Load Balancing Strategy for Hybrid Cloud-based Rendering Service

G. Vilutis[1], K. Sutiene[2], R. Kavaliunas[1], L. Daugirdas[1]
[1]*Department of Computer Network, Kaunas University of Technology, Studentu St. 50–416, LT-51368, Kaunas, Lithuania*
[2]*Department of Mathematical Research in Systems, Kaunas University of Technology, Studentu St. 50–222, LT-51368, Kaunas, Lithuania*
gytis.vilutis@ktu.lt

***Abstract*—This paper focuses on SaaS-type Hybrid Cloud architecture, where computing resources in Private Cloud are limited in quantity. In this case, the workload balancing problem must be solved, since some projects are lost because of under provisioning of Private Cloud resources, as well as costly resources are wasted during nonpeak periods. The workload balancing algorithm for Hybrid Cloud is proposed, assuming that a certain part of incoming projects in the workload can be postponed for a short time period. This algorithm is adapted for a rendering service to ensure its Cloud-based delivery. The cash flow model is built to determine the adequate quantity of computing resources in Private Cloud if the demand of resources is variable and has some pattern in time. The implementation of a proposed strategy for workload balancing is practically worth, since the same flow of incoming projects is serviced with a smaller amount of own computing resources, thus reducing their downtime. For this purpose, an experimental study of the expected effect is presented.**

***Index Terms*—Cloud, load flow control, computer networks.**

I. INTRODUCTION

A key feature in Cloud computing is an improved peak-load handling and dynamic resource provisioning without need to set up a new software or hardware infrastructure in every location. However, the owner of such environments must solve workload balancing problems, as well as the challenge of determining the optimal number of resources required depending on workload in time. Within this context, the resource allocation management problem is considered in this paper, mainly for SaaS platform deployments. The work is motivated by the fact, that in practice the estimates of average server utilization range from 5 % to 20 % but for many services the peak workload exceeds the average by factors of 2 to 10. Thus, it is a big challenge to make sure that all users will receive their service and the downtime of private resources will be minimized [1], [2].

Cloud service owner usually faces two problems (Fig. 1):

- to deploy the maximum quantity of resources (Fig. 1, Max line), wishing to satisfy all its users' requests but cost-effective scalability is not achieved because of idle processes and resources during nonpeak periods;
- to keep the minimum quantity of resources (Fig. 1, Min line) in full usage even if the users' load is at the minimum level but the revenues from potential customers are lost if the quantity of servers is too low.

Manuscript received July 18, 2013; accepted October 2, 2013.

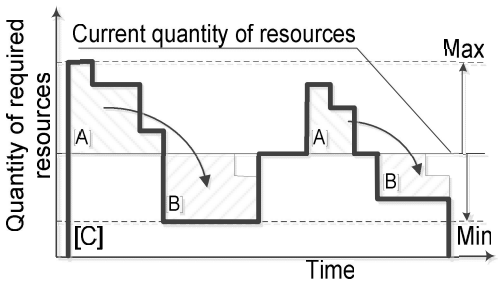

Fig 1. Allocation of projects in time.

II. THE CONTRIBUTION OF THIS PAPER

This paper is focused on projects management model, classifying them into two categories: users that send projects of high priority (HP) and pay a full service price (with a guarantee that the service will be fully granted), and users that send projects of low priority (LP) and pay a minimal service price (with a certain probability that the service may not be received) [3]. It is clear that such management model does not ensure that a certain part of LP projects will be serviced.

The contribution of this work is to present the workload balancing approach which ensures that both HP and LP projects will be carried out. All projects are arranged in a certain queue which is serviced according to the method presented in the paper. The main idea is depicted in Fig. 1: sector [A] denotes LP projects which executions are postponed for later time during workload balancing, since they cannot be serviced at the moment due to the low number of available resources that all are busy at the current moment; sector [B] denotes the execution of LP projects that have been postponed for a certain time period from sector [A]. The concept of this approach implies that all LP projects that cannot be serviced at the current moment are postponed for later time when the users' demand for SaaS type service will be decreased. Sector [C] denotes both HP and LP projects which are serviced at the current moment due to a sufficient amount of resources.

The workload balancing strategy to be considered in this paper is based on the scheduled reservations for future

requests of resources. The reservation is performed by forecasting model used to predict future workloads. It should be noted that forecasting is reasonable since users usually do have resource usage patterns or periodical load behaviour [4], [5], and it helps to plan and optimize balancing decisions.

Some of researches [6]–[8] focus on resources management strategy based on Service Level Agreement (SLA) of virtual machine to provision resources. As argued in [9], resource management approach based on SLA lacks flexibility, because virtual machines may not require as many resources as defined by the SLA, causing the waste of Cloud resources. Second, the load imbalance of the whole system is the reason of too large number of different types of virtual machines. Thus, model of load balancing and scheduling to be developed in this paper will be based on the workflow prediction in Cloud that devotes virtual machines to users.

The current quantity of resources has to be managed in an effective manner. Thus, the global utility function as an objective is constructed in order to maximize Cloud service owner's profit. The paper [10], [11] proposes that utility functions, combined with optimization algorithms that seek to maximize utility for a workload based on given certain resources, may provide an effective paradigm for managing workload execution in Cloud computing. Considering another aspect of this problem, the optimal quantity of resources can be computed based on the future workload forecasts. Such idea is applied in this paper.

We posit that the approach to be proposed for balancing workloads, consisting of HP and LP projects, will allow decreasing the quantity of current resources as well as ensuring the service for all users.

For this purpose, the expected effect will be evaluated by simulation. The simulation results to be obtained will quantify the performance of proposed strategy before real system design experiments.

III. A CASE STUDY

In this paper, the project rendering service will be considered as an instance of SaaS type service. Rendering service requires many computing resources and can be easily processed in parallel by matching one video frame to a server. The users of video rendering service are liable to wait for a certain time period, especially when big video projects must be rendered, i.e. they are willing to get many resources for their projects later but not to render them on their own computers. The priority system is also used in academic networks. Such network is usually comprised of several organizations. HP is assigned to the projects that are received from the members of the organization, which owns the cloud.

IV. DETERMINING THE ADEQUATE QUANTITY OF RESOURCES IN PRIVATE CLOUD

To ensure the delivery of SaaS service, the architecture of Hybrid Cloud is selected since it allows resending projects to Public Cloud if resources of Private Cloud are fully loaded.

It is important for the owner of Private Cloud to determine the adequate quantity of resources as it was shown in Fig. 1. Since the owner aims to gain the profit, its obtained utility $U(t)$ is maximized based on the equation

$$U(t,N) = R^{PB}(t) + R^{PR}(t) - C^{PR}(t,N) \to \max, \quad (1)$$

with constraint $C^{PB} \geq P_{HP}^{PR} > R_{LP}^{PR} > C^{PR}$, where:

$$R^{PR}(t) = i_{LP}^{PR}(t) \cdot P_{LP}^{PR} + i_{HP}^{PR}(t) \cdot P_{HP}^{PR}, \quad (2)$$

$$R^{PB}(t) = i_{LP}^{PB}(t) \cdot \left(P_{LP}^{PR} - C^{PB}\right) + i_{HP}^{PB}(t) \cdot \left(P_{HP}^{PR} - C^{PB}\right), \quad (3)$$

$R^{PR}(t)$ – earnings gained using Private Cloud resources, computed based on (2); $R^{PB}(t)$ – earnings gained from renting resources in Public Cloud, given in (3); $C^{PR}(t,N)$ – maintenance costs of Private Cloud resources, as time-dependent function from quantity of working stations N; $i_{LP}^{PB}(t)$ – a number of LP projects executed in Public Cloud per t; P_{LP}^{PR} – price of LP project settled by the owner of Private Cloud; C^{PB} – price paid by the owner of Private Cloud for rented resources in Public Cloud; $i_{HP}^{PB}(t)$ – a number of HP projects executed in Public Cloud per t; P_{HP}^{PR} – price of HP project settled by the owner of Private Cloud; $i_{LP}^{PR}(t)$ – a number of LP projects executed in Private Cloud per t; $i_{HP}^{PR}(t)$ – a number of HP projects executed in Private Cloud per t.

Equations (1) allows determining the adequate resources, defined through the number of working stations N^*.

V. CONCEPT OF LOAD BALANCING

As it was considered, another but also very important goal of SaaS service owners is – to service all projects sent by users. It was noticed that the dynamics of users' demand has some pattern in time (periodical variations). Thus, it is reasonable to postpone projects for later time (Fig. 1 from sector [A] to [B]). Only some part of projects that are of LP can be postponed for a certain time Δt (max 4 hours). In the following, the algorithm will be presented which ensures that all received projects during time period Δt will be started to execute, in some special cases resending them to Public Cloud.

The result of such projects balancing method is depicted in Fig. 2. Based on this, one LP project and one HP project are forwarded to Public Cloud, as all own resources (Private Cloud) are busy. LP project is sent to the Public Cloud, since the postponement time Δt has ended. In Fig. 2 Δt is equal to two time intervals. HP project is sent to the Public Cloud, as there are more HP projects than Private Cloud can service at the current time moment. In Fig. 2 depicted effect will be experimentally evaluated for different postponement time Δt.

Workload balancing method, which includes optimization component (1-3), reserves resources in advance to achieve the maximum service for HP projects, while LP projects can be postponed for later time. The reservation is organized based on users' demand forecasting.

VI. FORECASTING OF WORKLOAD DYNAMICS

The workload balancing strategy presented in this paper includes the forecasting module which is responsible for predicting the future resource usage according to variability patterns in client workload. Workload statistical data constitutes a time series, which is characterized by trends, seasonality, and peaks.

Prediction methods used to improve the utilization of Cloud computing resources can be grouped into two categories, namely statistical model based approaches and artificial intelligence based algorithms [12].

Since this paper is focused on workflow analysis and forecasts, the main attention is paid to time series based forecasting methods. Thus, the fitted time series model is accompanied by a hotspot detection algorithm to model the sudden peaks in the forecasts of resource usage [13], [14]. This approach is applied for forecasting of HP projects dynamics, since the algorithm of workload balancing to be presented in this paper requires the reservation of resources for HP projects only.

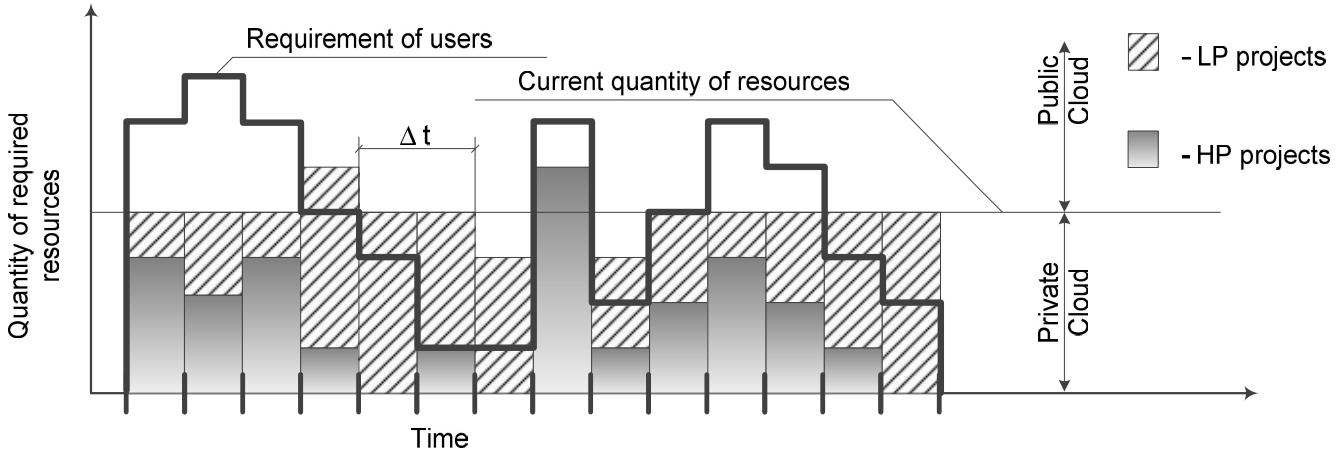

Fig. 2. Projects balancing between Clouds.

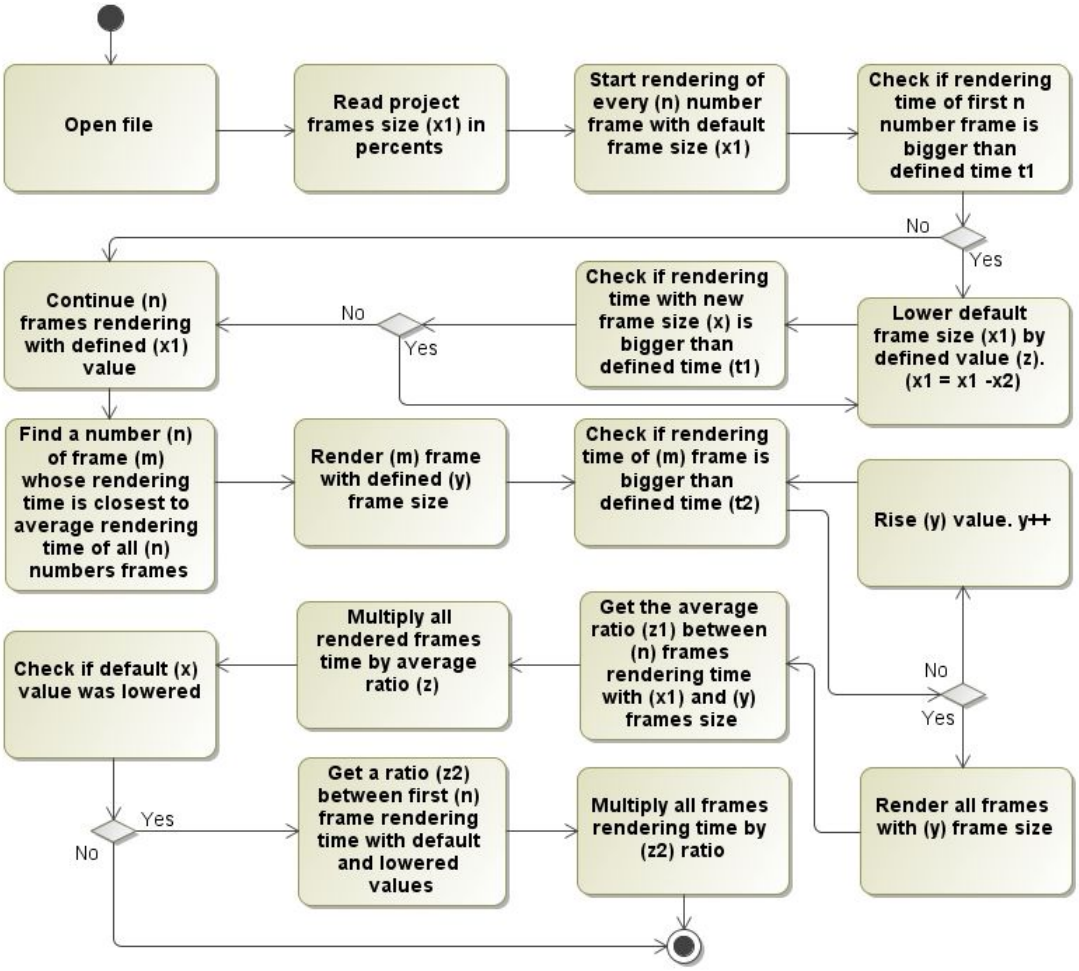

Fig. 3. Algorithm for predicting the rendering duration.

VII. METHOD FOR COMPUTING THE RENDERING DURATIONS

Forecasts are also computed for the durations of video frame rendering. To know a service time required for the whole project is only possible by rendering every frame. The main idea of our prediction algorithm is given in Fig. 3. Based on captured project rendering files, statistical relationships among rendering times for different video resolutions are estimated based on correlation coefficient. The obtained results showed that the frame's rendering time of current size can be predicted using the same frames only of lower resolution. Inaccuracies are significant when the resolution reaches 0.1 of original size or the video is stationary.

VIII. MAIN WORKLOAD BALANCING ALGORITHM

The purpose of an algorithm (Fig. 4) is to balance the projects to be rendered between Private and Public Clouds with minimum usage of Public Cloud and also to ensure that all projects will be started on time. In the rendering system, this algorithm starts each minute if the past instance is not running. All projects are rendered in Cloud clusters. The size of cluster can be, for example, 10-20 servers. Any cluster in the system can be in one of three states:

– Free state – means that a cluster is ready to render any project;
– Reserved state – means that a cluster is free but is waiting for a predicted HP project to be rendered. LP project cannot be rendered at this cluster;
– Busy state – means that a cluster is currently rendering a project and will not accept any other project until its state will be changed.

Main steps of the algorithm are explained further.

Reservation. Workload balancing model uses algorithms of forecasting workload dynamics and rendering durations to reserve clusters in Private Cloud for HP projects.

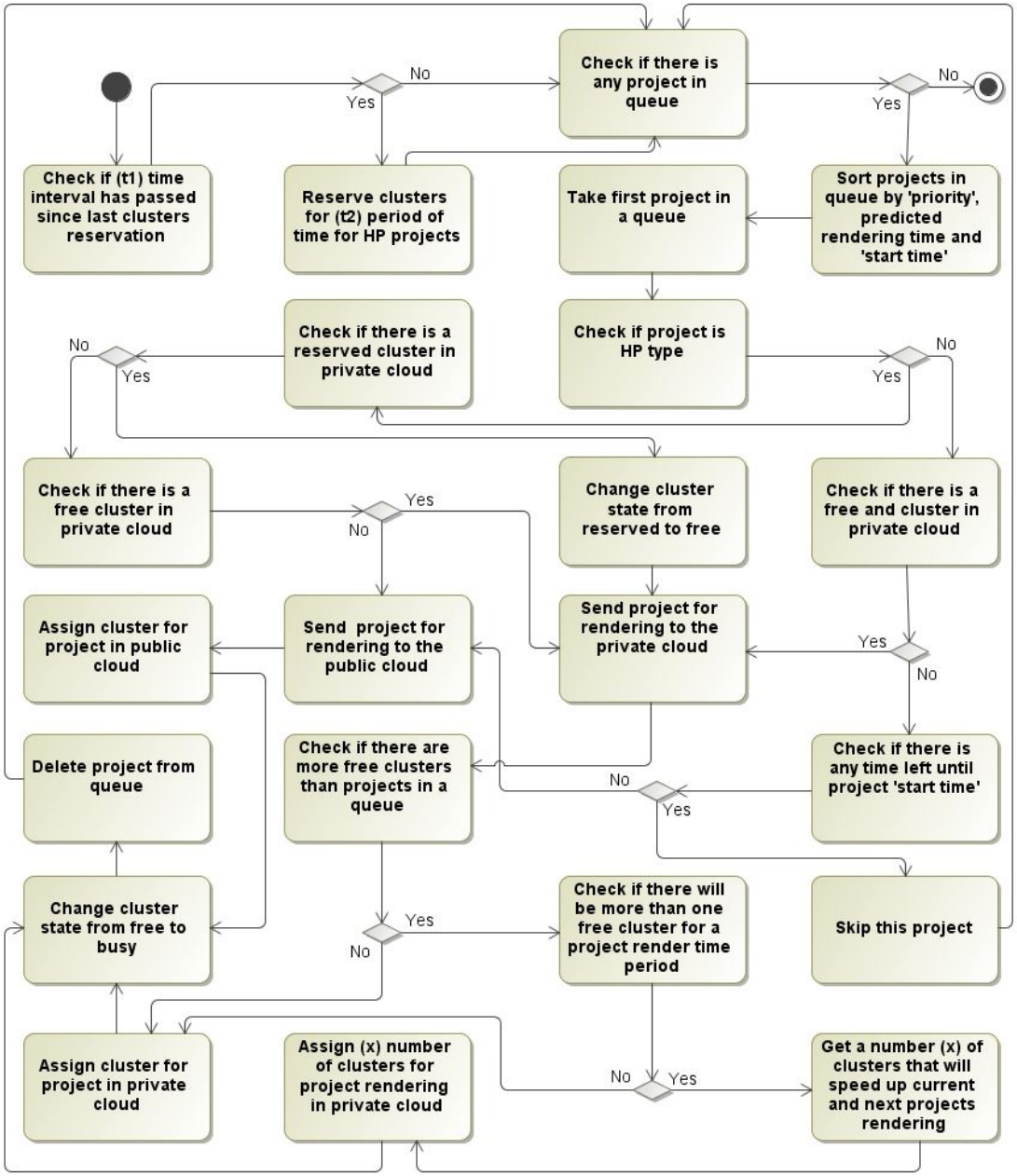

Fig. 4. Algorithm for rendering load balancing in Cloud system.

Based on historical observations, a cluster is reserved for each HP project for an average project rendering time duration. Let define t1 as time period which describes how frequent reservation is run, and t2 – as reservation period.

Queue sorting. It is checked if there are any projects in the queue. If there are, firstly, HP projects are assigned to clusters. HP projects are arranged by forecasted rendering time (in a decreasing order). This lets us assure that the longest HP projects will be rendered in Private Cloud. Secondly, LP projects are sorted by the time moment they must be started. The start time of LP project is computed as incoming time of the project plus a period of time that the project can be postponed (this period is declared in system settings and depends on system size). So if all clusters in Private Cloud are in states 'busy' or 'reserved' the LP project can be delayed until its start time is reached. If more than one LP project must be started at the same, moment, they are arranged by predicted rendering time (the same as HP projects). If start time of LP project becomes equal to the current time, then LP project becomes HP and is sorted with others HP projects.

Reservations influence. HP projects may have reserved certain clusters in Private Cloud. By this rule, LP projects cannot occupy reservations even if there is no HP task at the moment (it is presumed that predicted HP project may arrive next minute).

Balancing between Clouds. HP projects must be started without a delay. HP project is sent to the Public Cloud immediately if there is no reservation or a free cluster in the Private Cloud. LP project can be delayed a defined period of time to get a cluster in the Private Cloud. LP project becomes HP if waiting time ends.

Clusters allocation if there is no load in the Private Cloud. Every time when a project is sent to the Private Cloud, it is checked if there are more free clusters than projects in a queue. During this check, the estimation of a number of projects that will arrive in future is performed and forecasting of their rendering time is done to get the exact number of possible free clusters in the future. In the case of free clusters to be available, the project is devoted to additional clusters. The amount of clusters depends on a number of waiting projects and free clusters. All projects in a queue get an equal number of additional clusters starting from the first in a queue until there is some free additional cluster left.

IX. EXPERIMENT BY SIMULATION

The goal of simulation is to demonstrate the advantages of the workload balancing for rendering service presented in this paper if it is assumed that a certain part of incoming projects can be postponed for a short time. The research focuses on the estimating the adequate quantity N^* of resources in Private Cloud while matching them to a workload behaviour in time. The owner of Private Cloud aims to maximize its profit, which relates to the maintenance costs $C^{PR}(t,N)$, depending on the amount working stations N.

Functional dependency between resources and profit is depicted in Fig. 5. The figure is obtained from simulation, where the workload of projects has been predicted based on statistical data, and the current amount of working stations N has been increased by some step.

In Fig. 5, profit is growing while the amount of resources N is approaching value N^* from the left, since all working stations in Private Cloud are fully loaded. If the number of resources $N > N^*$, the obtained utility for the owner of Private Cloud starts to decrease. This is determined by the fact that certain time periods occur when all resources or part of them are not employed.

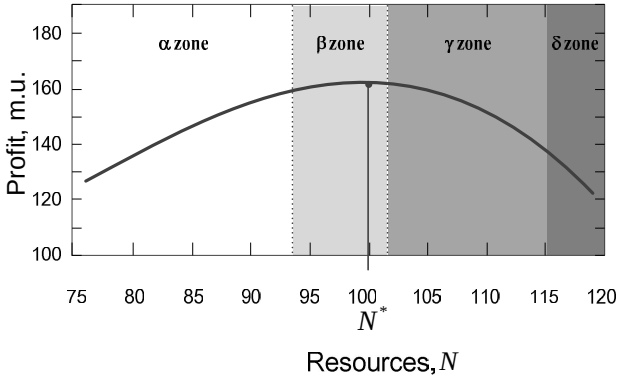

Fig. 5. Profit dependency from resources.

The simulation was performed for a fixed time horizon having the workload of a certain pattern. The ~ , , ¥ , and ¢ zones have been distinguished in conformity with different amount of resources. These zones are depicted in Fig. 5 and are described as follows:

~ zone is characterized with insufficient amount of resources, since a certain part of projects is resent to Public Cloud based on the algorithm presented in this paper. All available resources are fully loaded;

zone denotes the recommended amount of resources, when all incoming projects are serviced in Private Cloud. There exist no time periods when resources are in a 'free' state during all time horizon to be modelled;

¥ zone is characterized with short-run occurrences when all resources are free;

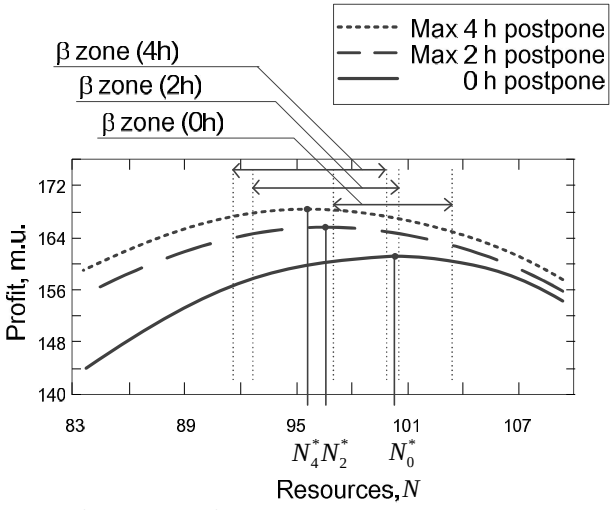

Fig. 6. Profit dependency from postponement time.

¢ **zone** denotes a situation when a certain part of resources are always in a 'free' state. It means that available

resources are not fully employed. This is caused by an excessive amount of current resources.

In general, the owner of Private Cloud can earn more profit with lower amount of available resources if users agree to wait till their LP projects will be started to execute. Figure 6 depicts functional dependencies between amount of resources and profit under different delays for LP projects. In this figure the curve with 0 h postponement time matches the curve from Fig. 5, and notation N_0^* coincides with N^*.

Thus, the owner of cloud-based rendering service gains more profit by proposing for users the lower service price P_{LP}^{PR} for LP projects. Figure 6 allows comparing the behaviour of profit's function for different postponements to be allowed on LP projects. It can be seen that the optimal number of resources has been decreased from N_0^* till N_4^*, as well as zone has been expanded if higher postponement time for LP projects is available. Broader limits of zone allows for the owner of Private Cloud to service the incoming workload in more flexible way.

X. CONCLUSIONS

The performed research can be summarized with following conclusions.

A novel workload balancing strategy for a Hybrid Cloud is proposed. This algorithm allows improving the utilization of computing resources in Private Cloud according to variability patterns in projects' workload.

The paper presents a profit making components important to the owner of Private Cloud. It is determined that one of the main components is $C^{PR}(t, N)$, which depends on the optimal value N to be computed.

The simulation has shown that the application of the proposed algorithm can increase the profit of the owner of Private Cloud due to the following factors: the broadening of zone caused by higher postponement time for LP projects; the optimal quantity of resources to be selected, and the more evenly distributed workload.

Simulation results ensured the benefit of proposed load balancing strategy for hybrid Cloud-based rendering service. In future, this research will continue with strategy-based software developments in real world Cloud environment.

REFERENCES

[1] B. Ambrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, "A view of cloud computing", *Communications of the ACM, New York: Association for Computing Machinery*, vol. 53, no. 4, pp. 50–58, 2010. [Online]. Available: http://dx.doi.org/10.1145/1721654.1721672

[2] H. Song, J. Li, X. C. Liu, "IdleCached: an idle resource cached dynamic scheduling algorithm in cloud computing", in *IEEE 9th Int. Conf. on Autonomic and Trusted Computing*, Fukuoka, Japan, 2012, pp. 912–917. [Online]. Available: http://dx.doi.org/10.1109/UIC-ATC.2012.24

[3] P. Marshall, K. Keahey, T. Freeman, "Improving utilization of infrastructure clouds", in *Proc. of 11th IEEE/ACM Int. Symposium on Cluster, Cloud and Grid Computing*, Newport Beach, CA, 2011, pp. 205–214. [Online]. Available: http://dx.doi.org/10.1109/CCGrid.2011.56

[4] E. C. Withana, B. Plale, "Usage patterns to provision for scientific experimentation in clouds", in *Proc. of IEEE 2nd Int. Conf. Cloud Computing Technology and Science*, Indianapolis, 2010, pp. 226–233. [Online]. Available: http://dx.doi.org/10.1109/CloudCom.2010.8

[5] H. Zhang, G. Jiang, K. Yoshihira, H. Chen, A. Saxena, "Intelligent workload factoring for a hybrid cloud computing model", in *Proc. of World Conf. on Services-I*, Los Angeles, CA, 2009, pp. 701–708. [Online]. Available: http://dx.doi.org/10.1109/SERVICES-I.2009.26

[6] S. Sakr, A. Liu, "SLA-based and consumer-centric dynamic provisioning for cloud databases", in *Proc. of the 5th IEEE Int. Conf. Cloud Computing*, Australia, 2012, pp. 360–367. [Online]. Available: http://dx.doi.org/10.1109/CLOUD.2012.11

[7] K. S. Patel, A. K. Sarje, "VM provisioning method to improve the profit and SLA violation of cloud service providers", in *Proc. of IEEE Int. Conf. on Cloud Computing in Emerging Markets*, Roorkee, India, 2012, pp. 1–5. [Online]. Available: http://dx.doi.org/10.1109/CCEM. 2012.6354623

[8] M. A. Bochicchio, A. Longo, "Modelling Contract Management for Cloud Services", in *Proc. IEEE Int. Conf. Cloud Computing*, Washington, USA, 2011, pp. 332–339. [Online]. Available: http://dx.doi.org/10.1109/CLOUD.2011.102

[9] Z. Zhang, H. Wang, L. Xiao, L. Ruan, "A statistical based resource allocation scheme in cloud", in *Proc. IEEE Int. Conf. on Cloud and Service Computing*, Beijing, China, 2011, pp. 266–273. [Online]. Available: http://dx.doi.org/10.1109/CSC.2011.6138531

[10] N. W. Paton, M. Aragao, K. Lee, A. Fernandes, R. Sakellariou, "Optimizing utility in cloud computing through autonomic workload execution", *IEEE Data Engineering Bulletin, USA: IEEE Computer Society*, vol. 32, no. 1, pp. 51–58, 2009.

[11] S. Di, C. L. Wang, "Dynamic optimization of multiattribute resource allocation in self-organizing clouds", *IEEE Trans. on Parallel and Distributed Systems*, vol. 24, no. 3, pp. 464–478, Mar. 2013. [Online]. Available: http://dx.doi.org/10.1109/TPDS.2012.144

[12] E. C. Withana, B. Plale, "Usage patterns to provision for scientific experimentation in clouds", in *Proc. IEEE 2nd Int. Conf. Cloud Computing Technology and Science*, Indiana, USA, 2010, pp. 226–233. [Online]. Available: http://dx.doi.org/10.1109/CloudCom.2010.8

[13] N. Roy, A. Dubey, A. Gokhale, "Efficient Autoscaling in the Cloud Using Predictive Models for Workload Forecasting", in *Proc. IEEE Int. Conf. on Cloud Computing*, Washington, DC, 2011, pp. 500–507. [Online]. Available: http://dx.doi.org/10.1109/CLOUD.2011.42

[14] M. Lassnig, T. Fahringer, V. Garonne, A. Molfetas, M. Branco, "Identification, modelling and prediction of non-periodic bursts in workloads", in *Proc. 10th IEEE/ACM Int. Conf. on Cluster, Cloud and Grid Computing*, Melbourne, Australia, 2010, pp. 485–494. [Online]. Available: http://dx.doi.org/10.1109/CCGRID.2010.118