Computing and Informatics, Vol. 31, 2012, 463–479

MANAGEMENT OF DATA ACCESS WITH QUALITY OF SERVICE IN PL-GRID ENVIRONMENT

Renata Słota, Darin Nikolow, Kornel Skałkowski
Jacek Kitowski

AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Electronics
Department of Computer Science
Al. A. Mickiewicza 30, 30-059, Krakow, Poland
&
AGH University of Science and Technology, ACC Cyfronet AGH
Ul. Nawojki 11, 30-950 Krakow, Poland
e-mail: rena@agh.edu.pl

Communicated by Aleksander Byrski

Abstract. e-Science applications increasingly require both computational power and storage resources, currently supported with a certain level of quality. Since in the grid and cloud environments, where we can execute the e-Science applications, heterogeneity of storage systems is higher than that of computational power resources, optimization of data access defines one of challenging tasks nowadays. In this paper we present our approach to management of data access in the grid environment. The main issue is to organize data in such a way that users requirements in the form of QoS/SLA are met. For this purpose we make use of a storage monitoring system and a mass storage system model – CMSSM. The experiments are performed in the PL-Grid environment.

Keywords: Data management, Service Level Agreement, Quality of Service, PL-Grid environment

1 INTRODUCTION

Modern e-Science applications require not only computational power but also a certain level of quality for the data access. The latter feature is a topic of interest in newly established distributed computing environments, like PL-Grid [1]. The goal of the PL-Grid project is to provide the Polish scientific community with resources to enable e-Science research in various fields. To make the operation easier and to optimize resources usage the concept of Virtual Organizations (VO) is introduced, with special attention to data management for any VO. Automatic configuration and creation of VO is challenging since Service Level Agreement (SLA) parameters need to be guaranteed. A subset of these parameters may concern Quality of Service (QoS) of the storage systems where the users keep their data, especially if the users are interested in running applications, which use large amount of data. Construction of such an environment which meets these demands is a serious task because of the heterogeneity of storage resources. Modern grid environments, as well as Clouds, have to take into account also the high dynamics of the virtualized resources.

In the grid and cloud environments there are various types of storage nodes (SNs) with constantly changing load. The SNs may use divergent storage systems (disk-based, tape-based, solid state disk, disk array, multi-tiered storage) giving wide range of performance and capacity capabilities. On the other hand there are user requests for data access, which need to be served with a certain QoS. So the problem which arises is how to schedule the access to storage resources, how to allocate the available performance and capacity, and how to organize data in such a way that users requirements in the form of QoS/SLA are met while maintaining effective use of storage resources. Accurate monitoring service is essential for successful deployment of this approach. For each storage node essential performance related parameters need to be monitored, so that current and future performance capabilities can be obtained and used in the decision process of selecting appropriate location of data. Another aspect which should be considered is to create an appropriate model, which defines a common set of essential performance related parameters and a common interface. Integration of various monitoring systems is easier if the common model for various storage systems is developed. Semantic-based solutions can be applied to deal with the heterogeneity of storage systems and their relevant QoS metrics.

In this paper we present our approach to data access management in the Grid environment with QoS by making use of a knowledge supported storage monitoring system and a mass storage system model – Common Mass Storage System Model (CMSSM), originally developed within OntoStor [2] project for such purposes. The model defines a set of classes representing mass storage system with storage performance related parameters in more detail. The storage system monitoring and storage resource scheduling are both essential for the successful operation of the PL-Grid and for fulfilment of SLA. Implementation of these services based on the model allows for better interoperability and flexible integration with other grid components. We show example of data access quality metrics specified and obtained by appropriate monitoring utilities and used for data management with QoS. The proposed approach of data management with QoS could be used, for example, in application storing data coming from HEP experiments [4] or in multi-player on-line games [3].

The rest of the paper is organized as follows: the next section presents state of the art focused on QoS, SLA fulfilment and relevant ontology. Description of the problem of data management with QoS and our idea of a relevant architecture are given in Section 3. Next, the adopted methods and technologies helping in achieving QoS in data management are described in Section 4. Section 5 describes the QoS data management policies implemented within the PL-Grid project. Section 6 shows test results and the last section concludes the paper.

2 STATE OF THE ART

Grid and Cloud computing paradigms become more useful in practice if they can provide stable and easy to use environments. These environments have reached a certain level of maturity where QoS provisioning becomes a common functionality. For service provisioning QoS becomes more and more important in respect of universality of Service Oriented Architecture (SOA) paradigm and construction of applications with services. Considering further that the above-mentioned applications are e-science applications accessing large amount of data, further in the paper called Storage Intensive Applications (SIA), then the problem of storage QoS needs careful attention. Storage QoS may be specified (like other quality requirements) in the form of SLA, defining a real challenge of its accomplishment in the environment with heterogeneous mass storage systems.

There are numerous scientific activities focused on QoS and SLA fulfilment in Grid and Cloud computing environment as well as on data management with QoS. These studies concentrate on the following:

1. building VO with contract negotiation and definition of SLA as a part of the contract [5, 6],
2. definition of SLA and their decomposition into low level parameters for SLA monitoring [7, 8],
3. monitoring of QoS/SLA parameters for services and workflows [9],
4. attempt of resource management for QoS and SLA accomplishment [10, 11, 12],
5. adopting of QoS data management in the fields like medicine [13, 14], real-time [7] and multimedia [15] applications, in which the data is an important issue.

However, these studies do not go deeply enough to consider the heterogeneity of the storage systems and therefore the problem of monitoring of such systems as well as their management for guaranteeing storage QoS are still a challenging task.

Semantics can be applied to assist in solving these problems. There are a few results of the investigations on this subject. One of them is the QoS ontology [16], which is aimed on a description of Web Services QoS aspects and allows to define different QoS metrics for the services. This ontology is quite simple and rather inextensible on domains different than Web Services one. Another ontology for describing the QoS aspects in services-based systems – the QoSOnt ontology [17] – is comprehensive and extendable. It constitutes a stub ontology, which requires adaptation to the domain under study (e.g., definition of metric types, definition of units associated with a described domain, etc.). In the presented paper this ontology is adopted for the description of the storage resources QoS aspects reported below (see Section 4.2).

We propose the CMSSM model [18] of heterogeneous storage systems, which allows for constructing of relevant ontology and semantic description of storage resources and therefore allows for developing of the appropriate SLA monitoring. In this paper we describe an attempt to adopt CMSSM for the need of data management with QoS in the PL-Grid environment.

3 DATA MANAGEMENT WITH QOS

SIA running on the Grid have specific requirements concerning quality of data access, called further Storage QoS (SQoS). The main parameters describing the SQoS are: available storage space, latency and transfer rate. SIA can have various data access paradigms: *steady* (burst) transfer (where all application's data is read or written at once), or *start-stop* transfer for which the data is accessed in regular portions and there is some wait interval between them (caused by the process of preparing, e.g., the next data portion by simulation), or *random* access (random data portion, random wait interval, e.g., interactive applications, servers/services, etc). Strict SQoS implies resource wasting since resource reservation is made for a given task and the resource cannot be used by other tasks even if it is not currently in use. With some acceptable uncertainty of SQoS more tasks could be served. The main goal is to achieve high storage utilization while keeping SQoS requirements for each request with a certain (almost 100 %) probability. In a virtualized environment the performance of shared storage resources is hard to predict. In order to meet SQoS requirements information about the current storage resource performance utilization (from monitoring) as well as information about the scheduled storage requests (from storage resource performance manager) are needed.

Our idea of such data management is presented in Figure 1. Application data is kept in a set of heterogeneous Storage Nodes (SNs). On every storage node a storage Monitoring Agent (MA) is running. Storage MAs send monitoring data (compatible with CMSSM) to the monitoring system. Storage Resource Performance Manager (SRPM) manages the available performance, assigns SN to a storage request and keeps track of scheduled storage requests. If there is no available storage performance then the request is postponed. Job scheduler asks SRPM for storage resource with storage performance capabilities, which are required by a client. SQoS watcher observes if the transfers are served with the desired SQoS and informs administrator or other management component/service or application when SQoS is not fulfiled so an action to cope with the problem could be initiated.

SRPM can apply various algorithms for resource assigning like: round robin, random, round robin QoS, max available QoS metric, best matching QoS metric, round robin with scheduling QoS, strict QoS. The round robin strategy just assigns

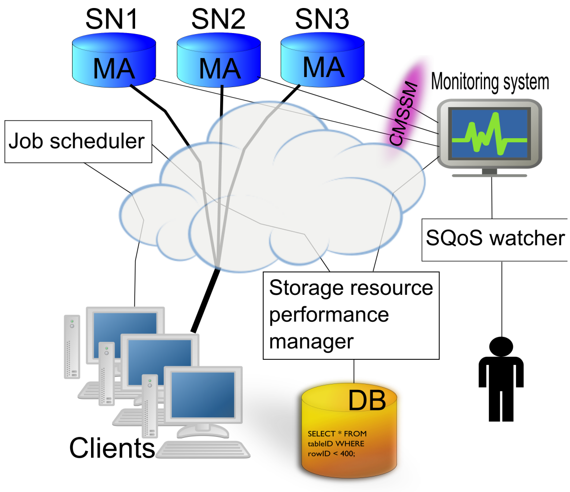


Fig. 1. Data management with QoS

next SN without checking monitoring. The random strategy selects a storage node randomly, no monitoring is used also. The max available QoS metric approach assigns the node with the maximum available QoS metric reported by the monitoring system. None is assigned if there are no available resources for providing the required QoS. The best matching QoS metric algorithm assigns the node with the closest available QoS metric to the required one. The idea here is to use storage nodes effectively especially when various QoS levels are requested simultaneously. The round robin QoS strategy assigns next SN with enough available QoS metric. The strict QoS approach assigns only SNs, which have enough free (unscheduled, not reserved) resources for providing the required QoS. For every data transfer request a bandwidth reservation is done.

In order to realize our approach a monitoring system taking into account the heterogeneity of mass storage systems is needed. The monitoring system must be configured automatically depending on the requirements of the created VO. So it should be possible for auto-selection of relevant monitoring adapters according to concrete type of the managed storage system. These adapters are part of MA and can be implemented as plugins. The system should also provide an unified monitoring layer for parameters independent of the mass storage system. Such functionality of monitoring could be achieved by applying ontology and appropriate model of storage system.

That is why in our solution the monitoring system uses the CMSSM model for unifying the monitoring layer and uses ontology based on this model for describing the storage resources. The semantic approach makes possible the transition from low-level monitoring to high-level QoS/SLA parameters. For this purpose a quality metrics ontology has been proposed based on the QoSOnt ontology. This ontology has been agreed with the CMSSM model.

Next section shortly presents the CMSSM based ontology for storage resource description and then the process of monitoring QoS/SLA parameters using the extended QoSOnt ontology is described in detail. The implementation of SRPM policies in PL-Grid environment is described in Section 5.

4 SEMANTIC BASED QOS/SLA MONITORING

For the purpose of semantic description of storage resources and their QoS aspects two ontologies are used. Storage resources and their attributes are described by the CMSSM ontology. To describe QoS capabilities of the storage resources an enriched version of the QoSOnt ontology [17] is used. The following subsections give detailed descriptions of the ontologies.

4.1 CMSSM Ontology

OntoStor [2] is an ongoing project aimed at developing an ontology-based methodology of data access in Grid environments. Within the project the Common Mass Storage System Model (CMSSM) [18] has been proposed. The model defines a set of classes and performance related properties for the following storage systems: disk arrays, Hierarchical Storage Management (HSM) systems and local disks. It is used by a monitoring system to provide the current state of a storage system in a unified way. Based on this state the storage system performance is predicted. A storage system ontology based on the CMSSM allows for semantic description of storage resources, monitoring and prediction services, which can be used for knowledge based data management with QoS.

4.2 QoS Metrics of Storage Resources

For the purpose of describing general QoS aspects of storage resources the following metrics are proposed:

- Total capacity of a storage resource.
- Mean read transfer rate of a storage resource.
- Mean write transfer rate of a storage resource.
- Minimum read transfer rate of a storage resource.
- Minimum write transfer rate of a storage resource.
- A storage resource MTBF (Mean Time Between Failures).
- A storage resource MTTR (Mean Time To Repair).
- A storage resource reliability.

The above metrics have different meaning for different kinds of storage resources. For example, the total capacity of a disk array or a local disk can be obtained directly from their parameters, but for a HSM system it is calculated as a sum of the disk cache capacity and tapes capacities. The reliability of a disk array results from its RAID level, while the reliability of the HSM system results from a number of copies created on different tapes. These examples show the necessity of a semantic description of the storage resources QoS aspects.

A base of the semantic description of the storage resources QoS capabilities constitutes the QoSOnt ontology. In the domain of the storage resources QoS aspects, the following improvements of the QoSOnt ontology are made:

1. New classes and individuals representing units associated with the storage services are introduced. The introduced classes define a unit of capacity and a unit of transfer, and both inherit from the UnitOfMeasurement class from the QoSOnt ontology. The introduced individuals represent specific units like gigabytes, megabytes per second, etc.
2. Additional individuals of the UnitConversionRate class are defined. The individuals enable an automatic conversion among the defined units.
3. New classes representing QoS aspects of the storage resources are defined. The classes represent the metrics introduced at the beginning of this subsection.
4. New individuals representing methods of obtaining of the defined metrics were added. The individuals have assigned the storage resources attributes from the CMSSM ontology. The attributes are used for the metrics calculations. The direct type of a concrete individual determines the way of a metric calculation (e.g. sum of parameters, mean value, minimum value, etc.).

A relation between the third and the fourth from the above improvements requires more detailed description. The first issue is the difference between a metric and a method of metric obtaining. A metric concept defines a general metric type (e.g. total capacity), a metric unit (e.g. terabytes) and an expected metric value, whilst a method of obtaining defines a concrete way of a metric evaluation (e.g. sum, mean value) and parameters, which are taken to the evaluation. The second issue is the way of assigning a concrete method of obtaining to a concrete metric individual. For solving this issue, the following steps are made:

1. Every general metric class (like total capacity) has subclasses, which constitute its concretes for different storage resources. This is depicted in Figure 2 where the total storage capacity metric class and its subclasses: the disk array total capacity (TotalDACap), the local disk total capacity (TotalLDCap) and the HSM system total capacity (TotalHSMCap) are outlined.
2. Every specific metric subclass (like TotalHSMCap) has assigned a proper method of obtaining individual (e.g. method of TotalHSMCap obtaining has to be a sum of the total disk cache capacity and the tape capacity attributes). Relation of assigning the method of obtaining individual to the specific metric class can be written in the Manchester OWL syntax [19] as follows:

$$\text{Specific metric} \ni \text{Method of obtaining.}$$

3. In the last step a type of a storage resource is assigned to a specific metric class in the following way:

$$\text{Specific metric} \equiv \text{General metric} \cap (\text{General metric} \ni \text{Resource}),$$

where the Resource term means the class of a storage resource (e.g. DiskArray), so for instance the disk array total capacity class is defined in the following way:

$$\text{DATotalCapacity} \equiv \text{TotalCapacity} \cap (\text{TotalCapacity} \ni \text{DiskArray}).$$

The execution of the above three steps and the utilization of an OWL reasoner allow to assign general metrics classes to storage resources individuals, since concrete metric classes and methods of obtaining are automatically assigned by the inference process (the reasoner should have OWL full inference support).

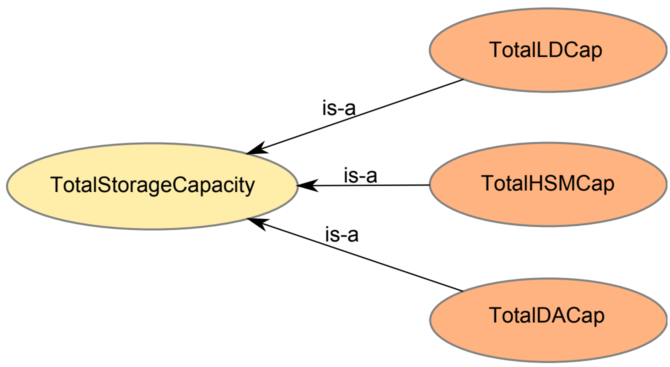


Fig. 2. Subclasses of the total capacity metric class

The above details show how the semantic description of the storage resources and their QoS aspects are made. Another challenge is a proper monitoring of the QoS aspects fulfilment and this is described in the next subsection.

4.3 QoS/SLA Monitoring

Since monitoring of the QoS/SLA metrics is essential for providing the storage services with an expected QoS level, an SLA metrics monitoring system has to be scalable, robust and reliable. Implementation of such a system constitutes the FiVO SLA monitoring system [9] – a part of the FiVO framework [20], which represents a complete system for negotiation and deployment of dynamic virtual organizations.

FiVO SLA monitoring system provides semantic-oriented, efficient and robust distributed monitoring of the SLA metrics based on the Service Oriented Architecture (SOA) paradigm. The system consists of several loosely-coupled services, which communicate with each other by normalized messages. Architecture of the FiVO SLA system deployment is shown in Figure 3.

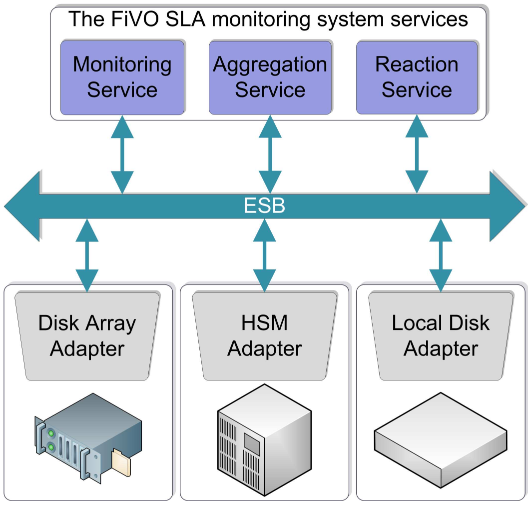


Fig. 3. Architecture of the FiVO SLA monitoring system deployment

A base for the communication among the FiVO system services constitutes the Enterprise Service Bus (ESB) [21]. The ESB provides asynchronous, secure and reliable communication among the system services. It also provides additional features like the Enterprise Integration Patterns (EIP) [22], which are very useful in case of messages passing. Input to the system constitutes an OWL document, which comprises the semantic description of monitored resources and their SLA metrics (presented in subsections 4.1 and 4.2). The system uses the Pellet semantic engine [23] for creation of an inferred ontology model. Results of the SLA metrics monitoring can be browsed by easy-in-use, intuitive web interface.

Among the FiVO system services one can distinguish core services (in Figure 3: Monitoring Service, Aggregation Service, Reaction Service) and Adapter services. The Adapter services are used for monitoring of different resources and they allow to expand the system functionality for monitoring new kinds of resources.

For the purpose of deployment of the FiVO SLA system in the PL-Grid project special adapters for monitoring of storage resources are implemented. The goal of the Adapters implementation is to provide the storage resources parameters according to the CMSSM model described in Section 4.1. The Adapters for local disks and disk arrays allow to monitor all the parameters defined in the CMSSM model and their implementations are generic, so they can be used for monitoring almost all types of disks and disk arrays. The number of parameters from the CMSSM, which are provided by a concrete HSM system adapter implementation depends on the HSM system type, but none of the existing implementations provides all parameters defined in the CMSSM, since the monitoring of the HSM systems parameters is constrained by its vendor and a system interface. Adapters can be implemented using OntoStor sensors library, which makes use of the CMSSM.

5 STORAGE RESOURCE PERFORMANCE MANAGEMENT WITH QOS IN PL-GRID

The storage requests sent to the SRPM, implemented in the PL-Grid environment, contain a set of QoS requirements, which should be fulfiled during the relevant data transfer. Optionally, a set of SNs, which have to be taken into account during the selection process can be specified. Processing of a storage request by SRPM is performed in several steps.

Policy name	**Description**
Round Robin (RR)	SNs are assigned in round robin fashion, no data from the monitoring system is taken into account.
Strict QoS (SQ)	The SQ policy is implemented by scheduling data transfers in such a way that no more than N_{max} concurrent transfers per SN are running. N_{max} is obtained empirically and is set to 2 for the presented tests.
Best Storage Node (BSN)	The current metric value (CV) for SN has to accomplish the following requirement: $CV > \alpha * MV$; where MV is the requested metric value and α is an experimentally chosen coefficient ($\alpha \geq 1$). SN with the greatest value of the difference $CV - \alpha * MV$ is chosen.
Round Robin with Monitoring (RRM)	SNs, which have the current metric value greater than the required one are taken into account, and they are assigned in round robin fashion.
Scheduling with Monitoring (SM)	Every assigned bandwidth for every SN is saved in an internal SRPM database. The current metric value (CV) for SN has to accomplish the following requirement: $CV > \alpha * MV + \beta * BS$; where MV is the requested metric value and BS is the sum of assigned bandwidths for a considered SN, but not yet observed by the monitoring system. α and β are experimentally chosen coefficients, both are greater than 1. SN with the greatest value of the difference $CV - \alpha * MV - \beta * BS$ is chosen.

Table 1. Detailed description of the SN assigning policies

In the first step, an SN assigning policy is selected according to the current user profile. In case of the system tests, described further in Section 6, the policy is specified explicitly by a configuration parameter. In the next step an external knowledge base is queried about the requested QoS metrics. The query result contains semantic definitions of the requested metrics, as it was described in Section 4.2. The definitions are interwined with monitored parameters of storage resources, according to the CMSSM model. In the further step SPRM asks the monitoring system about parameters, which are necessary to calculate the requested metrics. In the last step the required metrics are calculated from the received parameters, and the resulting SN (obtained by the assigning policy) is returned.

In the current version of the system five assigning policies are implemented. The policies are described in Table 1.

For both α and β coefficients different values from the intersection $[1, 2.5]$ were considered. The chosen values should give the best balance between a data transfer launch delay and SLA fulfilment degree.

6 TEST RESULTS

The purpose of the tests is to check SRPM policies used for data access management. The tests checks if the system properly selects SNs for subsequent storage requests with the QoS demand specified as the minimal data transfer rate (the "Minimum write transfer rate" metric, see Section 4.2). The α coefficient has been set to 1.6 whilst the β coefficient has been set to 2 (see Table 1). The quality metric of the tests is the percentage of storage requests with fulfiled SLA (the measured transfer rate $\geq$ requested rate). The testing script works as presented in the following pseudo code:

```
repeat 400 times {
  if (number_of_concurrent_transfers >= max_concurrent_transfers) {
    wait;
  }
  query_for_the_best_node(tested_policy, requested_bandwidth);
  fork_data_transfer_to_the_best_node_and_log_the_obtained_rate;
  sleep;
}
```

For test purposes only small amount of PL-Grid resources has been adopted. The testing environment consists of 3 virtual machines (SNs), in which data can be stored. Each virtual machine has one VCPU and 2 GB of RAM. The virtual machines are hosted on 3 separated physical hosts. The physical hosts possess $2 \times$ Intel Xeon CPU 2.80 GHz and 4 GB of RAM. MA with Local Disk Adapters are installed on each SN. One of the SNs hosts also the FiVO SLA monitoring system.

Round Robin (RR) and Strict QoS (SQ) policies are tested as reference policies, which do not use monitoring (see Table 1).

The following tests with selected policies have been conducted:

- T_1 – testing the RR policy,
- T_2 – testing the SQ policy,
- T_3 – testing the RRM policy,
- T_4 – testing the SM policy.

Two types of workloads have been used for the tests: the first generates data with low rate (63 MB/s) – less than the measured SNs total bandwidth (about 95 %), the second generates data with high rate (100 MB/s) – more than the system can process (about 130 %). The requested minimum write transfer rate has been constant for all tests and equal to 10 MB/s. For each tests the percent of requests with satisfied SLA[1] has been calculated.

				T_1 (RR)		T_2 (SQ)		T_3 (RRM)		T_4 (SM)	
GTT [MB/s]	CT_{max}	S [MB]	NR	TT	%SLA	TT	%SLA	TT	%SLA	TT	%SLA
100	8	500	400	27.5	55	87.7	86	28	58	45	79
63	8	500	400	27	62	59.5	86	28	55	41.7	86

GTT – Generated Total Throughput, CT_{max} – Max number of Concurrent Transfers, S – Size of a data transfer, NR – Number of Requests, TT – Total Throughput, %SLA – percent of requests with satisfied SLA

Table 2. Test results – summary

The results are summarized in Table 2. For the SQ policy SLA (T_2) is satisfied in 86 % of requests. The test has been conducted with the monitoring system running along with the active transfer rate measurements, which probably decreased the performance for some of the transfers. Similar results are for the SM policy (T_4). All tests have been conducted with no generated background transfers. For mass storage with changing performance (e.g. shared storage resources) the SM policy used for the T_4 is expected to be the most advantageous. It should be also mentioned that our testing environment is highly unpredictable because of the virtualization and resource sharing, which causes lower %SLA values.

In Figure 4 the measured values for each test transfer are shown. We can see that for the RRM policy (T_3) (see Figure 4 a)) the transfer rates vary between 1–100 MB/s. Some pendulum effect can be observed due to monitoring – when a storage node is loaded the monitoring senses it during the next measurement, so less or no requests are assigned to this node. On the next cycle the monitoring system senses that the system is not loaded and more requests are assigned to the node. The results for the SM policy, T_4, (see Figure 4 c)) are much better and similar to the SQ policy, T_2, test results (see Figure 4 b)). The last policies have also higher rates for requests with unsatisfied QoS.

7 CONCLUSION AND FUTURE WORK

In this paper we have presented our approach to data management with QoS in a distributed computing environment. We showed how our model (CMSSM) and its relevant ontology can be used for representing the performance state of storage systems and how it can be applied in an intelligent monitoring system. The process of adaptation of the QoSOnt ontology for the purpose of the storage resources QoS parameters description were presented. The paper also shows how a unified set

[1] SLA is satisfied when the obtained transfer rate is greater than or equal to 10 MB/s

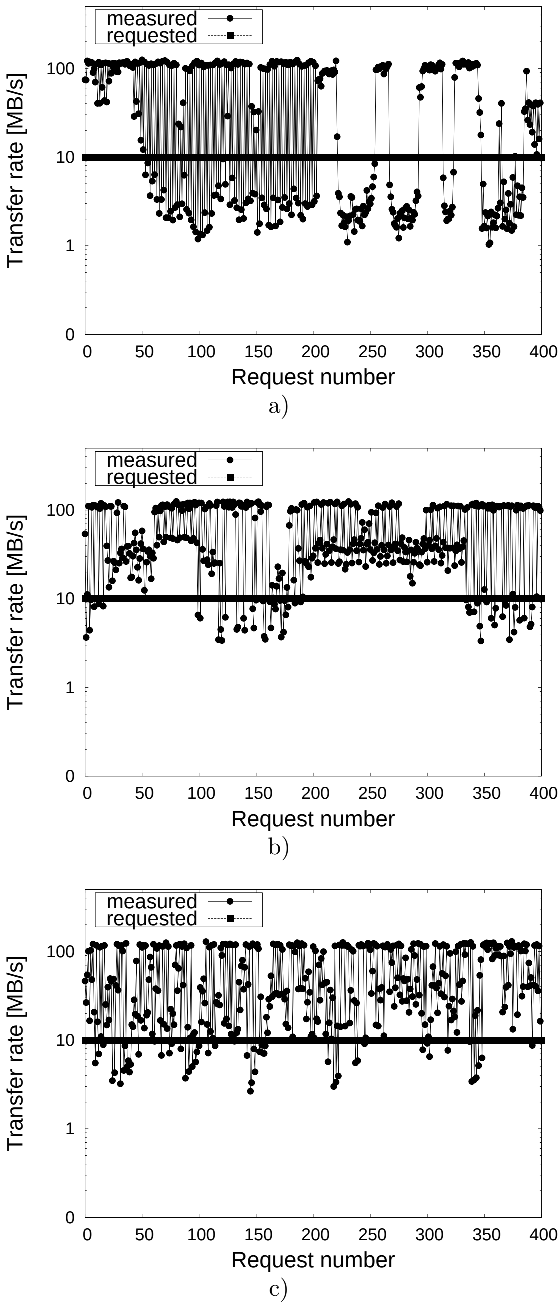


Fig. 4. Test results for selected policies with high load (100 MB/s): a) Round robin with monitoring, b) Strict QoS, c) Scheduling with monitoring

of QoS metrics can be semantically defined in various way for different storage resources. It was also emphasized that such definition of metrics automates their further computations and defining new ones in the system runtime. Test results in virtualized environment showed that best results in terms of satisfying QoS provide the strict QoS and monitoring with scheduling policies.

Our future work will concentrate on increasing the accuracy of the monitoring and the scheduling by taking into account various storage performance parameters available from less invasive operating system utils.

Acknowledgments

This research is partially supported by the PL-Grid project POIG.02.03.00-00-007/08-00 and by Polish MNiSW grant No. N N516 405535. AGH-UST grant No. 11.11.120.865 is also acknowledged.

REFERENCES

[1] The PL-Grid Project web site. Available on: `http://www.plgrid.pl/`, access: 10-07-2010.

[2] The OntoStor Project web site. Available on: `http://www.icsr.agh.edu.pl/ontostor/`, access: 10-07-2010.

[3] Han, J. H.—Lee, D. H.—Kim, H.—In, H. P.—Chae, H. S.—Eom, Y. I.: A Situation Aware Cross-Platform Architecture for Ubiquitous Game. Computing and Informatics, Vol. 28, 2009, No. 5, pp. 619–633.

[4] Funika, W.—Korcyl, K.—Pieczykolan, J.—Skital, L.—Balos, K.—Słota, R.—Guzy, K.—Dutka, L.—Kitowski, J.—Zieliński, K.: Adapting a HEP Application for Running on the Grid. Computing and Informatics, Vol. 28, 2009, No. 3, pp. 353–367.

[5] Kryza, B.—Dutka, L.–Słota, R.—Kitowski, J.: Dynamic VO Establishment in Distributed Heterogeneous Business Environments. LNCS, 5545, pp. 709–718, 2009.

[6] Zuzek, M.—Talik, M.—Świerczyński, T.—Wiśniewski, C.—Kryza, B.—Dutka, L.—Kitowski, J.: Formal Model for Contract Negotiation in Knowledge-Based Virtual Organizations. LNCS, 5103, pp. 409–418, 2008.

[7] Buccafurri, F.—De Meo, P.—Fugini, M.—Furnari, R.—Goy, A.—Lax, G.—Lops, P.—Modafferi, S.—Pernici, B.—Redavid, D.—Semeraro, G.—Ursino, D.: Analysis of QoS in Cooperative Services for Real Time Applications. Data Knowl. Eng., Vol. 67, 2008, No. 3, pp. 463–484.

[8] Tserpes, K.—Kyriazis, D.—Menychtas, A.—Varvarigou, T.: A Novel Mechanism for Provisioning of High-Level Quality of Service Information in Grid Environments. European Journal of Operational Research, Vol. 191, 2008, No. 3, pp. 1113–1131.

[9] Skałkowski, K.—Sendor, J.—Król, D.—Funika, W.—Kryza, B.—Słota, R.—Kitowski, J.: Monitoring of SLA Parameters within VO for the SOA Paradigm. LNCS, pp. 6067–6068, 2010 (to appear).

[10] Qiao, L.—Iyer, B. R.—Agrawal, D.—Abbadi, A.: Automated Storage Management with QoS Guarantee in Large-Scale Virtualized Storage Systems. IEEE Data Eng. Bull., Vol. 29, 2006, No. 3, pp. 47–54.

[11] Stantchev, V.—Schröpfer, C.: Negotiating and Enforcing QoS and SLAs in Grid and Cloud Computing. In GPC '09: Proceedings of the 4^{th} International Conference on Advances in Grid and Pervasive Computing, pp. 25–35, Springer Verlag Berlin, Heidelberg 2009.

[12] Kosar, T.—Balman, M.: A New Paradigm: Data-Aware Scheduling in Grid Computing. Future Gener. Comput. Syst., Vol. 25, 2009, No. 4, pp. 406–413, 2009.

[13] O'Donoghue, J.—Herbert, R.: A QoS Data Management System within a Pervasive Medical System. In Pfeifer, T.; Schmidt, A.; Woo, W.; Doherty, G.; Vernier, F., Delaney, K.; Yerazunis, B.; Chalmers, M.; Kiniry, J. (Eds.): Advances in Pervasive Computing 2006. Adjunct Proceedings of the 4^{th} International Conference on Pervasive Computing. `books@ocg.at`, Volume 207, Vienna, Austria, Austrian Computer Society (OCG).

[14] Stantchev, V.: Service-Based Systems in Clinical Environments. Technical Report 2008-023, UC Berkeley: School of Information 2008.

[15] Goebel, V.—Plagemann, T.: Data Management and QoS in Distributed Multimedia Systems – Towards an Integrated Framework. In 4^{th} Int. IFIP Workshop on Quality of Service (IWQoS 96) 1996, pp. 79–83.

[16] Papaioannou, I. V.—Tsesmetzis, D. T.—Roussaki, I. G.—Anagnostou, M. E.: A QoS Ontology Language for Web-Services. In AINA '06: Proceedings of the 20^{th} International Conference on Advanced Information Networking and Applications 2006, Washington, DC, USA, 2006, pp. 101–106.

[17] Dobson, G.—Lock, R.—Sommerville, I.: QoSOnt: An Ontology for QoS in Service-Centric Systems. Software Engineering and Advanced Applications, 31^{st} EUROMICRO Conference, August 2005, pp. 80–87.

[18] Nikolow, D.—Słota, R.—Kitowski, J.: Knowledge Supported Data Acess in Distributed Environment. In M. Bubak, M. Turala, K. Wiatr (Eds.): Proceedings of Cracow Grid Workshop (CGW '08), Krakow 2009, pp. 320–325, ACC-Cyfronet AGH.

[19] Brief Overview of the Manchester OWL Syntax. Available on: `http://www.co-ode.org/resources/reference/manchestersyntax/`, access: 12-06-2010.

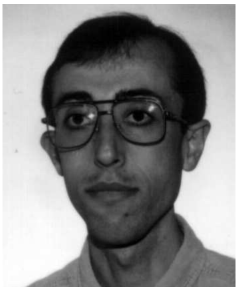
[20] Kryza, B.—Dutka, L.—Słota, R.—Kitowski, J.: Security Focused Dynamic Virtual Organizations in the Grid Based on Contracts. Collaboration and the knowledge economy: issues, applications, case studies, Vol. 5, pp. 1153–1160, 2008.

[21] David, A.—Chappell, D.: Enterprise Service Bus. 1–56, 101–145, 225–232, O'Reilly Media, Inc. 2004.

[22] Hohpe, G.—Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. pp. 99–137, 225–355, Pearson Education, Inc. 2004.

[23] The Pellet OWL Reasoner. Available on: `http://clarkparsia.com/pellet/`, access: 15-06-2010.

Renata Słota works at the Department of Computer Science of the AGH University of Science and Technology in Cracow, Poland. She obtained Ph. D. in 1998 in computer science at the same University. She is the author or co-author of about 90 scientific papers. Her topics of interest include distributed systems, Grid and Cloud environments, data management and storage systems, knowledge engineering. She has been involved in many national and international projects most notably in EU IST: CrossGrid, Pellucid, K-WfGrid, GREDIA, and Int.eu.grid projects.



Darin Nikolow obtained his Ph. D. in Poland at AGH University of Science and Technology in 2003. He is a lecturer at the Department of Computer Science of AGH UST. His research interests include storage systems and distributed computing.

Kornel Skałkowski obtained his M. Sc. at the Faculty of Electrical Engineering, Automatics, Computer Science and Electronics in 2010 from the AGH University of Science and Technology. He is co-author of nine publications concerning the SOA paradigm and storage systems. He has participated in several research projects. His research interests include the SOA-based systems, web technologies and semantic-based software engineering.

Jacek Kitowski is Full Professor of computer science, graduated in 1973 from the Electrical Department of the AGH University of Science and Technology in Krakow (Poland). He is Head of the Computer Systems Group at the Institute of Computer Science of the AGH University of Science and Technolog, researcher at the Academic Computer Centre CYFRONET-AGH, with responsibility on development of high performance systems. He is the author or co-author of about 200 scientific papers. His topics of interest include large-scale computations, multiprocessor architectures, high availability systems, network computing, Cloud and Grid services and Grid storage systems, and knowledge engineering. He participated in many national and international projects funded by European Commission and EDA. He is a member of program committees of international conferences, and evaluator of national and international IT projects. He serves as the Editor-in-Chief of Computer Science Journal and an Associate Editor of Computing and Informatics Journal. He is Expert in EU Program Committee "e-Infrastructures" (Unit F3) and Director of PL-Grid Consortium (NGI Project).