**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 529−532**

Ки&М

УДК: 004.02

Visualization of work of a distributed application based on the mqcloud library

O. O. Iakushkin, V. M Grishkin

Saint Petersburg State University, University ave. 35, Peterhof, St. Petersburg, 198504, Russia

E-mail: [a] Oleg.jakushkin@gmail.com

Received October 10, 2014

Independent components communicating with each other due to complex control make the work of complex distributed computer systems poorly scalable within the framework of the existing communication middleware. Two major problems of such systems' scaling can be defined: overloading of unequal nodes due to proportional redistribution of workload and difficulties in the realization of continuous communication between several nodes of the system. This paper is focused on the developed solution enabling visualization of the work of such a dynamical system.

Keywords: distributed systems, cloud computing, services, monitoring, networking, peer-to-peer

Визуализация работы распределенного приложения на базе библиотеки mqcloud

О. О. Якушкин, В. М. Гришкин

Санкт-Петербургский государственный университет, Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35

Независимые компоненты, взаимодействующие между собой при помощи комплексного управления, делают работу сложных распределенных вычислительных систем плохо масштабируемой в рамках имеющегося промежуточного коммуникационного программного обеспечения. Можно выделить две основные проблемы масштабирования таких систем: перегрузка неравноценных узлов из-за равномерного перераспределения нагрузки и сложности в реализации продолжительного взаимодействия нескольких узлов системы. В данной работе мы рассмотрели созданное решение позволяющее обеспечивать визуальное отображение работы такой динамической системы.

Ключевые слова: сетевые распределенные вычисления, облачные вычисления, сервисные системы, сети передачи данных, мониторинг, одноранговые сети

The work has been done with partial support of SPbU grant 9.38.674.2013, SPbU grant 9.37.157.2014 and RFBR grant No. 13-07-747. The research has been done using the computing resources of the resource center "SPbU Computing Facility" (http://cc.spbu.ru).

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 529–532.

Introduction

Modeling the architecture of a distributed system plays a fundamental role in terms of bottleneck analysis, development from the perspective of horizontal scaling as well as protection of the presented system from external factors [Degtyarev et al., 2010; Bogdanov et al., 2006; Bogdanov et al., 2010]. This paper is focused on the creation of a means of interactive monitoring of the distributed network architecture in a service system. Means of interactive monitoring of various network topologies are studied here. The existing solutions are evaluated and the necessity of research required to develop our own ones is substantiated as well. The developed solution and its qualitative characteristics are described.

In addition to the existing internode communication library [Iakushkin, Grishkin, 2014.], a statistics presentation system for the nodes involved in communication has been developed. The existing means of data presentation in similar systems have been studied and a way of presenting the aggregated data of network statistics has been developed. Then an application with an option of displaying the system's current operation and tracing the changes dynamically has been created. The system has been launched using the computer powers of the resource center "SPbU Computing Facility".

The assigned development and testing tasks have been accomplished; the developed system successfully copes with the assigned tasks within the framework of the current stage of its development.

In future this system will be applicable during extension of the service base in the course of debugging and monitoring of the services of the systems of SPbU research projects No. 01201453381 and No. 01201453375. The system has an open interface and can be used by other projects for the sake of debugging network communication between distributed system components.

While working on a prototype of the communication system for monitoring and diagnostics of the state of cultural heritage sites as well as for usage in particular simulation experiments in modeling hydrophysical and hydrometeorological processes in the waters of the Baltic Sea within the framework of SPbU research projects No. 01201453381 and No. 01201453375, we have developed a non-broker transportation system enabling dynamical workload redistribution during nodes communication, with the support of various communication means.

Task description

Peer-to-peer (P2P) communication libraries for message exchange, such as ZeroMQ and NanoMsg, contain a pre-installed set of program and component interaction patterns. Other P2P communication libraries, such as libjingle, aimed at information streaming provide a narrow set of communicative primitives that do much succeed in expanding the abstraction pool already introduced by the Berkeley Software Distribution (BSD) sockets. The absence of well-defined rules for the approach to expanding the set of system interaction patterns that could be safe for the system and elaborately scaled entails uncertainty during product development and requires deep knowledge of all the architecture of the transportation solution in order to make any changes.

Thus, the users of our system had to add to the standard message transmission patterns, such as Push-Pool, Publish-Subscribe, Request-Reply and Pair, certain asynchronous Request-Reply capabilities that make it possible to meet clients' requests at the server's convenience and some long chains of Request-Reply pairs with an option of switching from the server to the client. Within the framework of the traditional non-broker architecture the users would have to create extensions to the existing libraries that currently do not offer formalized interfaces for expansion. In other words, to exercise total control over the behavior of the pattern introduced by them, having information about only one of the system's nodes, that could, in the process of horizontal scaling, result in the loss of stability of the entire system's operation and the application's responsiveness for the end users.

While working on the tasks assigned, we have developed a library offering interfaces that make it possible to systemically expand the set of offered means of communication. Let us note that the addition of communication patterns was connected with a range of problems: the uniform addressing scheme required for horizontal scaling; the unification of interfaces for the development of new communication components and further interaction with them; the opportunity to change the transportation level protocol without any need of rewriting the already created means of communication.

The systems that scholars have to work with within the framework of their research projects are very difficult for debugging. That is caused by many things, the problems of end service applications and the space of network communications uniting them being the major factors. Moreover, after the system has been launched, it is quite a difficult task to monitor its operability and bottlenecks.

That made us develop a means of interactive monitoring of the distributed network architecture, in addition to the already created library. This work has been focused on the creation of such a means and on the study of methods and practices aimed at solving such tasks.

The assigned task solution

Development stages

Our research consisted of three main parts.

- The study of the existing solutions aimed at the visualization of distributed systems binding elements with each other and displaying auxiliary information.
- The selection of the existing components that can become a base for the development of the suggested solution.
- The implementation, debugging and testing of the solution.

The existing solutions

During the first stage we have studied many existing solutions for visualization of specialized distributed systems. Yet, none of them is able to solve the task assigned. The most interesting solutions are the following:

- The system of download data presentation of the Windows Azure Service Bus message broker software;
- The modeling system of the RabbitMQ message broker software;
- The Boston subway traffic system visualization.

At this stage it has been decided to present the system nodes and their communications as a directed acyclic graph with floating tips above the elements containing information about the current state and work history.

Technological base

At the second stage the existing technological solutions with an open source code have been defined within the framework of the technologies suggested in the application. They have formed the basis of the developed solution. Let us consider the selected libraries.

- C# (.NET/Mono) statistics collection service and testing system services.
  - o ASP.NET 4.0 (mono) for web interface presentation;
  - o ASP.NET SignalR for the server's communication with the web interface;
  - o EntityFramework and SQLite for structured data storage;
  - o protobuf-net for reading the input data.
- JavaScript for application business logic processing.
  - o JQuery and SignalR for application operation logic control;
  - o D3 and dagre-d3 for displaying node communication, presentation of statistical data and work with the SVG format.

Protocol Buffers have been chosen for the formatting of the input data due to their high performance in terms of message reading and small-size message packages. The existence of such a solution made it possible to make the system an open one in terms of accepting statistics not only from the nodes dependent on the developed communication library but also from third-party solutions. The open interface makes it possible to load data created by systems written in more than ten popular programming languages such as Java, C++, Python, etc.

Debugging and testing

The selected technological base enabled us to create, during a short period of time, a new system integrated into the general solution. This system makes it possible to display on a near real time basis any changes in the system state and their communication characteristics.

The test system where the solution has been launched consists of three nodes bound by the Fast Ethernet network. Service applications communicating with each other have been launched on these nodes. The system features the following message transmission patterns:

- Publish-Subscribe;
- Request-Response;
- Pipe (in the form of a closed ring and directed graph).

The created subsystem-observer is located on one of the nodes.

The following groups of load modes concerning message transmission between the nodes per one second (simultaneously and subsequently) have been tested in it:

- 10, 100, 1000 messages per second;
- 1000, 5000, 10000 messages per second;
- 10000, 15000, 20000 messages per second;
- 10000, 50000, 100000, 200000 messages per second.

The system slightly influences the network workload of the data-sending nodes (in all the cases less than one per cent of the network load: 0.5 Mbit/s). The system does not consume more than five per cent of the CPU resource, that being true both for the workload of the 3 studied nodes and 5 listening clients and the workload of 100 connected nodes and 500 listening clients.

Conclusion

Our study has resulted in the development of an interactive system for creating a model of a dynamical distributed system of directly communicating nodes. This system can be helpful in the process of debugging, development and protection of the service systems used within the framework of SPbU research projects.

References

Bogdanov A. V., Degtyarev A. B., Nechaev Y., Valdenberg A. Design of Telemedicine System Architecture // Healthcare IT management. — 2006. — Vol. 1, No. 2. — P. 31–33.

Bogdanov A. V., Degtyarev A. B., Nechaev Y., Valdenberg A. Design of High-Performance Telemedicine System // Healthcare IT management. — 2006. — Vol. 1, No. 1. — P. 29–31.

Bogdanov A. V., Degtyarev A. B., Lwin S. M., Lwin T. K. Problems of Development of Complex Multilayered Applications in Distributed Environment // Proceedings of the 4th Intern. Conf. “Distributed Computing and Grid-Technologies in Science and Education”. Dubna: JINR, 2010. — P. 51–56.

Degtyarev A. B., Logvinenko Y. Agent system service for supporting river boats navigation // Procedia Computer Science. — 2010. — Vol. 1, No. 1. — P. 2717–2722.

Iakushkin O., Grishkin V. Messaging middleware for cloud applications: Extending brokerless approach // Emission Electronics (ICEE). — 2014. — P. 5–9.