Journal of Theoretical and Applied Information Technology
15th November 2016. Vol.93. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DEVELOPING OF THE SOFTWARE IMPLEMENTATION OF A SIMULATION MODEL OF RESOURCE CONSUMPTION BY A CLOUD CONTACT CENTER DEPENDING ON THE NUMBER OF USERS

[1]**ANTON PAVLOVICH TEYKHRIB**
[1] Company Naumen (Nau-Service)
E-mail: [1]ateyhrib@naumen.ru

ABSTRACT

In the article, software implementation of a simulation model of resource consumption by a cloud contact center depending on the number of users has been made. Methods/Analysis: For modeling was chosen multi-agent system approach and simulation was conducted in the simulation system *Repast Simphony*. Further, a number of simulation experiments verifying the model have been conducted, as well as a forecast of the cloud contact center load in case of adding 200 new users. Findings: Main results shows that developed model is corresponding to real system and additional users will not exceed computational power of current hardware, so no additional investments in infrastructure should be made. Novelty/Improvement: Computational resource load multi-agent system model for cloud contact center was presented for the first time.
Keywords: *Discrete Multi-Agent Simulation, Cloud Contact Center, Server Load, Repast Simphony.*

1. INTRODUCTION

The growth of the number of users is a primary metric of any cloud service development. It raises the problem of cloud service availability to an increase in new users in terms of infrastructure capacity to service them. To determine the capabilities, it is necessary to build a forecast of infrastructure utilization, depending on the user flow. In this article, for predicting, software implementation of a model developed in [1], has been performed in the Repast Simphony simulation system [2], followed by simulation experiments. The integration is one of the leading trends nowadays, and as shown in [3] the need to integrate different systems and applications is often prominent for satisfying business requirements and needs. For modeling was chosen cloud service for internet communications which is industrial partner of this research. This service provides cold calling automation and its performance totally depends on user's activity in calling, i.e. each user's call consumes some computational resources. Therefore, after integration cloud service will receive new users from the system with which integration was performed. Modern simulation modeling assumes implementation of one of the following approaches based on the modeling purposes and task to be solved [4]:

1. System dynamics. System dynamics considers the system as a totality of variables, connected by feedback loops [5].
2. Discrete event simulation. Discrete event simulation describes the system as a hierarchical sequence of elements and operations [6].
3. Agent-based modeling. Agent-based modeling describes a system as a totality of complex objects (agents), connected by certain rules [7].

All the above-mentioned approaches can be used to achieve the simulation purposes, but as shown in [8] the best choice nowadays in agent-based modeling. During this article, we developed the software implementation of a simulation model of resource consumption by a cloud contact center depending on the number of users. This model will be applicable in most contact centers and can be used to predict call-center load depending on users count. Unfortunately, there is no available public models for such prediction.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This model should correspond for following requirements, filled by cloud service:

1. The model must represent the internal parameters available for editing and viewing
2. The model must be extensible
3. The input parameters of the model must be the following:
    a. the number of CIS users;
    b. typical user characteristics;
4. The output parameter of the model must be a forecasted number of Internet communications

2. METHODS

2.1. Simulation Model Developing

Developing a simulation model in the Repast Simphony software is the development of software classes in Java; each class is a description of the class of simulated agents.

According to the logical-mathematical model, the following classes have been developed:

1) Corporate information system is represented by the class:

```
class CorporateSystem extends ReLogoTurtle {
    int generatedUsers = 0
    def step()
}
```

The main method of the class is `step()`, which creates new users of the contact center and counts the total number of users created in the `generatedUsers` class variable. This method is called at the beginning of each new time slice. In such a case, users are created unless the number of created users reaches the model input value.

2) The user of a corporate information system, which starts to use an Internet communications cloud service, is represented by the class (this class is the basic in terms of the logic of model operation):

```
class User extends ReLogoTurtle{
    Call currentCall
    boolean sleeping
    class ScheduledCall
    ScheduledCall nextCall
    def step()
    boolean checkSleeping(int hour)
    def scheduleCall()
    int getCallsCount(int hour)
    def getCallLength()
}
```

The main method of the class is also `step()`, which describes the agent behavior in each moment of model time.

- *currentCall* variable contains the current communication of the user, in order not to create a new communication until the previous one ends;
- *sleeping* variable determines whether the user will be active in the nearest hour or not;
- *scheduleCall* inner class defines the next scheduled communication in accordance with the required distribution of the communication; this communication is stored in the nextCall variable;
- *boolean checkSleeping(int hour)* method allows to determine whether a user is to be active during this hour, in accordance with the calculated distribution;
- *def scheduleCall()* method is designed to plan the next communication;
- *int getCallsCount(int hour)* method is designed to determine the number of communications that the user is to perform within a new hour;
- *def getCallLength()* method is intended for determining the upcoming communication duration according to the calculated communication distribution.

3) Internet communications cloud service is a class which combines both the general management of the model operation and the calculation of the statistical parameters of the contact center operation:

```
class UserObserver extends ReLogoObserver{
    static int hour
    static int callsCount
    static int usersCount
    static int onlineUsersCount
    static int
    static double cpuUsage
    def setup()
    def perHourInit
    def go()
}
```

Journal of Theoretical and Applied Information Technology
15th November 2016. Vol.93. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The class description:

- *static int hour* is a current hour;
- *static int callsCount* is an instant value of the number of communications;
- *static int usersCount* is the total number of users;
- *static int onlineUsersCount* is the number of users online during the current hour;
- *static int perHourCallsCount* is the number of communications during the current hour;
- *static double cpuUsage* is CPU load during the current hour;
- *def setup()* is the original model initialization;
- *def perHourInit()* is parameters initialization, which are aggregated per hours;
- *def go()* is a model step, which is executed in each time unit, and the agent controls the steps and the other agents.

With that, the user sets the following parameters separately for the whole model:

- *totalUsers* are the users of corporate information system which will use the Internet communications cloud service;
- *usersPerGeneration* are the users of corporate information system which start using the Internet communications cloud service on each occurrence of the moment of adding (the frequency of adding is set in the parameter frequency);
- *frequency* is CIS-user adding period (hour);
- *maxUsersPerTurtle* is the number of Internet communications cloud service users per display unit;
- *defaultUsersCount* is the initial number of Internet communications cloud service users.

2.2. Planning And Execution Of Simulation Experiments

When conducting simulation experiments the following series of experiments have been planned and performed:

1) model verification;
2) integration process modeling.

2.2.1. Model verification

To verify the model, a series of experiments has been conducted, aimed to confirm the compliance of the output model values with the real statistical data with the same input data. Therefore, in the experiments, the parameters, corresponding to the current performance of the contact center functioning, have been used to make the actual model-based measurement: 480 users. The number of simultaneous communications, as well as the number of online users, complies with the defined in [1].

For verification, a certain number of experiments are necessary, which allow speaking about the validity of the results. The required number of experiments is determined to be 110, which corresponds to the number of working days taken into account in the calculations.

In the course of the experiment, the following model characteristics have been compared:

1) The number of communications per hour shown by hour;
2) the number of online users shown by hour;
3) CPU load, depending on the number of communications.

The values calculated at the stage of logical-mathematical model construction have been compared with the values obtained from the results of actual model-based measurement; and the results of the comparison are listed below. It was expected that the values would coincide.

On the basis of the average value of the number of online users, the following charts were obtained: the orange line represents the actual value of the parameter; the blue line represents the simulated one (Figure 1).

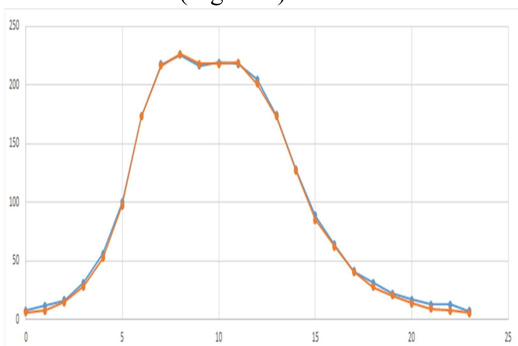

Figure 1 – The Average Charts Of The Number Of Online Users Shown Per Hours

In the charts in the Figure 1, the simulated value matches substantially completely to the actual value (with the difference less than 1%), obtained from the real statistical data; that proves the correctness of this parameter modeling.

On the basis of the average value of the number of simultaneous communications, the following charts have been obtained: the orange

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

line represents the actual value of the parameter; the blue line represents the simulated one (Figure 2).

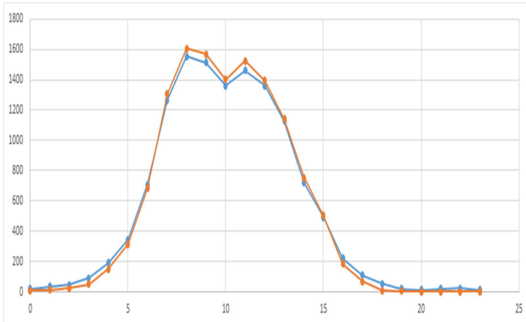

Figure 2 – The Average Charts Of The Number Of Simultaneous Communications Shown Per Hours

In the charts in the Figure 1, the simulated value matches substantially completely to the actual one, obtained from the real statistical data; that proves the correctness of this parameter modeling. The deviation in the range of 5% is accepted by the industrial partner.

On the basis of the average CPU usage value, the following charts have been obtained: the orange line represents the actual value of the parameter; the blue line represents the simulated one (Figure 3).

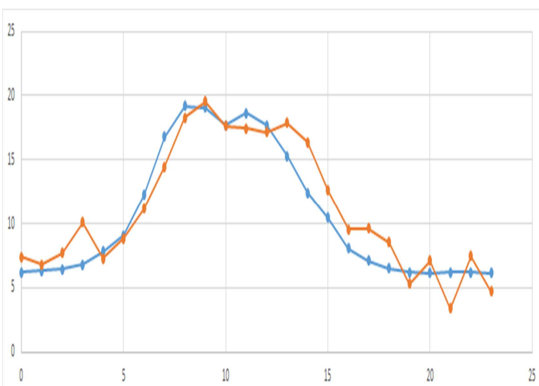

Figure 3 - CPU Load Average Charts

In the charts in the Figure 3, the simulated value matches substantially completely to the actual one, obtained from the real statistical data; that proves the correctness of this parameter modeling. The main point in this case is an exact matching of the peak value.

The biggest differences between the charts for the "CPU load" parameter are explained by the fact that the CPU is utilized not only by communications, but also by other applications that operate on the server; meanwhile the average value is used in modeling regardless of the hours.

2.2.2. Checking of the developed model compliance with the technical assignment requirements

The checking of the developed model compliance with the requirements, presented in introduction, is represented in table 1.

Table 1 – Checking Of The Developed Model Compliance With The Technical Assignment

Technical requirements	Compliance with the requirement
The model must represent the internal parameters available for editing and viewing	Graphical interface of working with the model allows you editing the internal parameters of the model
The model must be extensible	Extension is carried out by adding new software modules, as well as changing the original code of existing modules
The input parameters of the model must be the following: - the number of CIS users; - typical user characteristics;	The number of CIS users is defined in the model parameter *totalUsers* and *usersPerGeneration* The typical user characteristics are defined in such functions as *checkSleeping, scheduleCall, getCallsCount, getCallLength*
The output parameter of the model must be a forecasted number of Internet communications	The forecasted Internet communications are presented in the *perHourCallsCount* model variable, which contains the number of communications executed during the current hour

Thus, the developed simulation model fully meets the technical requirements.

2.2.3. Conducting simulation experiments for the integration process

The integration process has been simulated to determine the contact center CPU utilization related to the integration to a new corporate

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

information system. It is necessary to assess how cloud contact center load will change due to the integration to a new corporate information system, as well as whether an increase in the cloud contact center computing power utilization.

The experimental conditions. According to the requirements of the industrial partner, interest is the simulation of a situation of integration to the corporate information system, in which new 200 users will appear as a result of the integration. Presumably, users will appear with a frequency of 20 users per 1 day.

The following experiment has been conducted.

1) During the first 10 days of simulation, new users were added with a gradual increase in load.
2) During the next 110 days of simulation, the statistics was accumulated to conduct further analysis.
3) Next, an analysis of the results has been carried out.

3. RESULTS AND DISCUSSION

In the process of simulation experiments, average results with the following characteristics have been obtained:

1) The number of communications per hour shown per hours;
2) The number of online users shown per hours;
3) The CPU load according to the communications.

The results of comparison of values obtained from the integration process simulation, with the ones obtained on the basis of statistical analysis of the Internet communications cloud service, which has been made while building logical-mathematical model in [1], are listed below.

On the basis of the average value of the number of online users, the following charts have been obtained: the orange line represents the statistical value, the blue line represents the simulated one (Figure 4).

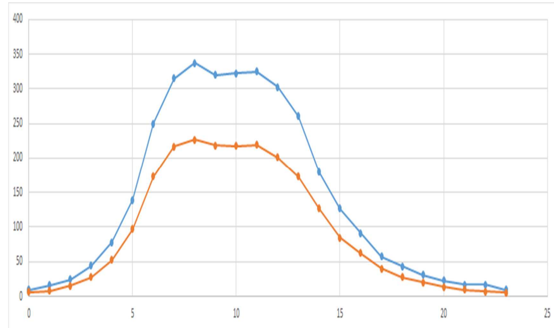

Figure 4 – Average Charts Of The Online Users Number

The graphs in Figure 4 show an increase in the number of online users due to an increase in the number of the users of the Internet communications cloud service. The number of users increased proportionally to the number of new users.

On the basis of the average value of the number of simultaneous communications, the following charts have been obtained: the orange line represents a statistical value; the blue line represents the simulated one (Figure 5).

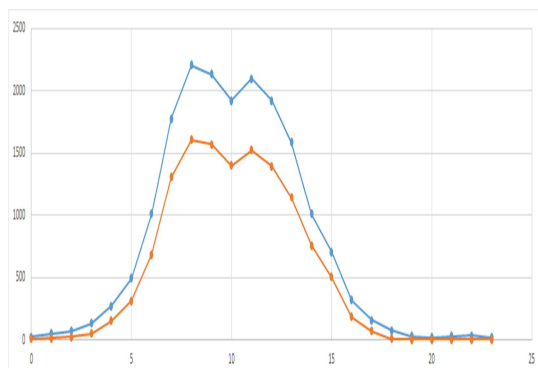

Figure 5 – Average Charts Of The Simultaneous Communications Number

The graphs in Figure 5 show an increase in the number of communications due to an increase in the number of cloud service users of Internet communications. The number of communications correlates with the number of new users.

On the basis of the average CPU load value, the following charts have been obtained: the orange line represents a statistical value, line represents the simulated one (Figure 6).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

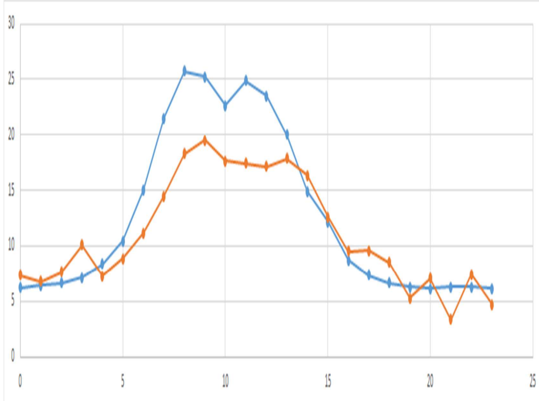

Figure 6 – Average Charts Of CPU Load

In the case of the CPU load, the difference of the simulated value from the statistical one also correlates with the number of users taking into account the presence of a constant component of 6%.

Thus, the resulting model provides an estimate of the characteristics of functioning of the cloud contact center, and provides an additional set of statistics, which allows estimating such parameters of functioning of the cloud contact center as the maximum CPU load, the maximum number of simultaneous communications, and the maximum number of concurrent users online.

By analyzing the obtained results of the experiments, the following data have been gathered:

1. The maximum CPU load is 38%;
2. The maximum number of communications per hour reaches 2500;
3. 42 simultaneous communications with an average duration of 62 seconds of a single communication.

These data lead to the conclusion that the integration of the system described in point 2.2.3 within system described in [1] does not require an increase in computing power. As this model of load prediction for cloud contact center has no analogues these results are significant for cloud service.

4. CONCLUSION

In the current stage of applied research, a simulation model has been developed representing Internet communications cloud services functional integration and on a data level one to corporate information systems. To build a model, agent-based simulation method has been chosen. For the models implementation Repast Simphony tool has been selected among open and free simulation tools.

In accordance with the requirements to the model, developed in [1], the logical-mathematical model has been built and developed in Java using the Repast Simphony tool, in particular:

1) The following logical elements involved in the modeling have been defined: corporate information system, the user, the Internet communications cloud service;
2) The numerical parameters of the functioning of these logical elements of the model have been determined: the distribution percentage of users making calls, per hours; the probability of communication with a certain duration; the number of communications made by user, per hours; the number of online users, per hours; CPU load, depending on the number of communications.

A comparison of the specified requirements to the developed model is presented in paragraph 2.2.2 and confirms compliance of the developed model with the requirements.

To verify the model, simulation experiments have been carried out, followed by comparison with the actual statistical data. The results show the model compliance with the existing system since the simulation parameters values are consistent with statistical values of these parameters.

CPU load experiment has been conducted to evaluate the effect of the planned integration to the corporate information system, in accordance with the requirements of the industrial partner. In the experiment, a corporate information system bringing 200 new users was considered. It was discovered that the maximum CPU utilization is 38%; the maximum number of communications

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

is up to 2,500 per hour, which corresponds to the 42 simultaneous communications with average communication duration of 62 seconds. Based on these results, it was concluded that the integration to the corporate information system similar to the one considered in this experiment does not require an increase in computing power.

Thus, the developed simulation model can be used to assess the impact of new integrations to different corporate information systems to internet service with prediction of resources load. It's commonly needed by cloud services to predict resource utilization to avoid resource overuse.

Further direction for research is to make modeling more complex to allow predict not only resource load but overall resource count needed to serve desired amount of users, for example, this model should ask the following questions: how many servers with how many CPUs and how many GB of RAM we need to serve 10 thousands of users.

5. ACKNOWLEDGEMENTS

The article is published under financial support of the Ministry of Education and Science of the Russian Federation. The unique identifier of the applied scientific research RFMEFI57614X0081.

REFERENCES

[1] Teykhrib A.P. Developing of the Logical-Mathematical Model of Cloud Call-Center to Predict the Utilization in Case of Increase in the Number of Users. *International Journal of Applied Engineering Research.* Vol. 11, No. 13, 2016, pp. 8005-8009

[2] The Repast Suite. http://repast.sourceforge.net/> Date accessed: 02/06/2016

[3] He, Wu, and Li Da Xu. "Integration of distributed enterprise applications: a survey." IEEE Transactions on Industrial Informatics 10, no. 1 (2014): 35-42.

[4] Maidstone R. Discrete Event Simulation, System Dynamics and Agent Based Simulation: Discussion and Comparison. System. 2012 Mar 7:1-6.

[5] Yarygin O.H., and E.S.Roganov. "The study of system dynamics as a tool for the formation of competence of managers and researchers." *Science Vector TSU. Series: Economics and Management 2* (2012): p.206-209.

[6] Timohin V.N., Podskrebko A.S. Discrete event simulation of conveyor lines // BI. 2013. №9. URL: http://cyberleninka.ru/article/n/diskretno-sobytiynoe-modelirovanie-konveyernyh-liniy (reference date: 01.06.2016). Scientific Library KiberLeninka: http://cyberleninka.ru/article/n/diskretno-sobytiynoe-modelirovanie-konveyernyh-liniy#ixzz4AJKN5TuT

[7] Borschev, A.V. "Practical agent-based modeling and its place in the arsenal of analyst." *Exponenta Pro* 3-4 (2004): pp.38-47.

[8] Siebers PO, Macal CM, Garnett J, Buxton D and Pidd M (2010) 'Discrete-Event Simulation is Dead, Long Live Agent-Based Simulation!'. Journal of Simulation, 4(3) pp. 204-210.