 \nu_2 (e^{-\nu_1 t} - e^{-\nu_2 t})}{\nu_2 - \nu_1}, \quad g_2(t) = \frac{\tau_1 \tau_2 (e^{-\tau_1 t} - e^{-\tau_2 t})}{\tau_2 - \tau_1},$$

where $\lambda_1=1,101$ (s^{-1}); $\lambda_2=10,899$ (s^{-1});

$\mu_1=0,0688$ (s^{-1}); $\mu_2=0,681$ (s^{-1});

$\nu_1=1,101$ (s^{-1}); $\nu_2=10,899$ (s^{-1});

$\tau_1=0,0688$ (s^{-1}); $\tau_2=0,681$ (s^{-1});

$\tau_1=11,01$ (s^{-1}); $\tau_2=108,99$ (s^{-1}).

And also the probability of finding data in the cache memory of the system $p=0,4$.

Let us compare the values of the mathematical expectation of the function (4) obtained by us and the mathematical expectation determined by the expression [14]:

$$T_+ = \frac{\sum_{i \in M_+} m_i \rho_i}{\sum_{e \in E \subset M_+} \sum_{j \in M_+} P_{ej} \rho_i}. \quad (5)$$

The mathematical expectation of the distribution function obtained by us is 1.7037037 sec.

While determined by formula (5) - 1.7037037 sec.

4 Conclusions

A comparison of the values of the mathematical expectation of the time for the complete processing of the read request by the data storage system determined using the distribution function obtained by us and obtained using the formula given in [16] showed a complete coincidence of the results.

The results allow to determine the response time for reading by the cloud storage system request threads cloud storage, which allows to evaluate the impact of the information system performance in the adaptability of the process and, if necessary, to make a decision to increase the speed of the system. The resulting simulation of the distribution function allows the use of this model in the construction of models of more complex hierarchically subordinate information and production systems.

This work was supported by grant s Russian Foundation for Basic Research № 15-01-05840, № 15-29-07936.

References

1. NIST Special Publication 500-307 Cloud Computing Service Metrics Description, NIST

Cloud Computing Reference Architecture and Taxonomy Working Group. The Web: <http://www.nist.gov/itl/cloud/uploadthe/RATAX-CloudServiceMetricsDescription-DRAFT-20141111.pdf>

2. A.V. Skatkov, V.I. Shevchenko, *Extension of the reference model of the cloud computing environment in the concept of large-scale scientific research*, Proceedings of ISP RAS. № 6, pp. 285-306 (2015)
3. F. Doelitzscher, T. Ruebsamen, T. Karbe, M. Knahl and C. Reich, *Sun Behind Clouds - On Automatic Cloud Security Audits and a Cloud Audit Policy Language*, International Journal on Advances in Networks and Services, № 6, (1&2) (2013)
4. S. Singha, S.D. Satav, *A Survey Paper on Cloud Storage Auditing with Key Exposure Resistance*, International Journal of Science and Research (IJSR), Vol. 4, pp. 1821-1823. (2015)
5. B. Wang, B. Li and H. Li. *Public Auditing for Shared Data with Efficient User Revocation in the Cloud*, Web: iqua.ece.toronto.edu/papers/bwang-infocom13.pdf
6. M. Hogan, F. Liu, A. Sokol and J. Tong, *NIST Cloud Computing Standards Roadmap*, Computer Security Division Information Technology Laboratory Gaithersburg, MD 20899-8930 (2011)
7. From data storage to information management. Peter. P.544 (2016)
8. M. V. Zamoryonov, V. Ya. Kopp, Yu. E. Obzherin, D.V. Zamoryonova, *Approbation of the method of trajectories on the example of modeling the process of functioning of the production element with devaluating failures*, Proceedings of Tula State University. Technical science № 8, pp. 57-71 (2015)
9. Yu. E. Obzherin, E.G. Boyko, *Semi-Markov Models. Control of Restorable Systems with Latent Failures* (Elsevier, Academic Press, 2015)
10. A.I. Peschansky, *Semi-Markov Models of One-Server Loss Queues with Recurrent Input* (LAP LAMPERT Academic Publishing, 2013)
11. F. Bayhelt, P Franken, *Reliability and maintenance. Mathematical approach*, Radio and Communication, (1988)
12. V.S. Korolyuk, *Superposition of Markov recovery processes*, Cybernetics №4, pp. 121 – 124 (1981)
13. V.S. Korolyuk, *Semi-Markov processes and their applications* (Science, Dumka, 1976)
14. V.S. Korolyuk, A.F. Turbin, *Markov recovery processes in reliability problems of systems* (Naukova Dumka, 1982)
15. V.S. Korolyuk, *Stochastic models of systems* (Naukova Dumka, 1989)
16. K. Rainshke, I.A. Ushakov, *Evaluation of the reliability of systems using graphs*, Radio and Communication (1988)