Task-level simulation of service based information systems

Tomasz Walkowiak
Wroclaw University of Technology, Poland
tomasz.walkowiak@pwr.wroc.pl

Maciej Woropay
University of Technology and Life Sciences in Bydgoszcz, Poland

ABSTRACT

The paper presents a model and simulator of an information system. The system is being analysed from the functional and user point of view. The model was developed to allow the simulation of an information system and calculation of functional metrics like response time to a user request. The user request execution is modelled as a sequence of task being executed on services delivered by the system. Results of an exemplar system analysis are given.

Key Words: computer systems, modelling, dependability, simulation

1. Introduction

Complex information systems (CIS) are nowadays the core of a large number of companies. And therefore, there is a large need to analyze various system configuration and chose the optimal solution during design and even operation of the information system.

In this paper we propose a common approach [2] based on modelling and simulation. The aim of simulation is to calculate some performance metrics which should allow to compare different configuration taking into consideration technical (like performance) and economical (like price) aspects. The metric is calculated using Monte Carlo techniques [3].

There is a large number of event driven computer network simulations, like OPNET, NS-2, QualNet, OMNeT++ or SSFNet/PRIME SSF[4,7]. However, they are mainly focused on a low level simulation (TCP/IP packets). We propose a higher level of modelling focusing on functional aspects of the system, i.e. performance aspects of business service realized by an information system (like buying a book in the internet bookstore). Therefore, we model and simulate a process of execution of a user request, understand as a sequence of task realised on technical services provided by the system.

The structure of the paper is as follows. In Section 2, a model of information system is given. In Section 3, information of simulator implementation is given, next an exemplar information system is analysed and simulation results are presented Finally, there are conclusions and plans for further work.

2. Computer information system modelling

As it was mentioned in the introduction we decided to analyze the CIS from the business service point of view. Generally speaking users of the system are generating tasks which are being realized by the CIS. The task to be realized requires some services presented in the system. A realization of the system service needs a defined set of technical resources.

Therefore, we can model CIS as a triple:

$$CIS = \langle C, BS, TI \rangle \qquad (1)$$

where: C – is the client model, BS – is business services provided by the system, and TI – technical infrastructure of the system.

During modelling of the technical infrastructure we have to take into consideration functional and reliability aspects of CIS. Therefore, the technical infrastructure of the computer system could be modelled as a pair:

$$TI = \langle H, N \rangle \qquad (2)$$

where: H - set of hosts (computers); N – computer network.

We have assumed that the aspects of TCP/IP traffic are negligible therefore we will model the network communication as a random delay. Therefore, the N is a function which gives a value of time of sending a packet form one host (v_i) to another (v_i).

The time delay is modelled by a Gaussian distribution with a standard deviation equal to 10% of mean value.

The main technical infrastructure of the CIS are hosts. Each host is described by its reliability parameter (mean time to failure and mean time of repair) and functional parameters:

- server name (unique in the system),
- host performance parameter – the real value which is a base for calculating the task processing time (described later),
- set of technical services (i.e. apache web server, tomcat, MySQL database), each technical service is described by a name and a limit of tasks concurrently being executed.

The *BS* is a set of services based on business logic, that can be loaded and repeatedly used for concrete business handling process (i.e. ticketing service, banking, VoIP, etc). Business service can be seen as a set of service components and tasks, that are used to provide service in accordance with business logic for this process [6].

Therefore, *BS* is modelled a set of business service components (*BSC*), (i.e. authentication, data base service, web service, etc.), where each business service component is described a name, reference to a technical service and host describing allocation of business service component on the technical infrastructure and a set of tasks. Tasks are the lowest level observable entities in the modelled system. It can be seen as a request and response form one service component to another. Each task is described by its name, task processing time parameter and optionally by a sequence of task calls. Each task call is defined by a name of business service component and task name within this business service component and time-out parameter.

The client model consist of set of users where each user is defined by its allocation (host name), replicate parameter (number of concurrently ruing users of given type), set of activities (name and a sequence of task calls) and inter-activity delay time (modelled by a Gaussian distribution).

Summarising a user initiate the communication requesting some tasks on a host, it could require a request to another host or hosts, after the task execution hosts responds to requesting server, and finally the user receives the respond. Requests and responds of each task gives a sequence of a user task execution as presented on exemplar Fig. 1.

The user request execution time in the system is calculated as a sum of times required for TCP/IP communication and times of tasks processing on a given host.

The task processing time is calculated taking into consideration the host performance parameter, the task processing time parameter and other task processed on the host in the same time. Since the number of tasks is changing in simulation time the processing time is updated each time a task finish the execution or a new task is starting to be processed.

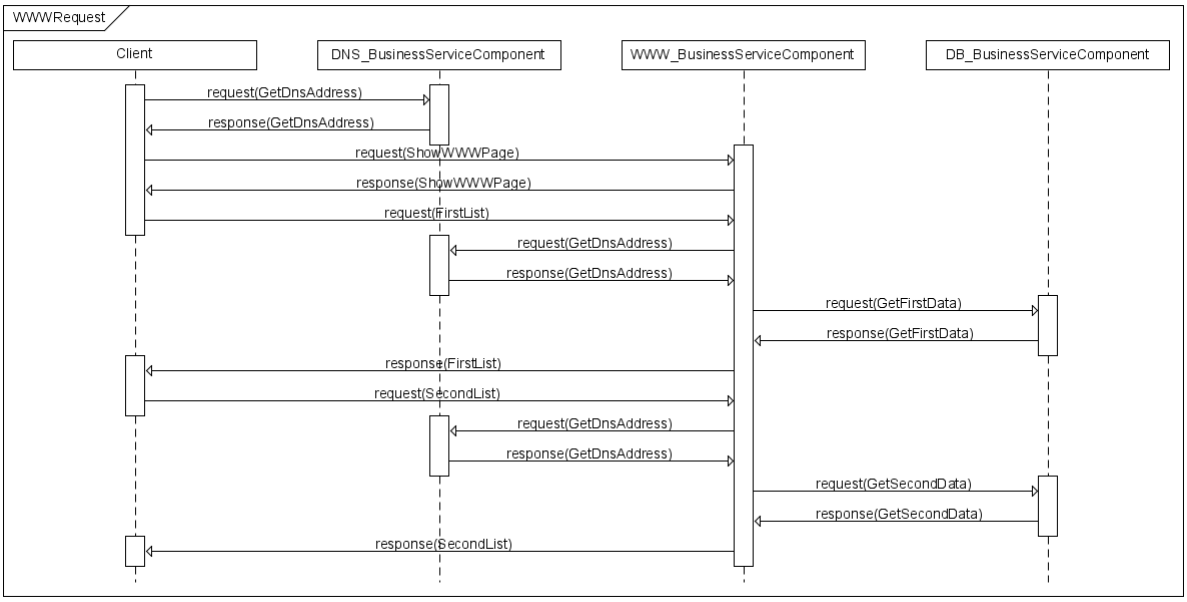

Fig. 1. Task and business services interaction

3. Task-level simulator

Once a model has been developed, it is executed on a computer. It is done by a computer program which steps through time. One way of doing it is so called event-simulation. Which is based on a idea of event, which could is described by time of event occurring, type of event (in case of CIS it could be host failure) and element or set of elements of the system on which event has its influence. The simulation is done by analyzing a queue of event (sorted by time of event occurring) while updating the states of system elements according to rules related to a proper type of event.

The event-simulation program could be written in general purpose programming language (like C++), in fast prototyping environment (like Matlab) or special purpose discrete-event simulation kernels.

One of such kernels, is the Scalable Simulation Framework (SSF) [5] which is a used for SSFNet [5] computer network simulator. SSF is an object-oriented API - a collection of class interfaces with prototype implementations. It is available in C++ and Java. SSFAPI defines just five base classes: Entity, inChannel, outChannel, Process, and Event. The communication between entities and delivery of events is done by channels (channel mappings connects entities).

For the purpose of simulating CIS we have used Parallel Real-time Immersive Modeling Environment (PRIME) [4] implementation of SSF due to much better documentation then available for original SSF. We have developed a generic class (named BSObject) derived from SSF Entity which is a base of classes modeling CIS objects: host and client which models the behavior of CIS presented in section 2. Each object of BSObject class is connected with all other objects of that type by SFF channels what allows communication between them. In the first approach we have realized each client as a separated object. However, in case of increasing of the number of replicated clients the number of channels increases in power of two resulting in a large memory consumption and a long time for initialization simulation objects. Therefore, we have changed the implementation, and each replicated client is represented by one object.

4. Computer information system analysis – case study

For testing purposes of presented CIS system model (section 2) and developed extension of SSF (section 3) we have try to analyze a case study information system. It consists of one type of client, four hosts, three technical services and three business service components. An interaction between a client and tasks of each business service component is presented on UML diagram on Fig 1. The CIS structure as well as other functional parameters were described in a DML file (see example in Fig 2). The Domain Modeling Language (DML) [4] is a SSF specific text-based language which includes a hierarchical list of attributes used to describe the topology of the model and model attributes values.

In the presented system we have observed the response time to a client request in a function of number of clients. The simulation time was set to 1000 seconds and each simulation was repeated 10.000 times (Monte-Carlo approach). The achieved results are presented in Fig 3.

```
Net [
Host [
  Name DNS-Server
     Service [
        Name DNSService
        Limit 110
       LocalTask [
         Name GetDnsAddress
         Time 0.01]]]
     …
Client [
  Name Client
  Replicate 1000
  Sleep 10.0
  Activity [
     Name WWWRequest
     TaskCall [
       Host DNS-Server
       Service DNSService
       Task GetDnsAddress
          ]
       …
```

Fig. 2 Exemplar CIS description in DML file

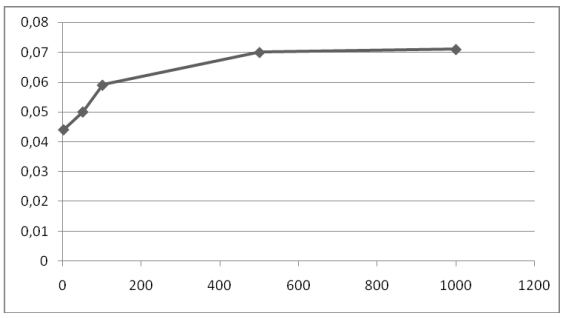

Fig. 3. Response time to clients requests in a function of number of clients.

5. Conclusions

We have presented a simulation approach to functional analysis of complex information systems. Developed simulation software allows to analyze the effectiveness (understood in given exemplar as a the response time to a client request) of a given configuration of computer system. Changes in a host performance or in a number of clients can be easily verified. Also, some economic analysis could be done following the idea presented in [7]. The implementation of CIS simulator done based on SSF allows to apply in a simple and fast way changes in the CIS model. Also the time performance of SSF kernel results in a very effective simulator of CIS.

References:

[1] Avižienis, A., Laprie, J., Randell, B.: Fundamental Concepts of Dependability. In: 3rd Information Survivability Workshop (ISW-2000), Boston, Massachusetts, USA, 2000.

[2] Birta L., Arbez G.: Modelling and Simulation: Exploring Dynamic System Behaviour. Springer London (2007)

[3] Fishman, G.: Monte Carlo: Concepts, Algorithms, and Applications. Springer-Verlag, New York (1996)

[4] Liu J.: Parallel Real-time Immersive Modeling Environment (PRIME), Scalable Simulation Framework (SSF), User's maual, Colorado School of Mines Department of Mathematical and Computer Sciences, 2006,

[Online]. Available: http://prime.mines.edu/.

[5] Nicol, D., Liu, J., Liljenstam, M., Guanhua Y.: Simulation of large scale networks using SSF. Proceedings of the 2003 Winter Simulation Conference, Volume 1, 7-10 December 2003,(2003) 650–657

[6] Michalska, K. and Walkowiak, T.: Hierarchical Approach to Dependability Analysis of Information Systems by Modeling and Simulation. In: Proceedings of the 2008 Second international Conference on Emerging Security information, Systems and Technologies. SECURWARE. IEEE Computer Society, Washington, DC, 356-361, 2008.

[7] Walkowiak T., Mazurkiewicz J.: Reliability and Functional Analysis of Discrete Transport System with Dispatcher, Advances in Safety and Reliability, European Safety and Reliability Conference – ESREL 2005, Poland, June 2005, K. Kolowrocki (eds). Taylor & Francis Group London, 2005, pp. 2017-2023