Request Stream Control for the Access to Broadband Multimedia Educational Resources in the Distance Learning System

http://dx.doi.org/10.3991/ijep.v3i4.2874

I.P. Bolodurina, A.E. Shukhman and D.I. Parfenov
Orenburg State University, Orenburg, Russia

Abstract—**This article presents a model of queuing system for broadband multimedia educational resources, as well as a model of access to a hybrid cloud system storage. These models are used to enhance the efficiency of computing resources in a distance learning system. An additional OpenStack control module has been developed to achieve the distribution of request streams and balance the load between cloud nodes.**

***Index Terms*—cloud computing, multimedia educational resources, load balancing, hybrid cloud system OpenStack.**

I. INTRODUCTION

Nowadays, the growth of information exchange demands flexible solutions for serving many concurrent users. Thus, one of the objectives is flexible control and efficient use of the available computing resources.

Among the most promising is cloud computing. This technology allows to unify the access to resources, that is very important for systems maintaining a high quality of service and 24/7 service availability. Cloud computing market includes not only proprietary software such as VMware ESX, Xen etc. but also well-documented open source solutions such as OpenStack [10]. These solutions are used for storage of video multimedia resources in distance learning technologies. Typically, distance learning systems are multicomponent systems, that are able to handle various information services at the same time. Therefore there is a need for a special approach to the organization of control infrastructure as a whole, as well as each component.

We have established the following features of software and hardware resources usage in distance learning system (DLS) of Orenburg State University (OSU):

- load on key resources is periodic and uneven;

- requests to several types of resources are simultaneous;

- the intensity of requests to resources may be outside dependent;

- due to the lack of load balance the equipment is not always able to serve all the requests at the peak load;

- up to 90% of the load are predetermined, because pre-registration is used for the access to the resources.

Besides 80% of the resources are requested only within 20% of the service time.

Currently the existing solutions based on cloud services use a universal approach to provide access to resources. Specific services are not taken into account, which leads to inefficient resource use. The goal of our research is to determine the key parameters affecting the performance of distance learning system resources, and optimize their use according to their computational problem solution ability. To implement this approach adequate mathematical models are requires to describe the work of involved services.

II. METHOD OF ACCESS TO MULTIMEDIA EDUCATIONAL RESOURCES FOR USERS OF A DISTANCE LEARNING SYSTEM

The deficiency of the broadband multimedia educational services in distance learning is caused by the limited bandwidth output of a link. This problem is especially important for users who access web applications from the Internet.

To analyze the effectiveness of existing architecture we have developed the level subsystem model for DLS OSU, including the following levels:

Level 1: subsystem of knowledge control;

Level 2: subsystem e-Library;

Level 3: subsystem of broadcasting and publishing video and audio materials.

Each level of the model has different requirements for the applied software and quality of service (QOS). To improve reliability and quality of the network multimedia services implementation of effective methods is required of providing load balancing on hardware and software resources. Having analyzed the usage of each component in the DLS, we have obtained a demand rating for key resources:

1. the communication channel;
2. data storage system;
3. database control system.

Methods, optimizing and improving the efficiency of the service requests from users, are applied to resources represented in the rating. At the same time we should consider the individual characteristics of the selected resources and algorithms of their work to ensure the required quality of service.

Today, the most effective method is predicting customer behavior. Using the mechanisms of pre-registration (subscription for services), as well as statistics on the consumption of resources for each of the subsystems, we can predetermine the amount of required computing power needed to maintain the stream of incoming requests [1].

However, on account of multiple types of requests to different service channels at the same time, effective stream control is essential for the resources of broadband multimedia distance education system. If subsystems are provided Web services, they can be described as queuing systems with a limited queuing time and a Poisson stream requests [4,5]. At the same time, giving the characteristics to each service, we define the method for request stream control (Figure 1) and three phases of request service: the accumulation of requests in the cloud system controller (phase 1), priority request service for selected compute nodes (phase 2), the generation of data packets requested by users (phase 3).

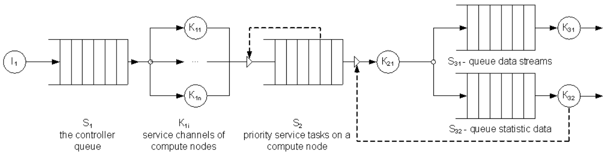
I_1
K_{11}
...
K_{1n}
K_{21}
S_{31} - queue data streams
K_{31}
K_{32}
S_{32} - queue statistic data
S_1
the controller queue
K_{1i}
service channels of compute nodes
S_2
priority service tasks on a compute node

Figure 1. Method for the request stream control in distance learning system

Let us consider each of the phases of the request stream in detail. When you receive a request in a distance learning system, it goes into the queue S1 of the cloud controller for further distribution via $K1_i$, i = 1, ..., n service computing nodes (n being the number of nodes). The number of nodes depends on the current load of the cloud, on the amount of tasks and on the number of copies of each of the subsystems that are running at the moment.

Request enters the queue priority service S_2 after the controller selects available compute node, using an algorithm minimizing the consumption of resources and maximizing customer service. According to the priority service algorithm request gets into channel K_{21} to perform requested computing operations. User requests are sent to one of the several queues: streaming data and static data for the generation and transmission of the processed data packet. Then they are transmitted to service channels K_{21} or K_{32} in compliance with the algorithm of providing access to data in the cloud storage system.

The described algorithm of selection and prioritization of requests to the cloud service model is based on user request model constructed within this research.

III. MODEL OF USER REQUEST SERVICE IN THE BROADBAND MULTIMEDIA EDUCATIONAL RESOURCES

The distinctive feature of cloud computing is the ability to control the order of execution of computational problems using a variety of algorithms for processing queues. It allows to effectively distribute requests in heavy request streams with critical response time. In this research, we propose an algorithm of selection and prioritization to handle incoming requests during the second phase of the service request. The algorithm is based on demands of basic resources at each of the previously identified levels of the subsystems within the DLS.

We can determine the numerical indicators of the basic resources, using the following expression:

$$R_{\text{i use}} = \frac{R_i \cdot 100}{(R_1 + \ldots + R_n)}. \qquad (1)$$

where R_1, ..., R_n are numerical indicators of the resource usage for each of the classification attributes obtained from measurements of the time interval Δ T [7].

Indicators of service priority in the model are based on the resource demand rating for the system in general. The total resource usage of distance learning is defined as the total area U_{sdo} which involves all levels of the model (U_i). So the maximum possibility of the resources of the server is denoted as the area obtained by using 100% of key services (Figure 2).

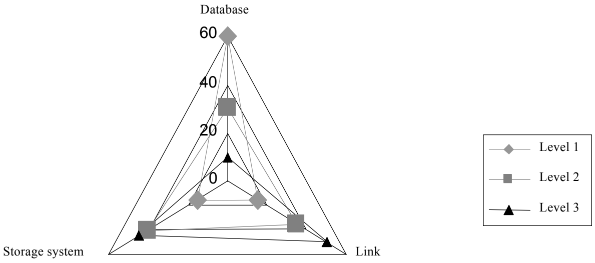

Figure 2. Diagram for the priorities of resource requests

As the operation of the system is carried out continuously, the receipt of requests to system resources can be described in discrete time:

I_j (T_j) = {j: t (0, T_i)} – the set of request numbers coming in within the time interval (0, T_i) to the subsystem i (i - level subsystem, i = 1, ..., M).

Processing status of the j-th request submitted to the i-th level is denoted x_{ij}; denial of service shall be assumed as $x_{ij} = 0$, success as $x_{ij} = 1$.

The intensity of the receipt and processing of applications for each level of the model is denoted λ_i, in which case it depends on the resource requirements of the subsystem. Besides, an index of priority (P_i) shall be introduced for each of the levels, its distribution depending on the quantity of simultaneously resources. Thus the load at each level can be limited by the following:

$$\sum_{I_j(T_j)} U_i x_{ij} \le H_i \text{ , } i=1,\ldots,M \qquad (2)$$

Specifying the objective function we introduce the following restrictions related to the subject area of this study:

– processing time (T) of any request is limited;

– server capacity (H) is fixed.

Due to the uneven use of basic resources at each level of a distance learning system, we should determine the condition of maximum load on each computing node of clouds when all the operations on all running instances of the application are reliable:

$$\sum_{i=1}^{M} \sum_{j \in I_j(T_j)} U_i x_{ij} \le H \text{ , } x_{ij}=\{0,1\} \qquad (3)$$

Thus, to handle the maximum number of user requests per unit of time we get the following objective function:

$$\sum_{i=1}^{M} \sum_{I_j(T_j)} \lambda_i x_{ij} P_i \to \max \qquad (4)$$

While setting priorities we evaluate the following request features: queuing time, current request queue length, request intensity for each of the components of a resource.

Selection of priorities and assessment of the current resource intensity is based on the resource components with individual thresholds related to the physical limits of the equipment. In this research, have modeled request service in a hybrid cloud system within the above described model. We have used the algorithm of prioritizing user requests based on the assessment of demand for resources. This has helped to improve the efficiency of the system components by reducing the queue time, which has resulted in a reduction of its length. The result is a 12-15% growth in the number of served requests as compared to the standard means of queue processing within the research period Δ T = 60 seconds (Figure 3).

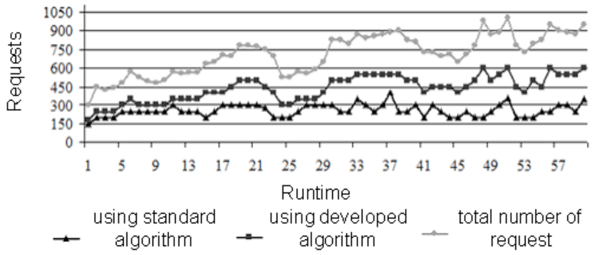

Figure 3. - Diagram of service requests in the hybrid cloud system using the priority service algorithm

In addition, during the research we have found that the common point of aggregation is the data storage system (NAS), which provides the processing of requests from consumers of multimedia educational services. Consequently, the effectiveness of the whole system of distance learning, as well as the quality of service depends on the performance of the data storage. Therefore, to control the request streams effectively we must develop a model of access to multimedia data storage in a hybrid cloud system.

IV. DATA ACCESS MODEL FOR HYBRID CLOUD SYSTEM STORAGE

The key differentiator of multimedia data storage is the heterogeneity of information (text, audio, video or data), and consequently different approaches are used to provide access to it. Besides the data access methods the request intensity for different elements is also crucial, which can be evaluated with the use of intersystem identification algorithms, which in its turn allows to estimate and forecast the load on the storage system. In this regard, an important aspect of system resource control leading to a significant increase in the number of simultaneous requests is a competent device organization of data element posting and distribution [2,3].

A distinctive feature of cloud storage is the reconfiguration of the structure depending on the resources consumed. This in its turn enables to implement optimization algorithms in terms of placement of data within the disk space, and control the change in the number of the system's devices. The process of optimizing the placement should not lead to lower customer service quality, for which we should take into account the capacity of the network and the maximum amount of data that can be transmitted at one time [4]. Also the current load on the devices themselves must be taken into account, as well as their positional relationship with each other and the clients connected to them.

To optimize data access mechanisms a general model of access to data storage should be built.

Let R = (U, M, Q), where $U = \{u_1, u_2, ...\}$ – is the set of users, $M = \{m_1, m_2, ...\}$ - is the set of unique data elements placed in storage devices. The minimum unit of data we assume as m_i – as a file with mandatory property h - size.

To ensure safety of storage and load balancing between storage devices we define the distribution function of data elements as

$$M_c = \{ m_1^{j_1}, m_1^{j_2}, m_1^{j_3}, \ldots, m_2^{j_1}, m_2^{j_2}, m_2^{j_3}, \ldots \},$$

where – k is a placed copy of the data element (m_i) to the j_k storage device, provided that $k \geq 3$ (at least three copies of the smallest unit of storage on different devices).

Then the distribution of data items for storage devices becomes P: $M_c \rightarrow D$. Based on the above, we write the user requirements for data items. Q: $U \rightarrow X \subseteq M_c$, where X – is a set of data requested by set of users U. Then, the data storage can be written as tuple S = (M_c, D, P, L, C, R, G), where:

$D = \{d_1, d_2, ...\}$ - set of storage devices;

$L = \{l_1, l_2, ...\}$ - the set of values characterizing the load of each storage device (the number of concurrent user requests to a specific device);

$C = \{c_1, c_2, ...\}$ - the set of values that characterizes the amount of each of the devices in the storage;

$G \in N$ - natural factor specifying the geographical (topological) priority of the storage.

Typically, for large cloud structures they use consolidated storage, consisting of farms joining multiple repositories in a single array. Consider it as $S_{farm} = \{S_1, S_2, ...\}$.

As user requirement characteristics change in time we transform the requirement tuple R (t) = (U, M_c, Q (t)). Then Q (t): $U \rightarrow X \subseteq M_c$ – user requirements for data elements that change over time. Taking into account the storage properties we define the tuple according to time S (t) = (M_c (t), D (t), P (t), L (t), C, R (t), G), where D (t) = $\{d_1, d_2, ...\}$ is the set of storage devices changing over time so that $\forall$ t, D (t)> 0; P (t): $M_c \rightarrow D$ is the distribution function of the data items for storage changing over time.

To optimize the use of hardware resources and reduce the number of simultaneously used devices we introduces relation tuple S_{cloud} (t) = {S (t), D (t), D_{use} (t)}, where $\forall$ t, D_{use} (t) $\subseteq$ D (t) is a set of storage devices used in scalable storage S at time t. Besides while scaling the storage and data migrating the condition $\forall$ t, i, j≠i $\Rightarrow D_i (t) \cap D_j (t) = 0$ must be true, which means the data storage should not use the same device while migrating. This guarantees the speed of information processing and ensures acceptable reconfiguration time.

Thus to minimize the number of simultaneously used storage devices within one scalable storage and maximize the number of processed requests per user we introduce the following objective function:

$$\sum_{i=1}^{N} P_i(t) \rightarrow \min \qquad (5)$$

$$\sum_{i=1}^{N} L_i P_i(t) \mathrm{R}_i(t) \rightarrow \max \qquad (6)$$

Using the data access storage model we have developed an algorithm to balance the load between devices. This algorithm was implemented in program module for Swift component of OpenStack cloud system. This cloud system was chosen due to its open architecture and the modifiability. The main disadvantage of OpenStack is its inefficient algorithm for task distribution between the data storage nodes. This algorithm does not consider some properties of LAN, virtual machines and data storages which affects the response time of both the cloud system and running application instances. Generally data distribution algorithms for cloud system storages result in inefficient information storing and network access to the requested data [7].

We have developed the algorithm to reduce the response time using the topology and routing of data streams. So this algorithm implies flexible storing control which reduces computing resources for data and virtual machine migration [8,9].

For algorithm efficiency evaluation we have determined the following varying parameter conformities of system functioning. An increase in the number of data copies leads to a significant load reduction on storage devices. However, the number of involved devices grows which does not correspond to our goal. Simultaneous access to several devices containing requested data of different sizes may result in a storage performance imbalance and consequently deny of service.

In case multiple access to the same data the devices containing popular data fail to serve the request because of the load imbalance between nodes. The storage caching algorithms prove in efficient.

The proposed algorithm allows to correct the above shortcomings of the storage control system, which provides performance growth in the cloud up to 5-9% against the standard OpenStack storage control system (Fig. 4).

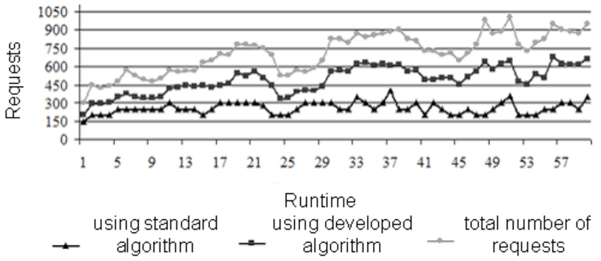

Figure 4. Diagram of service requests in the hybrid cloud system using intelligence data distribution algorithm

V. CONCLUSION

To evaluate the effectiveness of the developed OpenStack module as a load balancer, a comprehensive simulation of hybrid cloud system has been performed taking into account the features of distance learning system.

Load forecasting and queue prioritization enables cloud scaling control so reducing the resources involved. And the use of optimization algorithms in the storage system enables efficient access, regardless of the number of requests and types of data. As a result of integrated modeling of request stream service there was an output increase of 14% to 21% compared to the standard which is highly effective at high request intensity. Moreover, the developed optimization models can be used for further study of hardware and software resource effectiveness to improve QoS in distributed distance-learning systems, and for the development of multimedia resources in general.

REFERENCES

[1] I. Bolodurina, V. Reshetnikov, D. Parfenov; "Allocation of resources in the information system of remote support of the educational process" // Software products and systems. - 2012. - № 3. - pp. 151-155.

[2] D. Petrov, "Optimal algorithm for data migration in a scalable cloud storage" // Managing large systems. 2010. - №. 30 - pp.180-197.

[3] D. Petrov, A dynamic model of a scalable cloud storage data // Proceedings of the LETI. 2010. - № 4 - S. 17-21

[4] O. Gusev, A. Zhukov,V. Polyakov, S. Polyakov; The problem of adequate assessment of the performance of Web servers in corporate networks in enterprises PPI // Proceedings of the 6th Scientific Conference "New Information Technologies in the pulp and paper and energy." / Petrozavodsk, 2004. - S. 84-87

[5] A. Zhukov, Some of the models of optimal control input of applications in the intranet systems. // Proceedings of the 6th Scientific Conference "New Information Technologies in the pulp and paper and energy." / Petrozavodsk, 2004. - S. 87-90.

[6] I. Boychenko, S. Korytnikov; Resource management in service-oriented systems, such as "application as a service" // Proceedings of the Tomsk State University of Control Systems and Radio Electronics, Vol. 1-2, 2010. - S. 156-160.

[7] V. Tarasov, P. Polezhaev,A. Shukhman, J. Ushakov, A. Konnov; Mathematical models of cloud computing data center using OpenFlow // Bulletin of the Orenburg State University. - 2012. - № 9. - C. 150-155.

[8] D. Parfenov, Algorithm Comparison of dynamic data distribution in hybrid cloud systems of distance learning // Control Systems and Information Technology, № 4.1 (50), 2012. - S. 163-168

[9] D. Parfenov, Algorithm Comparison of dynamic data distribution in hybrid cloud systems of distance learning // Information Technology modeling and management, № 6 (78), 2012. - S. 491-498

[10] OpenStack Open Source Cloud Computing Software. [Electronic resource]. - Mode of access: http://www.openstack.org/

AUTHORS

I. P. Bolodurina is head of Department of Applied Mathematics, Orenburg State University, Orenburg, Russia (e-mail: prmat@mail.osu.ru).

A. E. Shukhman is head of Department of IT Management, Orenburg State University, Orenburg, Russia (e-mail: ais@mail.osu.ru).

D. I. Parfenov is leading expert of Department of distance learning, Orenburg State University, Orenburg, Russia (e-mail: fdot_it@mail.osu.ru).

Authors thank for support the Ministry of Education and Science of Russian Federation (projects 14.B37.21.1881 and 14.132.21.1801 by the federal target program "Scientific and scientific-pedagogical personnel of innovative Russia") and Russian Foundation for Basic Research (project 13-07-00198).

This article is an extended and modified version of a paper presented at the EDUCON2013 conference held at Technische Universität Berlin, Berlin, Germany from March 13-15, 2013. Submitted 04 June 2013. Published as re-submitted by the authors 11 October 2013.