The simulation model of the computer cluster

V V Sokolova[1], **O M Zamyatina**[1]

[1] Tomsk Polytechnic University, 30, Lenina Ave., Tomsk, 634050, Russia

E-mail: Veronica@tpu.ru

Abstract. Simulation is often used in cases when it is impossible to carry out experiments with real complex objects. The article represents the description of the computer cluster simulation model. Parameters, which affect the cluster performance, were selected, a simulation model was designed, and experiments were conducted. The obtained model allowed finding the optimal variant of the cluster performance, which consists of five computers.

1. Introduction

Nowadays, simulation is a universal method for investigating and evaluating the performance of complex systems, such as transport, financial and manufacturing systems. One of its types is discrete-continuous modelling [1], which is carried out step by step with changes in the values of the system parameters. In this case, the simulation model allows describing the structure of the system and reflects the changes in the states of the system. The test system parameters are presented in a tabular form in accordance with the sequence of events. The state of each system elements is described by a set of parameters that are stored in computer memory in the tabular form, and the interaction of the system elements are described algorithmically.

In general, the creation of a simulation model allows identifying the most important parameters to ensure the effective functioning of complex systems, see the processes at different periods of time and discover the hidden reserves of the simulated systems. For these reasons, a simulation model for a computer cluster was developed.

By a "*cluster*" we shall mean a group of computers, connected through high speed communication channels, and which is seen by an end-user as a single hardware resource [2]. Clustering allows one to create high-performance and reliable systems, especially in the computing.

Either storing or processing large amounts of data requires really powerful computer resources. One of the possible solutions here is using clusters for distributed-parallel computing. Using supercomputer clusters results in decreasing the number of expensive field tests, required to predict the system behavior in an emergency, which allows one to considerably reduce production costs of test samples. Again, every so often, there is a complex problem how to model the running of a computer cluster when it performs distributed-parallel computing.

The solution of this problem enables to do the following:

- facilitate and improve the process of incoming data distribution between cluster computers;
- single out cluster parameters, vital for successful functioning, and analyze their interconnections;
- examine processes in various time scales;
- develop processes, which prevent from deadlocks and indeterminancy, caused by system randomness and variability;

– detect hidden reserves and resources of a modelled system and make the most of them.

Therefore, we shall review functioning of a computer cluster as an example of computing network modelling in order to make the task assignment between its computers as optimal as possible.

2. Description of a supercomputer cluster model

Let us assume that a supercomputer cluster receives tasks as packets with the finite size varying from 10 Mb to 100 Mb, most often - 60 Mb. Every task requires a certain processing algorithm (5 algorithms in total) in a cluster. Tasks are assigned to algorithm complexity at random.

A cluster has a central computer that preprocesses and distributes tasks, as well as collects the results of processing from 5 computers. Machines can process up to 20 Mb tasks.

According to the size of a task, it is distributed between a number of computers. The cluster has got a star topology. Communication channels, connecting the central computer with others, are duplex; and have the same capacity of 1 Mbyte per second. Tasks arrive at the central computer and get distributed. Besides, in case the cluster has free resources (computers), they can be assigned to another (succeeding) task.

We choose the Arena software (developed by Rockwell Software, the USA), allowing to simulate various processes and systems in terms of the object domain for further application as a modelling environment [3]. This software product provides the ability to collect statistics after modeling, which speeds up the further analysis of the models.

A simulation model includes the following main elements: create, dispose, process and queue [4]. Sources are elements of the model which receive information or objects (entities). A task, in terms of the mathematical apparatus of queuering systems and the Arena software, will be called an entity. The speed at which data or objects from source is usually defined by a statistical function. Dispose – a device for receiving information or objects. Processes – an analogue of the works in the functional model, they are responsible for processing entities.

In the simulation model a process performance can be set. The concept of a queue close to the concept of the data warehouse is the place where the processing facilities are expected. Object processing time (productivity) in different processes may be different. As a result, in front of some processes we can build applications (entities), waiting for their turn. Requests, which are the items, documents, tasks or messages, streaming charts, standing in queues are handled, captured and released resources, divided, combined, etc.

For our purpose, we divide the logic of the model into 2 parts: *tasks arrival* and *tasks processing*.

1. Tasks arrival

The “Create 1” module generates the entity arrival into a system via a random variables generator. The maximum number of tasks – 100. Tasks arrive at the “Decide 3” module, from which they are sent at random and with equal probability to any of these modules: Assign 1, Assign 2, Assign 3, Assign 5 or Assign 6. Having passed through one of these modules, each entity (task) is assigned to processing time, which equals the value of Attribute 2. The processing time, Attribute 2, will affect the time, required to process an incoming task and it models the task processing algorithm of cluster computers, i.e. determines the workload of cluster nodes.

Then tasks pass through the “Assign 4” module, where the value of “Attribute 1” is set. “Attribute 1” – task size – varies from 10 Mb to 100 Mb. As a result, the computer cluster (the “Decide 2” module) receives tasks with a certain finite size (Attribute 1) which require certain processing time (Attribute 2).

Inside “Decide 2”, according to the task length, tasks are distributed into five queues. The “Hold 1” module receives tasks with the length varying from 10 to 20 Mb, the “Hold 2” module – with the length of 21–40 Mb, “Hold 3” – with the length of 41–60 Mb, “Hold 4” – with the length of 61–80 Mb, and “Hold 5” – with the length of 81–100 Mb. Every “Hold” module releases entities according to an assigned condition (the number of free resources and pre-set priority of a module).

2. Tasks processing

Computers, processing tasks, are presented as a single resource (Resource 1) with the given

capacity – 5. This resource is simulated in five different Process modules with varying capacity: 1, 2, 3, 4, 5. According to their size, tasks are processed by various processes and occupy the different number of resource units. Entities, leaving "Hold 1.Queue", arrive at the "Process 1" module and occupy one unit of "Resource 1". Accordingly, entities from "Hold 2.Queue" occupy two resource units, from "Hold 3.Queue" – three units, from "Hold 4.Queue" – four, and from "Hold 5.Queue" – all five units. In each case, task processing time is equal to the value of "Attribute 2" of a given entity (entity processing time). Having been processed by computers, tasks arrive at the "Dispose 1" module to be deleted from the system. At first, the model was developed in which the highest priority has a greater size on the task, requiring for its treatment of the maximum number of resources.

3. Analysis of tasks processing priorities

We simulate processing of 100 tasks to see how such parameters as task processing time, computer workload and average number of queueing tasks, change when the sequence of processing tasks varies. In case a cluster consists of five nodes, the number of variants for the sequence of processing equals 5, i.e. 120.

Now let us consider examples of possible processing sequences:

- first, tasks with the smallest "Attribute 1" will be processed, i.e. tasks with the size from 0 to 20 Mb, then in an ascending order with Attribute 1 – from 20 to 40 Mb and etc. This sequence is denoted as 1_2_3_4_5;
- the reverse order is possible, when tasks will be processed as the value of Attribute 1 goes down, beginning with tasks of the largest, 80–100 Mb, size, then 60–80 Mb and etc. This sequence is denoted as 5_4_3_2_1;
- by changing this way of processing sequences, we shall simulate all 120 processing possibilities.

Experiment results of these series have been very different. The smallest (and the best for this research) task processing time, equal to 6.28 s, has been achieved in the following sequences: 1_4_3_2_5, 1_4_3_5_2, 1_5_4_3_2, 4_1_3_2_5, 4_3_2_1_5, 4_3_2_5_1, 4_5_3_2_1. The worst time, equal to 15:03:27, has been obtained in sequences: 2_4_5_3_1, 2_4_5_1_3, 2_3_4_1_5, 2_3_5_1_4.

The best result for this research is the maximum workload of computers, i.e. incoming tasks are distributed so that to guarantee idle-free running of all cluster computers. The following sequences have shown the best experiment results (the maximum workload): 1_4_3_2_5, 1_4_3_5_2, 1_5_4_3_2, 4_1_3_2_5, 4_3_2_1_5, 4_3_2_5_1, 4_5_3_2_1, The sequences with the worst results are: 2_4_5_3_1, 2_4_5_1_3, 2_3_4_1_5, 2_3_5_1_4.

The best modelling result is the minimum number of tasks, queuing to be processed. Thus, the best results are:

- for computer 1: 1_3_4_5_2, 1_4_3_2_5, 1_4_3_5_2, 1_5_4_3_2, 2_3_4_1_5, 2_3_5_1_4, 2_4_3_1_5,2_4_5_1_3, 4_5_3_2_1;
- for computer 2: 2_1_5_4_3;
- for computer 3: 1_2_4_3_5, 1_2_4_5_3;
- for computer 4: 4_5_2_3_1, 4_5_3_2_1;
- for computer 5: 2_4_5_3_1.

We shall make a summary table of experiment data, which enables to define the optimum distribution of processing tasks (table 1).

Table 1 contains a data sampling from all 120 experiments, which have led to the best combinations according to at least 1 parameter: task processing time, the average number of tasks in queues, computer workload. The last column of the table shows the number of combinations of the best parameters.

Table 1. Comparative data of modelling results

Sequence	Time, s	Average number of entities					Computer workload, s	Number of matches
		Queue 1	Queue 2	Queue 3	Queue 4	Queue 5		
1_2_4_3_5	9:50:43	0.0418	0.1566	**0.2138**	4.2137	17.3516	4.2454	1
1_2_4_5_3	9:50:43	0.0418	0.1566	**0.2138**	4.2137	17.3516	4.2454	1
1_4_3_2_5	**6:22:29**	0.0359	5.0602	5.6218	3.2509	16.592	**4.7193**	2
1_4_3_5_2	**6:22:29**	0.0359	5.0602	5.6218	3.2509	16.592	**4.7193**	2
1_5_4_3_2	**6:22:29**	0.0207	5.5708	7.4727	6.1213	11.6352	**4.7193**	2
2_1_5_4_3	12:04:45	0.0138	**0.0838**	2.3764	10.4509	1.1969	3.8569	1
2_3_4_1_5	15:03:27	**0.0000**	4.0222	13.8152	3.0450	7.7039	3.7348	1
2_3_5_1_4	15:03:27	**0.0000**	3.9644	25.0443	2.9005	3.9653	3.7348	1
2_4_3_1_5	12:04:45	**0.0000**	4.9231	20.3758	3.1399	7.5969	4.0431	1
2_4_5_1_3	15:03:27	**0.0000**	5.2127	25.3230	2.9005	3.0675	3.7348	1
2_4_5_3_1	15:03:27	0.0429	5.3694	25.3566	5.7620	**0.2370**	3.7348	1
2_5_3_1_4	12:39:34	**0.0000**	4.6468	20.5508	3.0902	7.6191	3.9791	1
2_5_4_1_3	12:39:34	**0.0000**	5.9767	24.3638	3.0902	3.2681	3.9791	1
4_1_3_2_5	**6:22:29**	0.0487	6.5768	7.073	1.1409	16.592	**4.7193**	2
4_3_2_1_5	**6:22:29**	1.6565	2.6841	3.6266	5.5577	16.592	**4.7193**	2
4_3_2_5_1	**6:22:29**	1.6565	2.6841	3.6266	5.5577	16.592	**4.7193**	2
4_5_3_2_1	**6:22:29**	0.0584	12.4622	15.4654	**0.8454**	7.1423	**4.7193**	**3**

Having analyzed this table, we can make a conclusion that the highest efficiency of the supercomputer cluster is achieved when 60–80 Mb tasks are processed first, followed by ones of 80–100 Mb, 40–60 Mb, 20–40 Mb, and 0–10 Mb tasks are the last to be processed (4_5_3_2_1).

4. Conclusion

This article reviews the example of modelling the supercomputer cluster, wherefore the functioning basics of the cluster are examined, and its main parameters are defined.

The Arena 9.0 software is used to develop a simulation of the computer cluster, which enables to analyze distribution of incoming tasks between cluster computers.

In the creation of a simulation model of a cluster consisting of 5 computers, the follows main parameters were chosen: task processing time, loading computers and the average number of task in the queue. Building a simulation model allows testing different options for the cluster, allows finding the optimal distribution of resources between tasks, and simulating computers with different intensities boot.

After the development of the computer cluster model, various priorities of tasks queues have been analyzed. According to the obtained results, the highest efficiency of running the computer cluster for distributed-parallel computing can be achieved when 60–80 Mb tasks are processed first, then 80–100 Mb, 40–60 Mb, 20–40 Mb and last of all – 0–10 Mb (4_5_3_2_1).

References

[1] Law A M, Kelton W D 2000 *Simulation modeling and analysis* (New York: Mc. Graw-Hill) p 544
[2] High-performance computing // The official site of Trinity group [Internet resource]. Available at: http://www.trinitygroup.ru/ (last accessed date: 01.08.2016)
[3] Arena Simulation Software by Rockwell Automation [Internet resource]. Available at: http://www.arenasimulation.com/ (last accessed date: 01.09.2016)
[4] Zamyatina O M 2012 *Networks modelling and simulation* (Tomsk: SPB Graphics) p 151