Storage QoS Management in a Distributed Data Sharing and Archiving System

Darin Nikolow (1), Renata Słota (1) and Jacek Kitowski (1,2)

(1) AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Computer Science, Kraków, Poland
(2) AGH University of Science and Technology, ACC Cyfronet AGH, Krakow, Poland
emails: {darin, rena, kito}@agh.edu.pl

The scientific applications of the fourth paradigm [1] of science deal with large amounts of data, which can be stored in various storage devices or systems taking into account the possibility of sharing and archiving. Some of the requirements posed to those storage systems may concern the Quality of Service (QoS) aspects. Examples of such requirements are: specifying the minimal data transfer rate, the maximal data access latency, the minimal level of data protection and availability. These requirements are formally expressed in Service Level Agreement (SLA) being part of the contract between the client and the service provider.

The modern storage systems, used by the mentioned applications, are usually distributed storage systems, where the load is distributed among some number of Storage Nodes (SN), organized in a common filesystem namespace, e.g., Lustre, GlusterFS. The QoS management in such storage systems is a challenging task given the possible storage device heterogeneity, the dynamically changing data access patterns, the client’s concurrency and storage resource sharing. The problem gets even more complicated when distributed computing environments with virtualized and shared resources like Clouds [2] are considered [3].

In this paper we present our research concerning storage QoS management done within the NDS2 project [4]. The NDS2 project concerns the building of data storage and archiving system spread over the main academic computer centers in Poland. The stress is put on the high level of data availability, data protection and data security.

The proposed approach for storage QoS management is shown in Figure 1. The figure presents a part of the NDS2 system, where only the elements directly involved in the storage QoS management process are shown. The approach is based on the active monitoring of the QoS related low level parameters, which is done by the Live Daemon (LD). The parameters are obtained from the SNs and used for calculating the QoS metrics, which are put in a database. The SLA Monitor uses the QoS metrics and scans the logs file to calculate the SLA parameters and checks if the SLA is not violated. If a violation is discovered the monitor send an alert to the Quality Management Daemon (QMD), which takes the appropriate action for minimizing the possibility of the given violation. The actions, which can be taken by QMD includes repairing a bad replica, creating a new replica or speculative staging of selected replicas. In order to complete these actions the QMD communicates with the Replica Daemon (RD) and the Meta Catalog (MC).

Another method of storage QoS management is via changing the coefficients of SLA policies, which specify by means of formulas the SN preference for the given data transfer. The SN preference is calculated by the Data Daemon (DD) before selecting the SN for the given data transfer. The coefficients can be changed manually by the administrator or automatically by the QMD (not shown in the picture). The SLA limits for the given class of users is set via the SLA portlet and kept in the User Management System (UMS).

The preliminary test results showing the accuracy of SLA monitoring as well as the effect of QoS management in terms of experienced performance by the client application will also be presented in the paper.

1 This work is funded from NCBiR “KMD2” project (no. NR02-0025-10/2011).
Full Polish name of the project: System bezpiecznego przechowywania i współdzielenia danych oraz składowania kopii zapasowych i archiwalnych w Krajowym Magazynie Danych
Acknowledgments

This work is supported by Polish MNiSW grant nr N R02 0025 10. AGH-UST grant is also acknowledged.

References


