

DARIN NIKOLOV
RENATA SŁOTA
DANILO LAKOVIC
PAWEŁ WINIARCZYK
MAREK POGODA
JACEK KITOWSKI

MANAGEMENT METHODS IN SLA-AWARE DISTRIBUTED STORAGE SYSTEMS

Abstract

Traditional data storage systems provide access to user's data on the "best effort" basis. While this paradigm is sufficient in many use cases it becomes an obstacle for applications with Quality of Service (QoS) constraints. Service Level Agreement (SLA) is a part of the contract agreed between the service provider and the client and contains a set of well defined QoS requirements regarding the provided service and the penalties applied in case of violations. In the paper we propose a set of SLA parameters and QoS metrics relevant to data storage processes and the management methods necessary for avoiding SLA violations. A key assumption in the proposed approach is that the underlying distributed storage system does not provide functionality for resource or bandwidth reservation for a given client request.

Keywords

Service level agreements, mass storage systems, quality of service

1. Introduction

Traditional data storage systems provide access to the user's data on the "best effort" basis. While this paradigm is sufficient in many use cases it becomes an obstacle for applications with Quality of Service (QoS) constraints. These applications are sensitive to the QoS of the data stream and can fail or not behave correctly if the QoS constraints are not met. An example of QoS sensitive application is a video player displaying video sequences from a data stream. Another example is recording events, e.g. from scientific experiment, coming at certain rate. More often users of storage services require storage QoS, e.g., for fast recovery of lost data in business critical services.

Service Level Agreement (SLA) is a part of the contract agreed between the service provider and the client and contains a set of well defined QoS requirements regarding the provided service and the penalties applied in case of violations. If the provider does not redeem the SLA then additional costs are generated thus appropriate management is highly desirable.

In a multiuser SLA-aware environment clients may have different SLA/QoS requirements, which makes the management more complicated. The successful implementation of SLA in distributed environments strongly depends on proper definition of meaningful SLA parameters and QoS metrics [5] and the relevant monitoring of these metrics as well.

In the paper we propose a set of SLA parameters and QoS metrics relevant to data storage processes and the management methods necessary for avoiding SLA violations. A key assumption in the proposed approach is that the underlying distributed storage system does not provide functionality for resource or bandwidth reservation for a given client request. The management methods for achieving the best possible SLA fulfillment rely on monitoring of the QoS metrics for identifying SLA violations threats and are based on replication techniques and parallel file transfers for increasing the bandwidth for the threatened requests or for increasing the data protection level of the files which have accidentally lowered redundancy.

The proposed methods are considered for implementing in the NDS2 project [4]. The previous version of NDS implemented prediction and load balancing techniques to address the performance aspects [6]. The goal of the NDS2 project is to provide backup, archiving and general data storage services at high security and data availability levels using geographically spread storage nodes and advanced encryption techniques.

The rest of the paper is organized as follows. The next section presents the state of the art. The proposed SLA parameters, management system concept and management methods are described in section 3. Section 4 concludes the paper.

2. State of the art

There are research studies concerning QoS and SLA aspects for data storage systems. Some of them are presented in this section. In [7] a framework supporting QoS built upon the Ceph object based distributed storage system is proposed. The framework, which is called Bourbon, allows for isolating of storage performance between different classes of workloads. In [1] a QoS extension to the ext3 filesystem is proposed. The mechanism relies on matching QoS user requirements, which are given by hints, to storage system profiles provided by the administrator. Based on the appropriate storage system or device is selected for allocating the user data. Kim et al. in [3] describe a filesystem called Apollon, which is capable of handling mixed workloads requiring storage QoS. Their implementation relies on a deadline-driven I/O scheduler and provides admission control for the requests. In [2] a virtual storage system supporting QoS is presented. The clients of the system are provided with the abstraction of hard disk drive available via the iSCSI protocol. Performance management using replication techniques is also introduced. The presented storage QoS solutions do not provide the methods for automatic storage resource management in distributed storage systems needed to deal with the storage SLA/QoS aspects in the context of NDS2, which motivated us to start this research.

3. Quality of service management concept

In this section the concept of storage resource management system for SLA-aware distributed storage is presented. The used approach strongly relies on monitoring of performance related parameters of storage resources. A meaningful definition of SLA parameters and identifying the relevant QoS metrics are required to successfully implement the proposed management methods.

The concept of QoS management in the context of NDS2 project is shown in Figure 1. The figure shows a part of the NDS2 system tightly related with the QoS management.

Generally, NDS2 consists of:

- **Storage Systems**, where the files are physically stored,
- **Storage Nodes (SNs)**, allowing access to the storage systems,
- **Access Nodes (ANs)**, which are responsible for serving client requests,
- **Meta Catalog (MC)**, holding the meta data (not shown in the figure),
- **Live Daemon (LD)**, which monitors the status of other components (not shown in the figure),
- **Data Daemon (DD)**, providing access to data via virtual filesystem,
- **Replica Daemon (RD)**, which is responsible for creating and removing of replicas (not shown in the figure),
- **SLA Monitor**, which monitors QoS metrics and SLA parameters,

- **Quality Management Daemon (QMD)**, which provides hints or commands to other components for keeping the SLA fulfilled.

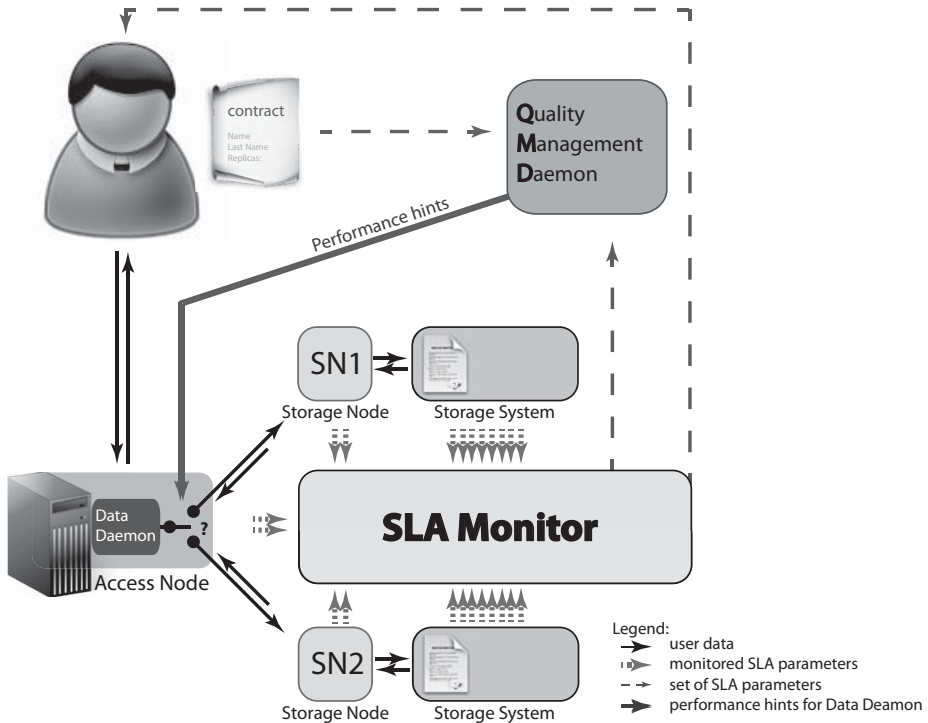


Figure 1. Concept of SLA-aware distributed storage system.

The Data Daemon (DD) is the heart of the NDS2 system realizing the functions of providing access to the underneath storage via a virtual filesystem. Files in NDS2 are replicated and encrypted by default to increase the data protection and security. DD chooses which replica to use for data access based on some predefined policy, which is chosen based on the user requirements. When a file in the NDS2 system is accessed the DD needs to decide which replica should be used. This decision is supported by hints from QMD, which constantly checks for SLA violations and tries to figure out (according to the proposed management methods – see Section 4) what actions can be performed to avoid these violations. QMD receives the necessary data from the SLA monitor, which is an essential element in the QoS management concept. The SLA monitor gathers low level monitoring data from the storage systems and calculates the QoS metrics necessary to check if the SLA is fulfilled. It should be mentioned that the approach for dealing with the performance related SLA violations is a “best effort” one since no resource (or bandwidth) reservation is supported.

3.1. SLA parameters

The proposed SLA parameters used to specify the user's QoS requirements for the data access service are shown in Table 1. These parameters are defined in accordance with the purpose of the mentioned NDS2 service in which the stress is put on data protection and data security.

Table 1
SLA parameters.

SLA param name	Unit	Description	QoS metrics
Performance parameters			
ReadTransferRate	MB/s	Average transfer rate when reading a file	UserReadTransferRate
WriteTransferRate	MB/s	Average transfer rate when writing a file	UserWriteTransferRate
ReadLatencyTime	s	Latency time for read access	UserReadLatencyTime
WriteLatencyTime	s	Latency time for write access	UserWriteLatencyTime
Data protection parameters			
DataProtectionLevel	n/a	Level of protection from data loss	NrOfReplicas, SNRedundancyLevel
DataSecurityLevel	n/a	Level of protection from unauthorized access to data	CypherKeySize
Availability parameters			
DataAvailabilityLevel	n/a	Level of data availability	UserDataAvailability
ServiceAvailabilityLevel	n/a	Level of service availability	UserServiceAvailability

The parameters are divided into 3 classes concerning performance, data protection level and service availability respectively. The SLA parameters are defined at the user level and in order to be measurable they need to be mapped to an expression containing one or more low level QoS metrics. Most of the SLA parameters map into a single QoS metric except the DataProtectionLevel parameter, which is calculated using the following equation:

$$DataProtectionLevel = \sum_{i \in N} (1 + SNRedundancyLevel_i), \quad (1)$$

where N is a set of SNs on which replicas of a file are stored. The number of SNs in this set is determined by the NrOfReplicas QoS metric. $SNRedundancyLevel_i$ is the redundancy level (see Table 2) of the i -th SN.

The relevant QoS metrics are given in Table 2. The metrics are divided into groups concerning: user side metrics, SN metrics, network metrics, configuration

metrics. The methods of obtaining metric values vary. Some of them are obtained using operating system utilities, e.g., `SNCurrentReadTransferRate`. Another method is log analysing, e.g., `UserReadTransferRate`. Some metrics are taken from configuration files or database specifying the user profile and some metrics, the aggregated ones, are calculated based on other QoS metrics (see Equations (2) and (3)).

SNLoad is calculated using Equation (2). This metric presents the load of a SN regarding its storage bandwidth.

$$SNLoad = \frac{SNCurrentReadTransferRate}{SNMaxReadTransferRate} + \frac{SNCurrentWriteTransferRate}{SNMaxWriteTransferRate} \quad (2)$$

The *SNRedundancyLevel* metric is calculated differently for HSM (Hierarchical Storage Management) systems and for RAID (Redundant Array of Independent Disks) systems (see Equation (3)).

$$SNRedundancyLevel = \begin{cases} NrOfCopies, & \text{for HSM} \\ \frac{NrOfRedundantDisks}{NrOfAllDisk - NrOfRedundantDisks'}, & \text{for RAID} \end{cases} \quad (3)$$

4. Storage-related management methods for SLA

The following use cases, in which automation of management could be provided, have been identified:

- SN overloading,
- Increased access rate to a shared file,
- Permanent damage of data storage medium,
- Single data stream need high throughput (more than the available throughput of any available single storage system).

The goal of management for SLA is to minimize the costs associated with SLA violations.

4.1. SN overloading

In this use case many data access requests are issued to the same SN, which in turn causes that this SN gets overloaded and the QoS requirements for some of the requests are not met. In this case we propose using of replication techniques to cope with the overloading. In the NDS2 project each file is replicated to at least one additional SN, which is located at another site. Obviously, this alternate SNs can be used to relieve the overloaded SN.

Table 2
QoS metrics.

QoS metric name	Description	Monitoring method	Location
User metrics			
UserReadTransferRate	Average transfer rate when reading a file	Instrumenting NDS2 software with additional measurement code or log analyzing	Access servers, client's applications
UserWriteTransferRate	Average transfer rate when writing a file		
UserReadLatencyTime	Read latency		
UserWriteLatencyTime	Write latency		
UserDataAvailability	Data availability		
UserServiceAvailability	Service availability		
SN metrics			
SNCurrentReadTransferRate	Current transfer rate for data read	OS tools (/proc/diskstats)	Concrete SN
SNMaxReadTransferRate	Maximal transfer rate for data read	Obtained from the monitoring database	
SNCurrentWriteTransferRate	Current transfer rate for data write	OS tools (/proc/diskstats)	
SNMaxWriteTransferRate	Maximal transfer rate for data write	Obtained from the monitoring database	
SNLoad	Load of SN	Complex metric – equation (2)	
SNIOps	Number of IO operations per second	OS tools (/proc/diskstats)	
SNRedundancyLevel	Redundancy level	Complex metric – equation (3)	
SNAvailability	Availability level	Read from LD	
Network metrics			
NetworkTransferRate	Current throughput between two nodes	periodic active measure	two NDS2 nodes
NetworkLoad	Current load of a network interface	OS tools	Concrete node
Configure metrics			
CypherKeySize	The key size for encryption	User profile	User profile
NrOfReplicas	Number of NDS2 replicas		

4.2. Increased access rate to a shared file

In the use case when a shared file gets popular for some reason, which is manifesting by an increased access rate, the SN which is hosting the popular file may not be able to serve all of the requests to the file at the desired transfer rate. Again, the usage of replication is proposed to distribute the load among more SNs. The problem of optimal selection of SNs to which the file should be replicated is not a trivial task since many factors can influence the transfer rate. Some of these factors are: current trend of storage and network load, possibly changing request's origin, size of the file, overall QoS constraints, cost factors.

4.3. Permanent damage of data storage medium

Damages of storage media decrease the data availability level specified in the SLA. A quick system reaction should bring the data redundancy back to the desired level to prevent or rather minimize the probability of data loss. When such a situation is detected the system checks which files have had replicas on the fault medium and the appropriate replication requests are issued.

4.4. Single data stream need high throughput

In some cases, e.g., disaster recovery, there is a need to read the data at rates, which are higher than the rate a single SN can provide. In this case the file is previously replicated and transferred using parallel file transfers from all replicas. An essential assumption here is that the network bandwidth available to the client is higher than the transfer rate capabilities of the SNs.

5. Conclusion and future work

In this paper storage QoS related management methods for SLA-aware distributed storage environments in the context of the NDS2 project have been presented. The methods strongly relies on monitoring for identifying SLA violations threats. Relevant SLA parameters and QoS metrics have been defined allowing for unequivocally specifying of user's storage QoS requirements. The analyses showed that for such systems the focus of the QoS management would be in the field of data protection and availability, while the performance aspects would be addressed on the best effort base. The applying of storage performance management with respects of QoS requires complete architecture re-thinking in which the storage performance management is an issue at every level of the system – starting from the low device/driver levels, through the mid levels of filesystem, and ending at global distributed system level with storage and network resource reservation. Our future work will concentrate on extending our methods with support for storage performance management.

Acknowledgements

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

References

- [1] Franciosi F., Knottenbelt W.: Data allocation strategies for the management of quality of service in virtualised storage systems. In *Mass Storage Systems and Technologies (MSST), 2011 IEEE 27th Symposium on*, pp. 1–6, May 2011.
- [2] Huang H. H., Grimshaw A. S.: Design, implementation and evaluation of a virtual storage system. *Concurr. Comput. : Pract. Exper.*, 23:311–331, March 2011.
- [3] Kim T., Won Y., Kim D., Koh K., Shin Y.: Apollon : File system level support for QoS augmented I/O. In *Advances in Multimedia Information Processing – PCM 2005*, Lecture Notes in Computer Science, chapter 6, pp. 59–70. 2005.
- [4] National data storage project. <http://nds.psnc.pl>.
- [5] Paschke A., Schnappinger-Gerull E.: A categorization scheme for SLA metrics. In Schoop M., Huemer C., Rebstock M., Bichler M., editors, *Service Oriented Electronic Commerce*, volume 80 of *LNI*, pp. 25–40. GI, 2006.
- [6] Słota R., Nikolow D., Polak S., Kuta M., Kapanowski M., Skalkowski K., Pogoda M., Kitowski J.: Prediction and load balancing system for distributed storage. *Scalable Computing. Practice and Experience*, 11(2):121–130, 2010.
- [7] Wu J. C., Brandt S. A.: Providing Quality of Service Support in Object-Based File System. In *Proceedings of the 24th IEEE Conference on Mass Storage Systems and Technologies*, pp. 157–170, Washington, DC, USA, 2007. IEEE Computer Society.

Affiliations

Darin Nikolow

AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Computer Science, al. Mickiewicza 30, 30-059 Krakow, Poland, darin@agh.edu.pl

Renata Słota

AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Computer Science, al. Mickiewicza 30, 30-059 Krakow, Poland, rena@agh.edu.pl

Danilo Lakovic

AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Computer Science, al. Mickiewicza 30, 30-059 Krakow, Poland

Paweł Winiarczyk

AGH University of Science and Technology, ACC Cyfronet AGH, ul. Nawojki 11, 30-950 Krakow, Poland

Marek Pogoda

AGH University of Science and Technology, ACC Cyfronet AGH, ul. Nawojki 11, 30-950
Krakow, Poland

Jacek Kitowski

AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics,
Computer Science and Electronics, Department of Computer Science, al. Mickiewicza 30,
30-059 Krakow, Poland,
AGH University of Science and Technology, ACC Cyfronet AGH, ul. Nawojki 11, 30-950
Krakow, Poland, kito@agh.edu.pl

Received: 9.12.2011

Revised: 18.05.2012

Accepted: 9.07.2012