Future Generation Computer Systems: The Int'l Journal of Grid Computing, vol. 23, no. 6, pp. 816-824, July 2007.

StReD : A Quality of Security Framework for Storage Resources in Data Grids

Mais Nijim Ziliang Zong Xiao Qin*
Department of Computer Science
New Mexico Institute of Mining and Technology
801 Leroy Place, Socorro, New Mexico 87801-4796
{mais, zzong, xqin}@cs.nmt.edu

Abstract

Securing storage resources in Grid environments has increasingly become a major concern to make Grids attractive for a wide range of data-intensive applications, because security requirements are often imposed on storage resources to support security-critical applications. However, existing storage systems are unable to dynamically adjust quality of security to meet the flexible security needs of complex data-intensive applications. To remedy this deficiency, we propose in this paper a quality of security framework or StReD for storage resources in Data Grids. In this framework, we integrate quality of security adaptor with an array of security services. Applications running in the framework are enabled to specify flexible security requirements and desired response times of disk requests. The framework leverages the adaptor to dynamically control quality of security for disk requests, thereby achieving good tradeoffs between security and storage system throughput. Experimental results based on a simulated Grid with storage resources show that the proposed framework is capable of significantly improving quality of security and guaranteeing desired response times of disk requests.

Keywords Quality of security, disk systems, desired response time, data-intensive applications, adaptive framework, Grid computing environments

1. Introduction

In the last decade, Grids have become popular for various scientific and commercial applications [27]. A Grid is a collection of geographically dispersed computing and storage resources [17][28] providing services to fit needs of applications like simulation analysis tools [3] and high-energy physics applications [2][15]. A Grid is comprised of five kinds of resources: computation, storage, software, communications, and network bandwidth [26]. Essentially, the

* Corresponding author. *http://www.cs.nmt.edu/~xqin*

Data Grid is defined as an infrastructure that manages large scales data-intensive files and provides computational resources across widely distributed systems [16][26]. Storage systems including hard disks and any storage media are the most commonly used resources in the Data Grids. In this study, we are mainly concerned about securing storage resources like parallel disk systems in the Data Grid.

Parallel disk systems have been widely used in a wide variety of data-intensive applications including but not limited to remote-sensing database systems [4], video surveillance [1], digital libraries [24], and long running simulations [25]. This is mainly because parallel disk systems, which are highly scalable in nature, can alleviate the problem of disk I/O bottleneck. An efficient way of improving performance of disk systems is to make use of I/O parallelisms, which can be achieved by partitioning and distributing data among an array of disks. Disk I/O parallelisms can be provided in forms of either inter-request or intra-request parallelism. While inter-request parallelism enables multiple independent requests to be served simultaneously by a parallel disk system, intra-request parallelism allows a single disk request to be processed by multiple disks in parallel [23].

It is imperative for computer systems to embrace a wide range of security services to support security-critical applications [29][28]. Importantly, security requirements of next generation data-intensive applications are flexible, meaning that it is highly desirable for storage resources in Data Grid computing environments to dynamically adjust quality of security to meet the flexible security needs of complex applications. Providing quality of security for storage resources in Data Grid is challenging, because requests in some data-intensive applications need to be completed within desired response times [6]. In this study, guaranteed response times and high quality of security are two performance goals to be achieved in storage systems. Automatic adjusting quality of security is of a critical issue in development of secure storage systems for Data Grids.

In this paper, we propose a framework for quality of security in the context of storage systems for Data Grids. The framework is by no means limited itself to Data Grids and, thus, we can integrate the framework into any distributed system to make the system adapt to changing security requirements imposed by data-intensive applications. We implement a quality of security adaptor to demonstrate the effectiveness of the framework. Experimental results based

on disk traces show that the proposed framework achieves high quality of security while guaranteeing desired response times of disk requests.

The rest of this paper is organized as follows. Section 2 includes a summary of related work in this area. Section 3 presents a Data Grid model. Section 4 describes the control framework. Section 5 introduces a way of quantifying quality of security. In Section 6, an analytical model is built to estimate the response time of each disk request. Section 7 evaluates the effectiveness of the proposed framework using a simulated disk system. Section 8 concludes the paper with summary and future research directions.

2. Related Work

Security services are important to various data-intensive applications in Data Grids. This is mainly because data stored in storage resources of Data Grids requires special safeguard and protection against any unauthorized access. Samar and Stockinger studied the key security issues of sensitivity of data and unauthorized use of network bandwidth to transfer huge files [21]. Chervenak *et al.* implemented a Data Grid security infrastructure [5], where they developed an authorization and authentication services for the Data Grid storage resources.

The efficiency of parallel disk systems has been extensively researched in recent years because I/O throughput is the major bottleneck for data-intensive application due to the widening gap between processor speeds and disk access speeds [19]. In order to find an optional solution to alleviate the problem of I/O bottleneck, a large body of work has been done on parallel disk systems. Kallahalla and Varman designed an on-line buffer management and scheduling algorithm to improve performance of parallel disks [11]. Kotz and Ellis proposed investigated several write back policies used in a parallel file system implementation [12]. Rajasekaran and Jin developed a practical model for parallel disk systems [16]. Scheuermann *et al.* addressed the problem of making use of striping and load balancing to tune performance of parallel disk systems [23].

Besides efficiency, protecting data in untrusted disk systems is also of critical importance. Now, the issue of security has been paid more attention both experimentally and theoretically in parallel disk systems. A number of cryptographic file systems have been implemented in a way that data is stored in encrypted form [8]. Riedel et al. proposed a framework of core functions to improve the system security level [20]. Huang et al. developed a middleware-oriented Global

Resource Management System [7]. Nahrstedt et al. designed a QoS-aware middleware that can offer a new generation QoS-sensitive applications [13]. Although the above works address applications' QoS requirements in parallel systems, none of them can provide a wide range of security services for parallel disk systems.

We need to find a solution which can quantitively evaluate the security overhead and easily change the security level, because flexible security requirements will be one of the main characteristics of the next-generation parallel disk systems. However, to the best of our knowledge, the way of calculating costs of security service has received little attention. Irvine et al. proposed a model of computing costs for quality of security service [9][10]. Wang et al. presented a security measurement framework, which is based on theory and practice of formal measurements [33]. Their work provided us an insightful view of a future direction of security measurement.

In our previous work, we have proposed an array of security-aware scheduling algorithms for clusters [30][31][32] and Grids [29]. Moreover, we recently developed an adaptive write strategy for local disk systems [14]. In this paper, we propose an adaptive quality of security control mechanism based on our previous work for parallel disk systems. Our scheme can be readily integrated into existing parallel disk systems to substantially improve security of the systems.

3. System Model

In this study we propose a Data Grid model, which can be envisioned as a collection of storage subsystems. We model a Data Grid as a network of *n* storage subsystems with various system architectures. Each storage subsystem consists of data stores and computational facilities. Fig. 1 shows the block diagram of the system model of a Data Grid. A job manager in each storage subsystem accommodates data-intensive jobs submitted to the Data Grid. The job manger aims at tracking load information by periodically changing load status with other job manager in the Data Grid. The incoming data-intensive jobs are placed into a waiting queue managed by a job scheduler in each subsystem.

The complete functionality of the quality of security manager is two-fold. First, the manager chooses the most appropriate security services provided in the security mechanism for disk requests. Second, the manager selects the most suitable security services in a judicious way to guarantee the security and the timely requirements for disk requests.

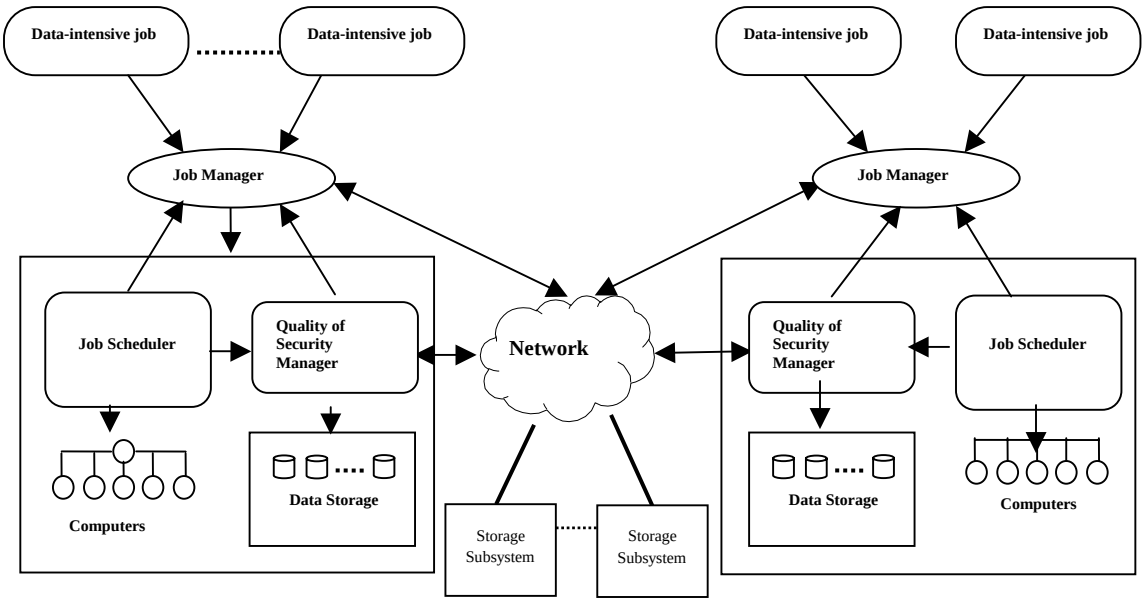

Fig. 1 Block diagram of the system model of a Data Grid.

4. Quality of Security Framework

In this section we propose a quality of security framework, which is a development environment for security-critical applications. The framework is constructed with the goals of providing application programmers with an efficient way to adjust the security levels and the desired response time for their applications in a timely manner to meet both the timely and security requirements. With the framework in place, end-users are allowed to dynamically adjust security levels and desired response times for applications in accordance with changing workload conditions.

The proposed framework for disk systems consists of requirement specification APIs, a quality of security adaptor; a security mechanism, a response time estimator, a low-level disk I/O APIs, and a disk system (see Fig. 2). Disk requests can specify their flexible security and timely requirements using the specification APIs. The APIs are used to specify security and timing parameters, defining how these parameters may be degraded under high system load. The specification API layer, which is an abstraction of underlying security services, are implemented in light of the security services coupled together to form a security mechanism. Disk requests' security and timing requirements, defined as level of security services and desired response

times, are derived from corresponding data-intensive applications. A security level specified by applications may be high, medium, or low. The desired response time of a disk request is the time at which the request must be completed.

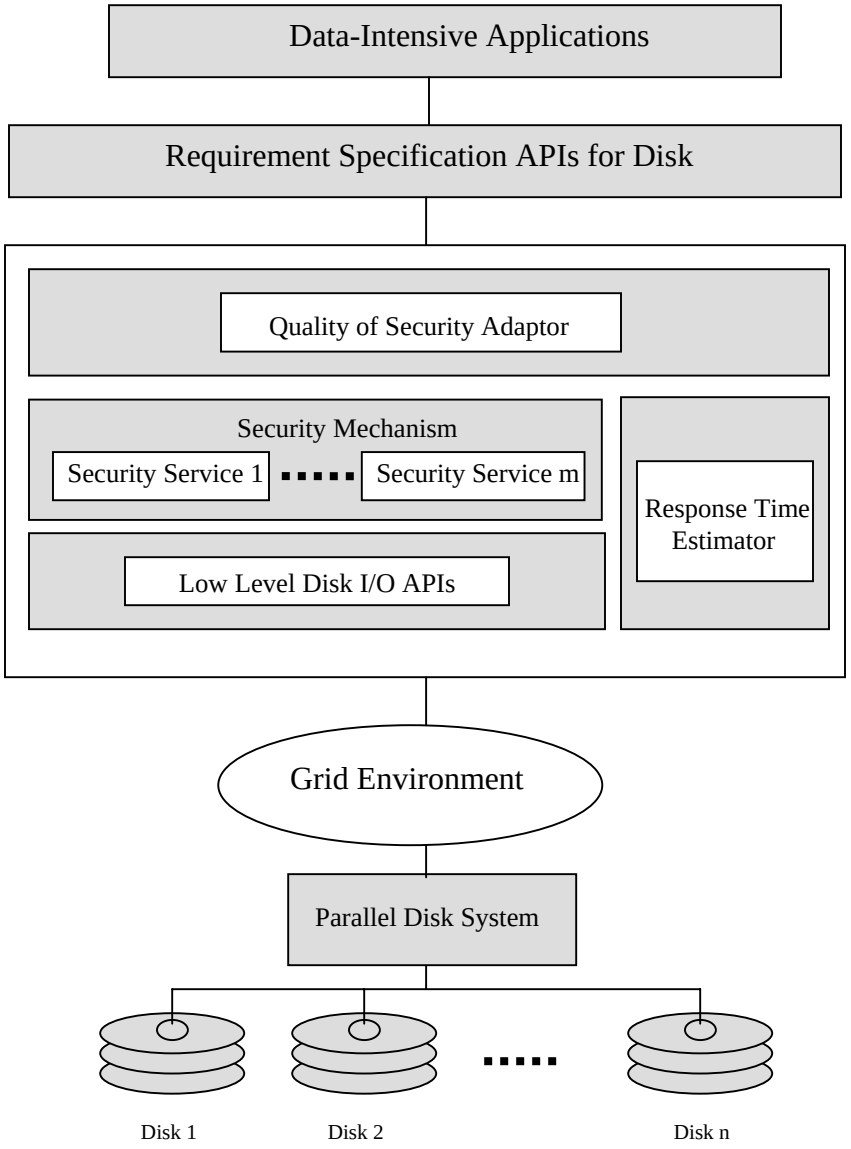

Fig. 2 Quality of Security Framework for Disk Systems in a Grid.

Application developers are enabled to make use of the requirement specification API to implement applications with high flexibility in response time and security service selection. As such, applications with flexible security and timing constraints can be efficiently implemented using the proposed framework.

The security mechanism in the framework is comprised of an array of security services with various security levels to meet applications' flexible security needs. Commonly used security services include but not limited to authentication service, auditing service, encryption service and access controller. Quality of security provided by each service largely depends on the robustness of the mechanism used to implement the service, on how long the service has been used, and on how rigorously the mechanism was tested.

The security mechanism is constructed in a flexible way that new security services can be readily incorporated and old security services can be dropped. The quality of security adaptor is responsible for making use of specified security and timing constraints to choose appropriate security services in the security mechanism for disk requests. The adaptor aims at selecting security services in a judicious way to guarantee both security and timely requirements of disk requests. The input of the adaptor is security and timing constraints of disk requests specified by the requirement specification APIS, whereas the output is an appropriate security service that can meet both the security and timely requirements.

The design of the quality of the security adaptor is a challenging task, because the adaptor has to optimize the quality of security for each disk request based on dynamically changing workload characteristics. The first step toward the development of the adaptor is an approach to predicting the response time of a disk request. The response time estimator, which is focused on estimate response times, plays a critical role in the framework. We analytically present the response time estimator in Section 6.

Framework-private services provide specific functions in addition to the underlying middleware services to meet framework's own needs. Low-level disk I/O APIs are employed to perform disk access operations, including read and write operations. The low-level disk access operations in most cases are implemented at operating system kernel level.

5. Modeling Quality of Security

5.1 Quality of security

Recall that the proposed framework encompasses an array of security services providing various quality of security. Each service is characterized by its quality of security measured by security level ranging from 0.1 to 1.0 [29]. Thus, the higher the security level of a service, the better the security quality of the service. Security services generally fall into three categories:

confidentiality, integrity, and availability. Without loss of generality, in our quality of security model we address the confidentiality issues by employing nine cryptographic algorithms in our framework [14]. Note that the quality of security model can be easily extended to incorporate the other two security service categories. We assume that the clients, network, and parallel disk system are always available by the virtue of fault-tolerant mechanisms residing in these components. This assumption is valid, because the overhead of supporting reliability in the system can be envisioned as a part of security overhead.

Security overheads incurred by the cryptographic algorithms depend on size of data to be encrypted and performance of the algorithms, each of which is assigned a security level as shown in table 1. For example, level 0.3 means that we use the Blowfish cryptographic algorithm. Note that the security levels provided for the Cryptographic Algorithms is assigned based on the assumption that people accept a slower security mechanism if it provides a higher security level compared with its faster peers.

Table 1. Cryptographic Algorithms used for Encryption Services

Cryptographic Algorithms	Security Level, σ	Performance KB/ms, $P(\sigma)$
SEAL	0.1	168.75
RC4	0.2	96.43
Blowfish	0.3	37.5
Knufu/Khafre	0.4	33.75
RC5	0.5	29.35
Rijndael	0.6	21.09
DES	0.7	15
IDEA	0.8	13.5

Let σ_i denote the security level of a cryptographic algorithm used to encrypt the data for disk request r_i, and d_i is the size of data to be encrypted. We can obtain the security overhead of request r_i using Eq. 1, where $P(\sigma_i)$ is a function mapping security level σ_i to the performance (measured by KB/ms) of the corresponding confidentiality service [14]. In other words, the security overhead is defined as the time spending in encoding the data over the performance of that encoding algorithm. For an instance, to calculate the security overhead for the Blowfish cryptographic algorithm we divide the data size to be encrypted over the Blowfish performance which is 37.5. Suppose that the data size to be encrypted is 10KB, thus the security overhead for

the Blowfish algorithm is 10KB divided by 37.5, which is equal to 0.27ms. It is worth noting that equation 1 is a simple yet efficient way to quantify security overhead.

$$T_{security}(\sigma_i, d_i) = \frac{d_i}{P(\sigma_i)}. \quad (1)$$

5.2 Security and timing constraints

We consider in this study data-intensive applications with both security and performance constraints, meaning that disk requests issued by the applications to a parallel disk system impose both security and performance requirements. While the security requirement of a request, e.g., r_i, is specified by a lower bound s_i on security level that the networked disk system has to provide. Hence, the security service controller must ensure that σ_i is greater than or equal to s_i. In this paper, we investigate desired response time, which is a specific performance requirement; and the desired response time of request r_i is represented by t_i. We denote parallelism degree of r_i by p_i. It is worth noting that parallelism degrees play a critical role in performance tuning of networked parallel disk systems. As such, it is appealing to devise the data partitioning mechanism to automatically determine a parallelism degree for each request in a way to improve throughput of the system. Given parallelism degrees of requests, quality of security for the requests can be tuned in a judicious manner by the security service controller.

The first step toward improving quality of security is to quantitatively measure security benefits of a disk request. To achieve this goal, we calculate the security benefit of request r_i using the following security level function.

$$S(r_i) = \sum_{j=1}^{p_i} \sigma_{ij}\ ,\ \sigma_{ij} \geq s_i \text{ and } p_i \leq m \quad (2)$$

where m is the number of disks in the parallel disk system and σ_{ij} is the security level of a confidentiality service chosen for the jth stripe unit of r_i.

Given a sequence of requests $R = \{r_1, r_2, \cdots, r_n\}$, we can obtain the security benefits experienced by the requests. Thus, we have

$$S(R) = \sum_{i=1}^{n} S(r_i), \quad (3)$$

Now we obtain the following non-linear optimization problem formulation to compute the optimal security benefit of the networked parallel disk system

$$\text{maximize } S(R) = \sum_{i=1}^{n} \sum_{j=1}^{p_i} \sigma_{ij} \ ,$$

$$\text{subject to (a) } \forall 1 \le i \le n : \max_{1 \le j \le p_i} \{\theta_{ij}\} \le t_i \ ,$$

$$\text{(b) } \sigma_{ij} \ge s_i \text{ and } p_i \le m \ , \tag{4}$$

where θ_{ij} is the response time for the *j*th stripe unit of request r_i. In an effort to enhance security of the system, we have to guarantee that the following three conditions are met. First, the response time of all stripe units in request r_i must be smaller than the desired response time. Second, the low bound on security level can not be violated. Third, the parallelism degree of r_i has to be smaller than or equal to the number of disks in the system. (Extend this paragraph)

When a disk request is issued by a client to a storage subsystem in the Data Grid, the proposed framework of the storage subsystem inserts the newly arrived request into a waiting queue based on the requests' earliest desired response times. The maximum response time of the stripe units of the disk requests is estimated using the response time estimator. The StReD algorithm keep increasing the security level of each stripe unit of the disk request until it can not meet the desired response time for the disk request.

6. Response Time Estimator

To adaptively adjust security levels for disk requests, we need to estimate each request's maximum response time, which is defined as the interval between the time a request is sent by a client and the time the parallel disk system completes corresponding disk I/O operations. Given a newly issued request *r*, the response time of *r* is estimated by Eq. (5).

$$T(r, p, \sigma) = T_{queue} + T_{partition} + \max_{i=1}^{p} \{T_{proc}^{i}(r, p, \sigma_i)\}, \tag{5}$$

where *p* is the parallelism degree determined by the data partitioning mechanism (see Eq. 5), $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_p)$ is the request's vector of security levels for *p* stripe units, T_{queue} is the queueing delay at the client side, $T_{partition}$ is the time spent in data partitioning, and T_{proc}^{i} is the system processing delay experienced by the *i*th stripe unit of the request. With respect to the *i*th stripe unit of the request, the system processing delay T_{proc}^{i} can be expressed as

$$T_{proc}^{i}(r, p, \sigma_i) = T_{security}^{i}(r, p, \sigma_i) + T_{network}^{i}(r, p, \sigma_i) + T_{disk}^{i}(r, p, \sigma_i) \ , \tag{6}$$

where $T^i_{security}$, $T^i_{network}$, and T^i_{disk} are the delays at the security mechanism, network subsystem, and parallel disk subsystems, respectively.

The delay at the security mechanism, which is also referred to as security overhead, depends on the assigned security level and data size of the stripe unit. Thus, we can easily derive $T^i_{security}(r, p, \sigma_i)$ from Eq. (1) as

$$T^i_{security}(r, p, \sigma_i) = T_{security}(\sigma_i, \frac{d}{p}) = \frac{d}{p \cdot P(\sigma_i)}, \quad (7)$$

where *d* is the data size of the request, and *d/p* is the data size of the *i*th stripe unit.

We assume that when the *i*th stripe unit of a request arrives at the network queue, there are *k* stripe units waiting to be delivered to the parallel disk sub-system. Suppose stripe units are transmitted in a first-in-first-out order, all the stripe units that are already in the queue prior to the arrival of the *i*th stripe unit must be transmitted earlier than the *i*th stripe unit. Hence, the delay in the network subsystem $T^i_{network}(r, p, \sigma_i)$ can be written as

$$T^i_{network}(r, p, \sigma_i) = \frac{i \cdot \frac{d}{p} + \sum_{j=1}^{k} d_j}{B_{network}}, \quad (8)$$

where d_j is the data size of the *j*th stripe unit in the network queue, and $B_{network}$ is the effective network bandwidth.

Similarly, it is assumed that when the *i*th stripe unit of the request arrives at disk *j*, there are *k* disk requests must be processed by disk *j* before handling the stripe unit. Thus, the delay in the disk subsystem $T^i_{disk}(r, p, \sigma_i)$ is given by the following formula

$$T^i_{disk}(r, p, \sigma_i) = T_{disk,j}(d/p) + \sum_{l=1}^{k} T_{disk,j}(d_l), \quad (9)$$

where $T_{disk,j}(d)$ is the disk processing time of a request containing *d* bytes of data. We can quantify $T_{disk,j}(d)$ as follows

$$T_{disk,j}(d) = T_{seek} + T_{rot} + \frac{d}{B_{disk}}, \quad (10)$$

where T_{seek} and T_{rot} are the seek time and rotational latency, and $\frac{d}{B_{disk}}$ is the data transfer time depending on the data size *d* and disk bandwidth B_{disk}.

7. Implementation and Results

The proposed framework takes full advantage of the quality of security adaptor to guarantee a diversity of security requirements while finishing disk requests before their desired response times in a Data Grid. To evaluate the effectiveness of the framework, in this section we compare a simulated Data Grid where our framework is integrated with another Data Grid without employing the framework. In our simulation experiments, we made use of the following three metrics to demonstrate the effectiveness of the proposed scheme.

(1) *Satisfied ratio* is a fraction of total arrived disk requests that are found to be finished before their desired response times.

(2) *Security level* is a sum of security level values of all disk requests issued to the parallel disk systems.

(3) *Overall performance* is performance metric measured by a product of the satisfied ratio and the security level.

7.1 Overall Performance Comparisons

This experiment aims to compare a simulated Data Grid with the StReD framework against the same Data Grid that makes no use of StReD. Note that storage subsystems used in our simulations consist of parallel disks. With different settings of parallelism degrees and data sizes, we study the impacts of varying arrival rates on system performance. To achieve this goal, we increased the arrival rate of I/O requests from 0.1 to 0.5 No./Sec. while setting the parallelism degree to 3 and data size to 100KB, 10MB, and 100 MB, respectively.

Fig. 3 plots the satisfied ratios, security levels, and overall performance of a storage subsystem with and without StReD. Figs 3(aa), (ba) and (ca) reveal that the StReD framework yields satisfied ratios that are very close to those of the storage subsystems making no use of StReD. This is mainly because StReD endeavors to guarantee timing constraints of disk requests while maximizing security of parallel disks in storage subsystems. Figs. 3(ab), 3(bb), and 3(cb) illustrate that StReD significantly improves the quality of security of the storage subsystems by an average of 126%. We can attribute the improvement in security to the fact that StReD strives to increase security level of each parallel disk request provided that the corresponding real-time requirement can be met. It is observed that as the value of arrival rate increases, the security levels of the both systems decrease. This result is not surprising because high arrival rates lead to

high workload, forcing the storage subsystems to merely meet the minimal security requirements of a vast majority of requests in order to process a large number of requests in a timely manner.

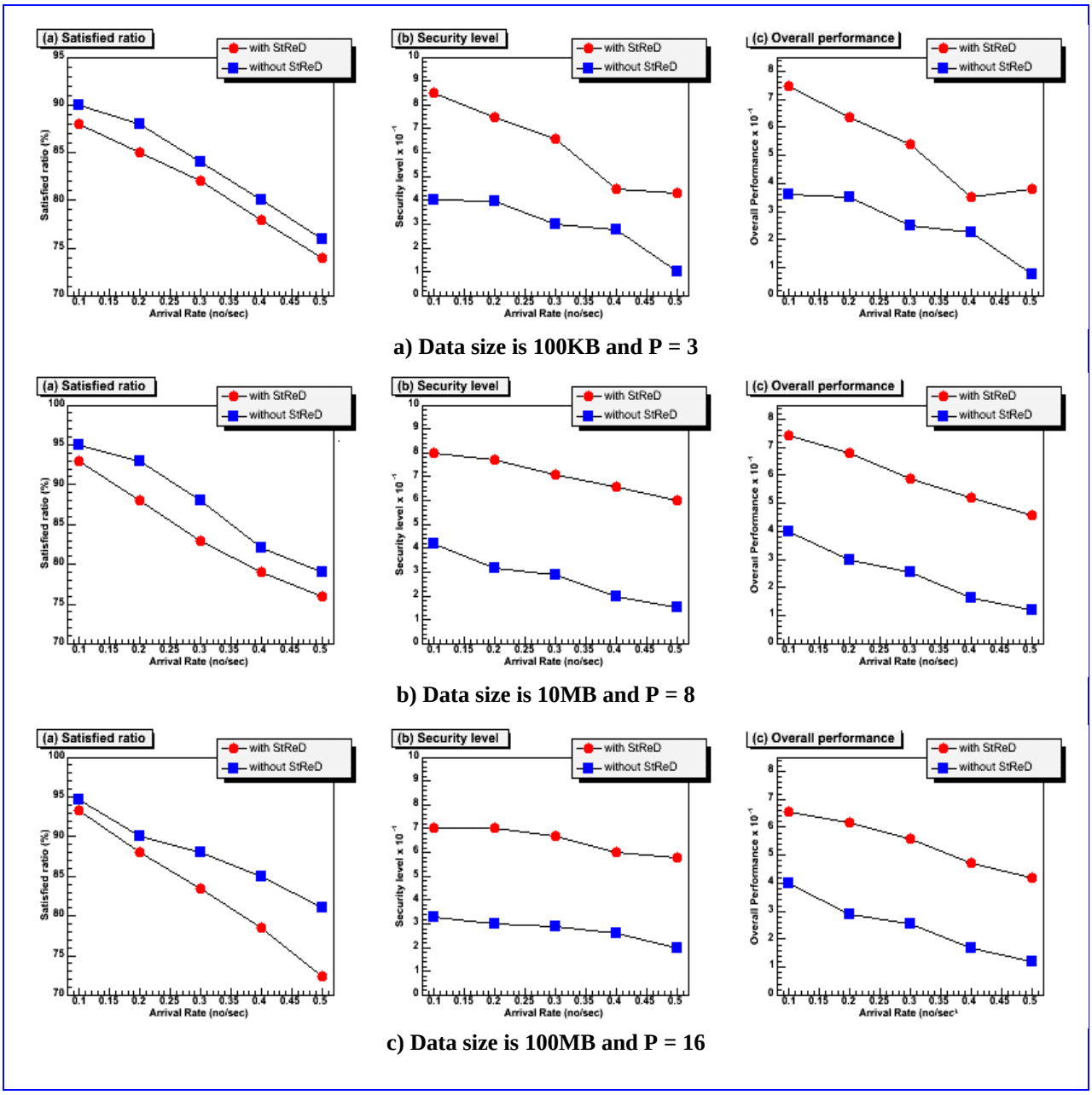

Fig.3. Impact of arrival rate on performance of storage subsystems in a Data Grid.

Interestingly, StReD always achieves higher security levels compared with the storage subsystems without StReD. It is worth noting that the security improvement comes at the cost of satisfied ratios (see Figs 3(aa), 3(ba) and 3(ca), because the satisfied ratios of StReD are slightly reduced due to relatively high security overhead caused by the StReD strategy. Figs 3(ac), 3(bc),

and 3(cc) reveal that StReD substantially boosts the overall performance. The reason of the expected overall performance improvement is two-fold. First, StReD adaptively enhances the security levels for disk I/O requests. Second, the performance gains in security level can eventually offset the extra security overhead.

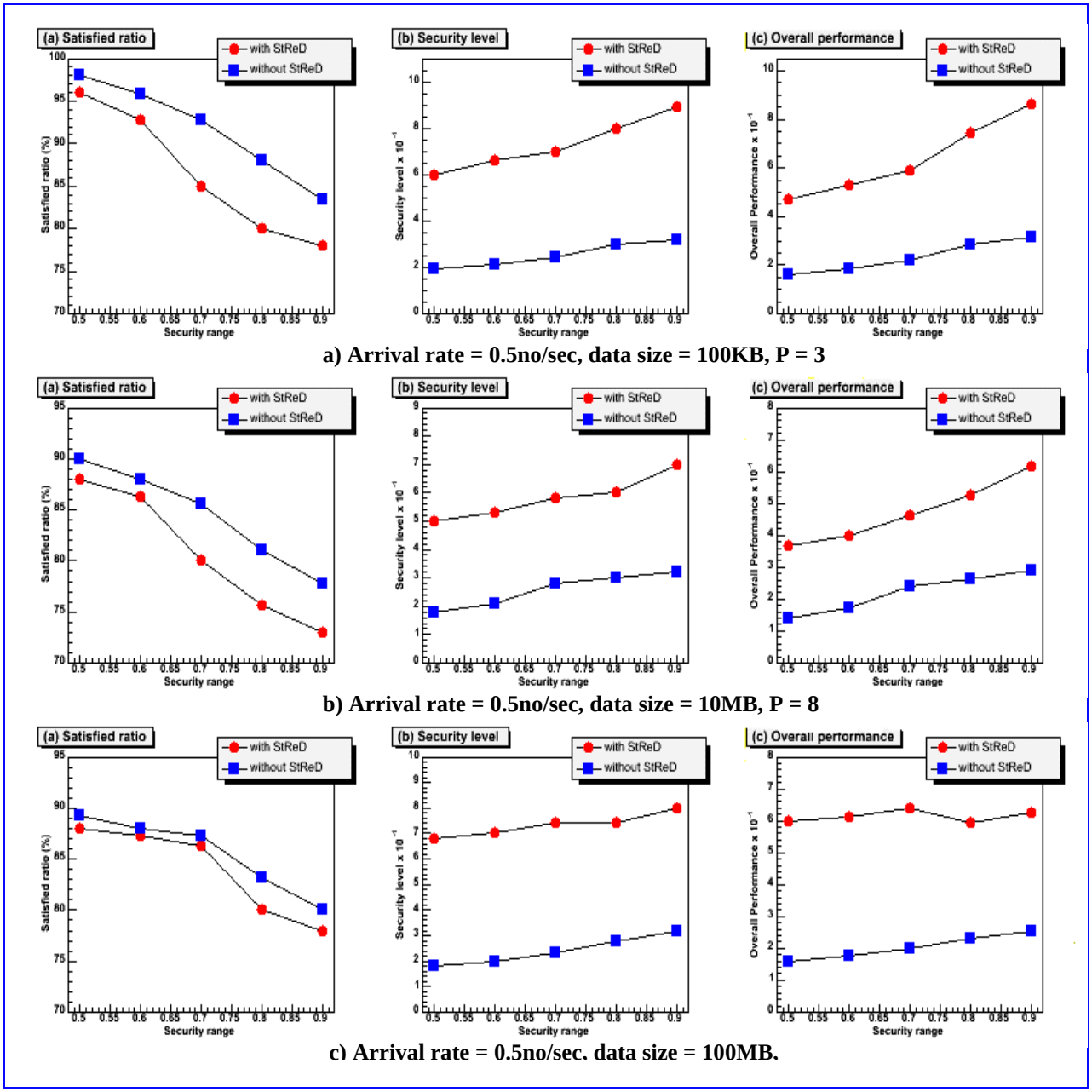

Fig. 4. Impact of Security Range on performance of storage subsystems in a Data Grid.

7.2 Impacts of security requirements

In this group of experiments, we investigate the performance impacts of security requirements imposed by storage subsystems with parallel disks. As mentioned earlier, the security

requirement of each disk request is characterized by a security range varying from s_{min} to s_{max}. We varied s_{max} from 0.5 to 0.9 while fixing s_{min} to 0.1.

We observed from Figs. 4(aa), 4(ba), and 4(ca) that increasing the maximal security level of the security range leads to the decreasing values of satisfied ratio of storage subsystems in two Data Grids. This is because when security level requirements of disk requests are high, the parallel disks in the storage subsystems need to fulfill the security requirements with high security overheads, which in turn reduce satisfied ratios. Again, the satisfied ratios of StReD are reasonably close to those of the alternative strategy. It is observed form Figs. 4(ab), 4(bb), and 4(cb) that when s_{max} goes up, the security levels of the two evaluated storage subsystems in the Data Grids gradually increase. This result implies that the security levels of the storage subsystems heavily rely on the security requirements of disk requests. Figs. 4(ac), 4(bc), and 4(cc) illustrate that the overall performance of the two Data Grids is improved with the increasing values of s_{max}, because the overall performance is more sensitive to security level than to satisfied ratio.

8. Summary and Future Work

In this paper, we presented a novel quality of security framework for storage resources in a Grid Environments. The quality of security adaptor, a centrepiece of the framework, is responsible for choosing the most appropriate security services for disk requests to guarantee both security and timely requirements. This adaptor paves the way to the design of applications with flexible security and timely requirements. The effectiveness of the proposed framework was evaluated by comparing the performance of a Grid with the framework against that of the same Grid without the framework. Experimental results based on disk traces show that the proposed framework is capable of significantly improving quality of security and guaranteeing desired response times of disk requests.

Future studies in this research can be performed in the following directions. First, we will implement the framework in a real world Grid. Second, we plan to integrate memory resources into our framework to further improve performance of data-intensive applications. Last but not least, we will extend the framework to deal with heterogeneous Data Grids, where different sites have different storage capabilities.

Acknowledgements

The work reported in this paper was supported in part by the New Mexico Institute of Mining and Technology under Grant 103295 and by Intel Corporation under Grant 2005-04-070.

References

[1] D. Avitzour, "Novel scene calibration procedure for video surveillance systems," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 40, No. 3, pp. 1105-1110, July 2004.

[2] W. Allcock, et al., J. Bresnahan, J. Bunn, Grid-enabled particle physics event analysis: experiences using a 10 gb, high-latency network for a high-energy physics application, Future Generation Computer Systems, 19(6), (2003), pp.983–997.

[3] A. Breckenridge, L. Pierson, S. Sanielevici, J. Welling, R. Keller, U. Woessner, J. Schulze, Distributed, on-demand, data-intensive and collaborative simulation, analysis, Future Generation Computer Systems, 19(6), (2003), pp.849–859.

[4] C. Chang, B. Moon, A. Acharya, C. Shock, A. Sussman, and J. Saltz. "Titan: a High-Performance Remote-Sensing Database," *Proc. the 13th Int'l Conf. Data Engineering*, Apr 1997.

[5] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke, "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets," *Journal of Network and Computer Applications*, 23(3): 187-200, 2000.

[6] Z. Dimitrijevic and R. Rangaswami, "Quality of Service Support for Real-time Storage Systems," *Proc. Int'l Conf. IPSI*, Sv. Stefan, Montenegro, October 2003.

[7] J. Huang, Y. Wang and F. Cao, "On Developing Distributed Middleware Services for QoS- and Criticality-Based Resource Negotiation and Adaptation," *Real-Time Systems* 16(2): 187-221; May 1999.

[8] J. Hughes and D. Corcorna, "A Universal Access, Smart-Card-Based, Secure File System," *Atlanta Linux Showcase*, Oct. 1999.

[9] C. Irvine and T. Levin, "Towards a taxonomy and costing method for security services," *Proc. 15th Annual Computer Security Applications Conference*, 1999.

[10] C. Irvine and T. Levin, "Quality of Security Service", *Proc. of New Security Paradigms Workshop2000*, Cork, Ireland, September, 2000.

[11] M. Kallahalla and P. J. Varman, "Improving parallel-disk buffer management using

randomized writeback," *Proc. Int'l Conf. Parallel Processing*, pp. 270-277, Aug. 1998.

[12] D. Kotz and C. Ellis, "Cashing and writeback policies in parallel file systems," *Proc. IEEE Symp. Parallel and Distributed Processing*, pp. 60-67, Dec. 1991.

[13] K. Nahrstedt, D. Xu, D. Wichadakul, and B. Li, "QoS-aware middleware for ubiquitous computing," *IEEE Communications Magazine*, 39(11):140--148, November 2001.

[14] M. Nijim, X. Qin, T. Xie, and M. Alghamdi, "Awards: An Adaptive Write Scheme for Secure Local Disk Systems," *Proc. 25th IEEE Int'l Performance Computing and Communications Conference,* Phoenix, AZ, April 2006.

[15] M. S. Prez, J. Carretero, F. Garca, J. M. Pea, V. Robles, Mapfs: A flexible multiagent parallel file system for clusters, Future Generation Computer Systems, 22(5), (2006), pp.620–632.

[16] X. Qin and H. Jiang, "Data Grids: Supporting Data-Intensive Applications in Wide Area Networks," *High Performance Computing: Paradigm and Infrastructure*, pp. 481-494, edited by Laurence T. Yang and Minyi Guo, John Wiley and Sons, spring, 2005.

[17] X. Qin, "Design and Analysis of a Load Balancing Strategy in Data Grids," Future Generation Computer Systems 23 (1) (2007) 132-137.

[18] S. Rajasekaran and X. Jin, "A practical realization of parallel disks Parallel Processing," *Proc. Int'l Workshop Parallel Processing*, pp. 337-344, Aug. 2000.

[19] S. Rajasekaran, "Selection algorithms for parallel disk systems," *Proc. Int'l Conf. High Performance Computing*, pp.343-350, Dec. 1998.

[20] E. Riedel, M. Kallahalla, and R. Swaminathan, "A Framework for Evaluating Storage System Security," *Proc. the 1st Conf. File and Storage Technologies*, Monterey, CA, Jan. 2002.

[21] A. Samar, H. Stockinger, "Grid Data Management Pilot (GDMP): A Tool for Wide Area Replication," *Proc. Of international Conference on Applied Informatics*, Feb. 2001.

[22] S. Song, Y.-K. Kwok, and K. Hwang, "Trusted Job Scheduling in Open Computational Grids: Security-Driven Heuristics and A Fast Genetic Algorithms, "*Proc. Int'l Symp. Parallel and Distributed Processing*, 2005.

[23] P. Scheuermann, G. Weikum, and P. Zabback, "Data portioning and load balancing in parallel disk systems," *VLDB Journal*, Vol.7, pp.48-66, 1998.

[24] T. Sumner and M. Marlino, "Digital libraries and educational practice: a case for new

models," *Proc. ACM/IEEE Conf. Digital Libraries*, pp. 170 – 178, June 2004.

[25] T. Tanaka, "Configurations of the Solar Wind Flow and Magnetic Field around the Planets with no Magnetic field: Calculation by a new MHD," *J. Geophysical Research*, pp.17251-17262, Oct. 1993.

[26] M. Tang, B.-S. Lee, X. Tang, C.-K. Yeo, "The impact of data replication on job scheduling performance in the data grid," Future Generation Computer Systems, 22(3) (2006) 254-268.

[27] B. Tierney, W. Johnston, J. Lee, M. Thompson, A data-intensive distributed computing architecture for grid applications, Future Generation Computer Systems, 16(5), (2000), pp. 473–481.

[28] T. Xie and X. Qin, "Scheduling Security-Critical Real-Time Applications on Clusters," *IEEE Transactions on Computers*, vol. 55, no. 7, pp. 864-879, July 2006.

[29] T. Xie and X. Qin, "Enhancing Security of Real-Time Applications on Grids through Dynamic Scheduling," *Proc. 11th Workshop Job Scheduling Strategies for Parallel Processing*, June 2005.

[30] T. Xie and X. Qin, "A New Allocation Scheme for Parallel Applications with Deadline and Security Constraints on Clusters," *Proc. IEEE Int'l Conf. Cluster Computing*, Boston, USA, Sept. 2005.

[31] T. Xie, X. Qin, and Andrew Sung, "SAREC: A Security-Aware Scheduling Strategy for Real-Time Applications on Clusters," *Proc. 34th Int'l Conf. Parallel Processing*, Norway, June 2005.

[32] T. Xie, X. Qin, and M. Nijim, "SHARP: A New Real-Time Scheduling Algorithm to Improve Security of Parallel Applications on Heterogeneous Clusters," *Proc. 25th IEEE Int'l Performance Computing and Communications Conf.*, Phoenix, AZ, April 2006.

[33] C. Wang, and W. A. Wulf, "Towards a Framework for Security Measurement," *Proc. National Information Systems Security Conference*, Baltimore, MD, pp. 522-533, October, 1997.