

Methods and Tools for Profiling and Control of Distributed Systems

Roman Sukharev¹, Evgeniy Pluzhnik², Oleg Lukyanchikov^{1,*}, and Dmitry Biryukov^{1,2}

¹Moscow Technological University (MIREA), 119454 Moscow, Russia

²Moscow Technological Institute, 119334 Moscow, Russia

Abstract. The article analyzes and standardizes methods for profiling distributed systems that focus on simulation to conduct experiments and build a graph model of the system. The theory of queueing networks is used for simulation modeling of distributed systems, receiving and processing user requests. To automate the above method of profiling distributed systems the software application was developed with a modular structure and similar to a SCADA-system.

1 Introduction

All complex systems tend to be separated into smaller modules that perform autonomous tasks. Each developed software module may be included in different programs, if the conditions of its use are met, declared in the documentation for this module. Thus, a software module can be considered as a tool to deal with the complexity of the programs, and as a tool to combat duplication in programming. But modular programming has failed to fully implement all its concepts in full [1]. Therefore, with the development of networks and distributed systems, modules became separate network applications, which formed a service-oriented system Software-as-a-Service (SaaS) and more complex cloud systems. However, the complexity of the analysis of such systems also increased, since operations are performed on different computing nodes, and for each service a different load is applied. So today it is important to develop methods and tools for profiling and management of distributed systems.

All existing applications for the analysis of distributed systems are based on the analysis of logs gathered from all services, such as ELK [2, 3] or Dapper [4] by Google. Systems like SCADA (Supervisory Control And Data Acquisition) are preferred for profiling and management of distributed systems. The information systems produce a great amount of operations. In result, the operator is unable to keep up, so it is preferable to only show false and time-consuming operations, this will help finding the bottlenecks in the system and reconfigure it if necessary.

2 Methods for profiling of distributed systems

Today there is no widely accepted methods and tools for profiling of distributed systems, but there is a number of proposed methods in [5]. Both of these methods for analysis and profiling of distributed systems are focused on simulation to conduct experiments and build the graph model of a system. After combining these methods, it is possible to identify the main stages of profiling distributed systems:

To form the graph representation of the process of solving functional problems as a coherent set of query processing.

To build and configure the simulation model of query processing for a specific version of a computer network and the distributed data warehouse.

To carry out the experiment on the simulation model to evaluate the execution time of the queries.

To determine the bottlenecks in the system, dramatically affecting the performance of the distributed system.

In these methods the profiling results depend on the simulation model, which determines the load on the system with a certain requests' flow intensity. When profiling a distributed system using the calculation indicators for monitoring the system's effectiveness, following should be considered:

- a plurality of storage devices in the queuing systems (QS). This factor leads to a consideration of the QS as a system with a total lineup of applications, eliminating the receipt of the individual characteristics of the queues;
- generally non-deterministic time for the service of application unit. The time spent on application maintenance depends on the total volume of the package,

*Corresponding author: luk-it6@yandex.ru

with the result that there is a mutual dependence of the average time of service applications and the number of requests waiting for service;

– correlations between the time of receipt of applications coming from different elementary sources, leading to lack of steady-state flow.

Having calculated performance of individual QS's, it is possible to calculate the performance of stochastic queuing network (SQN).

After building the simulation model, for profiling of a distributed system it is necessary to conduct a series of experiments with different system architectures, in which the number of calculated nodes, their location, data flow, and other parameters change. SCADA-like software will allow automating the construction of these systems, immediate implementation and the collection of data based on the results of the experiment.

3 Development tools for control and profiling

Tools that automate profiling of distributed systems are implemented in hmiSCADA system (<https://github.com/RSukharev/hmiSCADA>), written in C++, using the Qt libraries. This application uses the principle of modular construction. The system consists of a main module, providing a complete set of basic functions for processing and visualization of information, as well as additional functional modules, implementing the data collection functions and optional monitoring and control functions in the application. Additional modules for data collection and management are included in the source code of the profiled system.

The core module's functionality includes: obtaining data from external modules; data visualization; options to manage the software profiling process; simulation.

Additional modules' functionality includes (Fig. 1):

- data collection on the times of start and end of the test piece of code;
- sending profiling data over the network to the main module of the system;
- execution of the commands, received from the visualization and control modules.

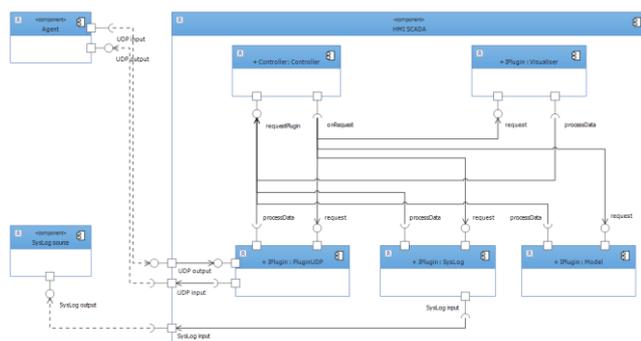


Fig. 1. Structure hmiSCADA

The main difference of this system from other SCADA systems is that to render data received from an operator, it is not required to build the mnemonic of what should be displayed in advance. The mnemonic is

generated automatically based on the data obtained. However, due to the modular structure of the system it is possible in the future to bring the graphical user interface to a common standard. It is also possible to provide compatibility with the data transmission standards for SCADA systems (OPC UA). Enhanced system functionality occurs as new modules are written and connected.

The architecture of the system allows following the process, in which messages may pass through a series of modules, each of which performs its work on data processing and forwards the result further, modifying and extending the functionality of the system as a whole.

An example of profiling the two applications is shown in Fig. 2.

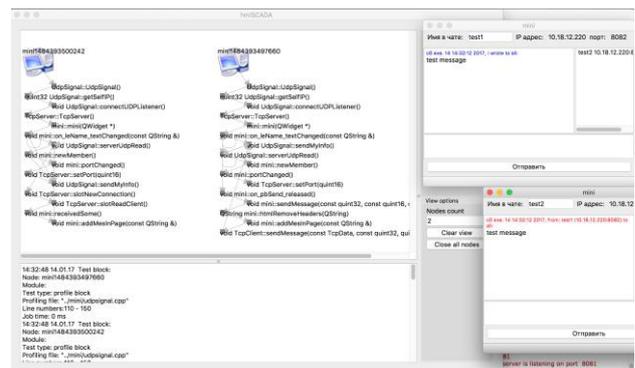


Fig. 2. An example of profiling

4 Conclusion

This article describes the methods and SCADA-like tools for profiling and management of distributed systems. Those allow automating the process of profiling systems, which makes it possible to locate bottlenecks and errors in the system.

An important factor in the development and promotion of these SCADA-like systems is the ability to integrate them into functioning distributed systems. That's why the prototype hmiSCADA proposes modular architecture. With this architecture, third-party developers can develop their modules for hmiSCADA. Costs for the development of these modules are easily compensated by the time saved on profiling and analysis of the system.

References

1. I.B. Bresler et al., Radio industry, **1**. 72-88 (2009)
2. A. Reelsen, *Using elasticsearch, logstash and kibana to create realtime dashboards* (2014)
3. S. Bagnasco et al. Journal of Physics: Conference Series, **608**, 012016 (2015)
4. B.H. Sigelman et al. *Dapper, a large-scale distributed systems tracing infrastructure* (2010)
5. O.I. Lukyanchikov, V.V. Filatov, Engineering Vestnik, **2** (2016)