16thConference on Reliability and Statistics in Transportation and Communication,
RelStat'2016, 19-22 October, 2016, Riga, Latvia

Influence of Data Delivery on Availability of Web Systems

Tomasz Walkowiak*, Wojciech Kamiński

Faculty of Electronics, Wroclaw University of Science and Technology, Wybrzeze Wyspianskiego 27, 50-270 Wroclaw, Poland

Abstract

Meeting the performance expectation of web systems is a challenging task. Data grids, used as an additional tier in n-tier architecture of web applications, allow to speed up access to data. The key process in building of a data grid solution is the initial load, since the default methods of the initial load are very time consuming. Therefore, they have a big influence on lowering system availability. Authors propose a custom adapter that shorts the initial load time and, as a result, enlarges availability of a web system. Authors analysis the default initial load process and comparethe performance of default methods with the adapter developed by them.

© 2017 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the scientific committee of the International Conference on Reliability and Statistics in Transportation and Communication

Keywords: data grid, caching, initial load, availability

1. Introduction

As modern Web applications are complex systems with a wide range of features and access too many data sources, meeting performance expectations of nowadays user is a challenging task. Large enterprise applications are responsible for handling huge amount of data. The modern web application architectures must fulfill these requirements in order to provide high throughput and low latency (Molyne aux, 2014).

There are many different techniques of improving the web application performance (Caban and Walkowiak, 2015). They are using third-party software such as: queue systems, distributed processing tools, resizable compute

---

* Corresponding author.
  E-mail address: tomasz.walkowiak@pwr.edu.pl

capacity in the cloud, caching libraries, etc. or using asynchronous way of handling incoming requests (Palczynski, Walkowiak, 2015).

The most important element in many organizations is the data they collect each day. Working with data and accessing various types of information is an important factor of day-to-day work activities. In many enterprises data is spread across many different databases, what cause difficulties to access them in standard way. Many applications also use data that is stored in different data sources at one time. Access to this data very often take more time as it need to be read from storage devices. This problem can be solved by implementing in memory data gird (Colmer2011). The data grid is a data structure that entirely exists in the random access memory and is spread across many servers. As the data resides entirely in the memory, access to it is much faster than to classic data stored on mass storage media such as hard disks.

However, each data grid (as discussed in section 3) requires the initial load of data.The default implementations of initial load aretime consuming (see Fig. 6). Therefore, as it is shown in section 2, the initial load has a big influence of the system availability. Therefore, authors analyzed (section 4.1) delivery methods to data grid and improved its performance by the custom adapter (section 4.2). In section 5 performance of initial load methods areanalyzed. Next,multithread aspects of initial load are investigated. It is followed by summary and final remarks.

2. System availability

One of the system quality measure, important from the business and administrator perspective, is the availability function $A(t)$. It is defined as the probability that the system provides correct responses at the specific time t. In stationary conditions, the function is time invariant. Availability is then characterized by a constant coefficient, denoted as A. It could be calculated as a ratio of the expected value of the uptime (t_{up}– time when system operates) over the downtime (t_{down} time when system is not operating):

$$A = \frac{E[t_{up}]}{E[t_{up}] + E[t_{dwon}]} \ . \tag{1}$$

Both up- and downtimes have influence on the system availability. In this paper, we focus only on decreasing of the downtime (to achieve higher availability).

Modern web systems are deployed in the virtualized environments where operating systems are independent from the hardware so they can be easily moved to an-other server (Portnoy, 2012). This allows improvingavailability of the systems by moving affected by failure system or software components to other hosts. The process could be automatized, so in case of a failure the affected part of the system is restarted on the same or other server. Therefore, the system downtime is equal to the time of restarting moved system components. Starting of the operating system and web servers takes usually less then few minutes. However, as it is shown in Fig. 6, moving 20 million of records to data grid using the default initial load technique takes more than 3 hours. It is a very long time. Therefore, the decrease of the initial load time is an important factor to achieve higher availability.

3. Data grid

3.1. Data grid as additional tier in n-tier architecture

N-tier architecture is an architecture pattern in which every key component is separate tier (Schuldt, 2009). It allows building modular application. Every module can be changed without changing other modules. The most popular *n*-tier architecture example is the three-tier architecture: presentation, logic and data tier. Data tier could contain many databases from which logic tier can obtain data and deliver them to presentation tier. In large system using load of data, the time needed for communication between logic tier and data tier could be bottleneck. Solution for that could be in memory data grid (Waehner, 2014). It could work as cache layer that provide data to logic tier. It provides many advantages (Hult, 2014), like fast access to data, unified access and simple scale-out. An important