International Journal of Engineering Research and Modern Education (IJERME)
ISSN (Online): 2455 - 4200
(www.rdmodernresearch.com) Volume I, Issue I, 2016

ENHANCED MULTIPLE KEYWORD SEARCHING MECHANISM FOR SECURE DATA IN CLOUD

A. Manoj Prabakaran* & K. Ramamoorthy**
* PG Scholar, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu
** Assistant Professor, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

Abstract:
Storing data on untrusted storage makes secure data sharing a challenge issue. On one hand, data access policies should be enforced on these storage servers; on the other hand, confidentiality of sensitive data should be well protected against them. Cryptographic methods are usually applied to address this issue – only encrypted data are stored on storage servers while retaining secret key(s) to the data owner herself; user access is granted by issuing the corresponding data decryption keys. The main challenges for cryptographic methods include simultaneously achieving system scalability and fine-grained data access control, efficient key/user management, user accountability and etc. To address these challenge issues, this dissertation studies and enhanced public-key cryptography – attribute-based encryption (ABE) can be used and it applies for fine-grained data access control on un-trusted storage. The first part of this paper discusses the necessity of applying ABE to secure data sharing on untrusted storage and addresses several security issues for ABE. The second part presents our ABE-based secure data sharing solutions for two specific applications – Cloud Computing and Wireless Sensor Networks (WSNs). In Cloud computing cloud servers are usually operated by third-party providers, which are almost certain to be outside the trust domain of cloud users.

Index Terms: Cloud Computing, Public-Key Cryptography, Attribute-Based Encryption (ABE)

1. Introduction:

Cloud services are becoming more and more important for people's life. People are more or less requested to submit or post some personal private information to the Cloud by the Internet. As people rely more and more on the Internet and Cloud technology, security of their privacy takes more and more risks. On the one hand, when data is being processed, transformed and stored by the current computer system or network, systems or network must cache, copy or archive it. These copies are essential for systems and the network.

A pioneering study of Vanish supplies a new idea for sharing and protecting privacy. In the Vanish system, a secret key is divided and stored in a P2P system with distributed hash tables (DHTs). With joining and exiting of the P2P node, the system can maintain secret keys. Some special attacks to characteristics of P2P are challenges of Vanish. In considering these disadvantages, this paper presents a solution to implement a self-destructing data system, or SeDas, which is based on an active storage framework. The SeDas system defines two new modules, a self-destruct method object that is associated with each secret key part and survival time parameter for each secret key part. In this case, SeDas can meet the requirements of self-destructing data with controllable survival time while users can use this system as a general object storage system.

It is basically the collection of computers on the internet that companies are using to offer their services. One cloud service that is being offered is a revolutionary storage method for your data. From music files to pictures to sensitive documents, the cloud invisibly backs up your files and folders and alleviates the potentially endless and costly search for extra storage space. An alternative to buying an external hard drive or deleting old files to make room for new ones, cloud storage is convenient and cost-effective. It works by storing your files on a server out in the internet somewhere rather than on your local hard drive.

However, if you wish to store information virtually, you must consider the added riskthat your information may be accessible to other—potentially people who you do not wish to have access. Below, we outline a few security risks to take into account and how to protect yourself and your data.Cloud computing is a relatively new tool for the average consumer. It is important to explore the service that most fits your needs.

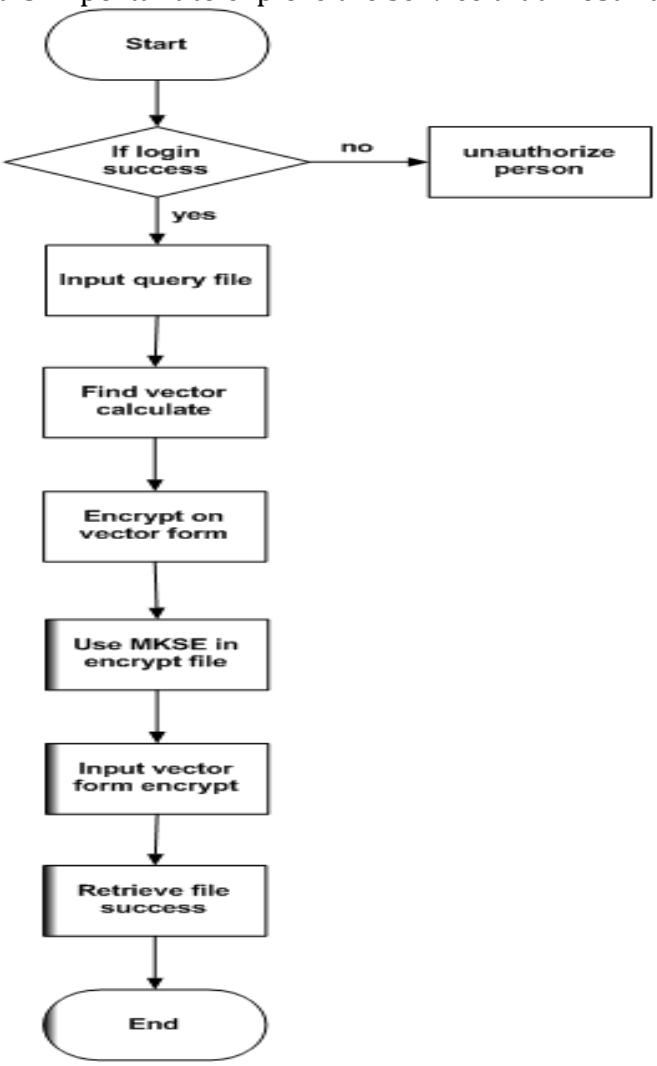


Figure 1: System Flow Diagram

The reminder of this paper is organized as follows. Section II, describes the Related Works. Section III, describes the Proposed Work. Section IV, describes the Experimental Evaluation and Results. Section V summarizes the Conclusion and Future Enhancement.

2. Related Works:

Vanish: Increasing data privacy with self-destructing data implemented by R. Geambasu, T. Kohno, A. Levy, and H. M. Levy. The goal of creating data that self-

destructs or vanishes automatically after it is no longer useful. Moreover, it should do so without any explicit action by the users or any party storing or archiving that data, in such a way that all copies of the data vanish simultaneously from all storage sites, online or offline. Numerous applications could benefit from such self-destructing data. As one example, consider the case of email. Such emails may cease to have value to the sender and receiver after a short period of time. Nevertheless, many of these emails are private, and the act of storing them in definitely at intermediate locations creates a potential privacy risk.

Defeating vanish with low-cost sybil attacks against large DHEs was developed by S. Wolchok, O. S. Hofmann, N. Heninger, C. J. Rossbach, B. Waters, and E. Witchel. As storage capacities increase and applications move into the cloud, controlling the lifetime of sensitive data is becoming increasingly difficult. Even if users cleanse their local files, copies may be retained long into the future by email providers, backup systems, and other services, and these may be targets of theft or subpoena.Vanish encapsulates data objects so that they "self-destruct" after a specified time, becoming permanently unreadable. It encrypts the data using a randomly generated key and then uses Shamir secret sharing to break the key into n shares where k of them is needed to reconstruct the key.

Safevanish: An improved data self-destruction for protecting data privacy was surveyed by L. Zeng, Z. Shi, S. Xu, and D. Feng. In cloud, self-destructing data mainly aims at protecting the data privacy. All the data and its copies will become destructed or unreadable after a user-specified period, without any user intervention. Besides, anyone cannot get the decryption key after timeout, neither the sender nor the receiver. The Washington's Vanish system is a system for self-destructing data under cloud computing, and it is vulnerable to "hopping attack" and "sniffer attack". It provide a new scheme, called Safe Vanish, to prevent hopping attacks by way of extending the length range of the key shares to increase the attack cost substantially, and do some improvement Original Vanish system.

Active storage framework for object-based storage device has surveyed by L. Qin and D. Feng. Active storage provides a service migration mechanism for applications to exploit processing capabilities in storage devices. However, in recent research, the model of service execution still remains passive request-driven mode. In self-management situation, a mechanism for automatic service execution is necessary. To address this problem, an active storage framework for object-based storage device is presented, which provides a hybrid approach to combine request-driven model and policy-driven model.

Active storage using object-based devices was surveyed by T. M. John, A. T. Ramani, and J. A. Chandy. This eliminates server as a bottleneck for data transfers between disks and clients and promises significant improved scalability through higherlevel interfaces. At the same time, as systems get faster and cheaper, people compute on larger and larger data sets. A large server system today will easily have a hundred disk drives attached to it. This large number of drives is necessary either to provide sufficient capacity or sufficient aggregate throughput for the target application. Taking this trend and extrapolating to future drive capabilities gives a promising picture for ondrive processing. A pessimistic value for the on-drive processing already in today's commodity SCSI disk controllers is 400 MHz, with perhaps 100 MB/s of sustained bandwidth in sequential access. This means that a system with one hundred disks has 40 GHz of aggregate processing power and 10 GB/s of aggregate bandwidth at the disks.

Typical multiprocessor systems cannot achieve these aggregate values in a single node, but computing clusters can be constructed to achieve the same capability. The advantage of the active disk cluster over a compute node cluster is the ability of the disk to manage the data directly on disk in terms of layout and scheduling. Data processing at the disk can also reduce the amount of data transferred, thereby reducing communication costs In addition, as storage is connected to a large collection of hosts by taking advantage of network-attachment and storage area networks, the interconnection network will rapidly become a principal bottleneck in large-scale applications.

An active storage system for high performance computing was surveyed by Y. Zhang and D. Feng. Traditionally, active storage devices execute custom application code on large amounts of data by utilizing the unused processing power of the storage nodes. For computation-intensive applications, the performance might be quite low due to insufficient processing power on the storage devices. In this paper we present a reconfigurable computing solution that can provide flexible, high-performance processing capabilities for the storage nodes.

3. Proposed Work:

The concepts of similarity relevance and scheme robustness to formulate the privacy issue in searchable encryption schemes, and then solve the insecurity problem by proposing a Multi Keyword searchable encryption (MKSE) scheme.

Novel technologies in the cryptography community and information retrieval community are employed, including homomorphic encryption and vector space model. In the proposed scheme, the majority of computing work is done on the cloud while the user takes part in ranking, which guarantees top k multi-keyword retrieval over encrypted cloud data with high security and practical efficiency.

A. User Creation for SSE: The user creates their own login credential when they want entering in cloud. The keys are automatically generated when the user register in cloud. Considering the large number of data users and documents in the cloud, it is necessary to allow multi-keyword in the search query and return documents in the order of their relevancy with the queried keywords. Scoring is a natural way to weight the relevance. Based on the relevance score, files can then be ranked in either ascending or descending. Several models have been proposed to score and rank files in information retrieval (IR) community.

B. Vector Calculation: Although all data files, indices and requests are in encrypted form before being outsourced onto cloud, the cloud server can still obtain additional information through statistical analysis. Denote the possible information leakage with statistic leakage. There are two possible statistic leakages, including term distribution and inter distribution. The term distribution of term t is t's frequency distribution of scores on each file .The inter distribution of file f is file f's frequency distribution of scores of each term. Term distribution and inter distribution are specific. They can be deduced either directly from cipher text or indirectly via statistical analysis over access and search pattern. The vector calculation is performed to the term frequency user querying file. Here access pattern refers to which keywords and the corresponding files have been retrieved during each search request, and search pattern refers to whether the keywords retrieved between two requests are the same.

C. MKSE Design: Server-side ranking based on order-preserving encryption violates the privacy of sensitive information, which is considered un-comprisable in the security-oriented third-party cloud computing scenario, i.e., security cannot be tradeoff for efficiency. To achieve data privacy, ranking has to be left to the user side. Traditional

user-side schemes, however, load heavy computational burden and high communication overhead on the user side, due to the interaction between the server and the user including searchable index return and ranking score calculation.

The user side ranking schemes are challenged by practical use. A more server-siding scheme might be a better solution to privacy issues. a new searchable encryption scheme, in which novel technologies in cryptography community and IR community are employed, including homomorphism encryption and vector space model. In the proposed scheme, the data owner encrypts the searchable index with homomorphism encryption.

When the cloud server receives query consisting of multi-keyword, it computes the scores from the encrypted index stored on cloud, and then returns the encrypted scores of files to the data user. Next, the data user decrypts the scores and picks out the top-k highest-scoring files' identifiers to request to the cloud server. The retrieval takes a Multi-Keyword communication between the cloud server and the data user. Thus name the scheme as Multi Keyword Searchable Encryption MKSE scheme, in which ranking is done at the user side while scoring calculation is done at the server side.

D. User Query for SSE: To alleviate the computational burden on user side, computing work should be at the server side, so need an encryption scheme to guarantee the operability and security at the same time on server side. MKSE allows specific types of computations to be carried out on the corresponding cipher text.

The result is the cipher text of the result of the same operations performed on the plaintext. That is, MKSE allows computation of cipher text without knowing anything about the plaintext to get the correct encrypted result. Although it has such a fine property, original fully homomorphic encryption scheme, which employs ideal lattices over a polynomial ring, is too complicated and inefficient for practical utilization.

4. Experimental Analysis and Results:

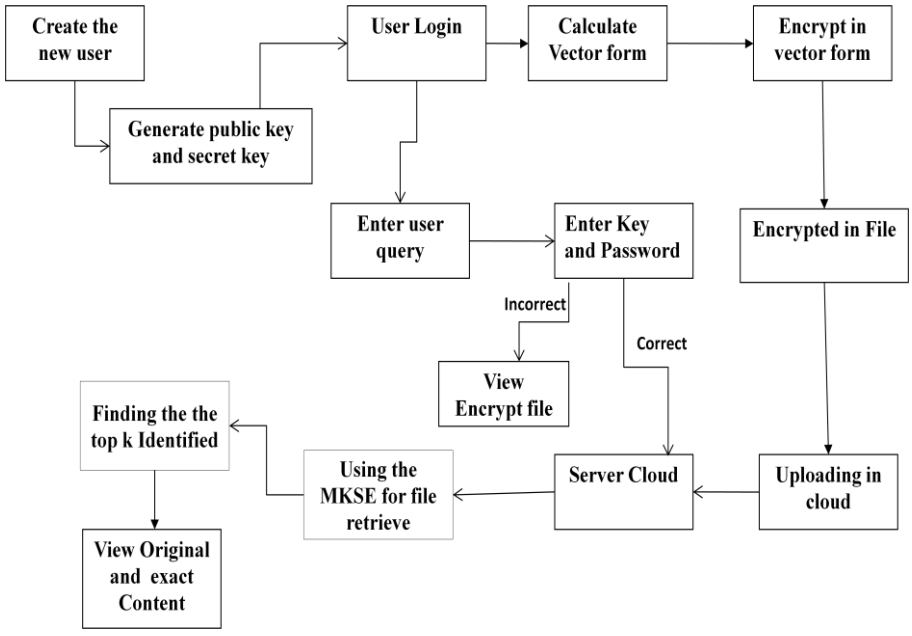


Figure 2: Architecture of Secure cloud data

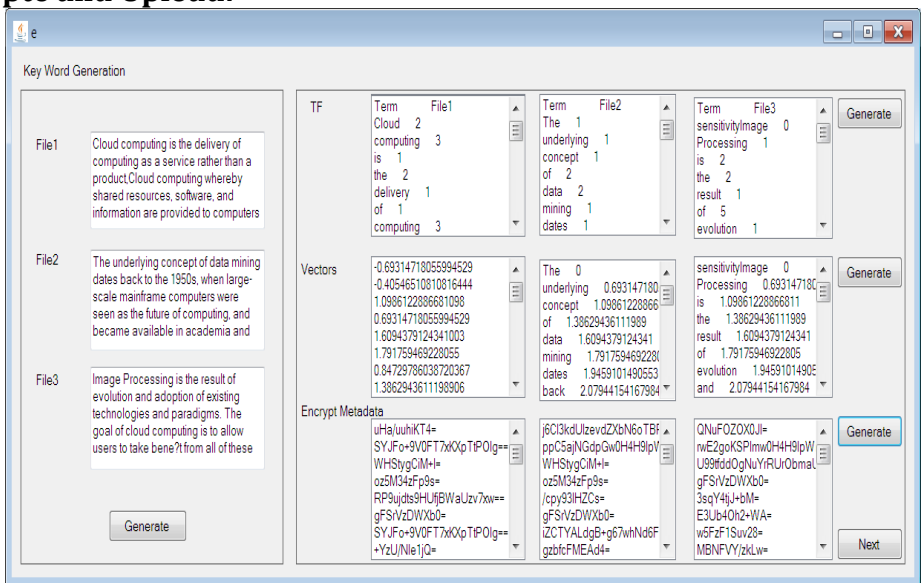
Implementation is the process of translating design specification in to source code. The primary goal of implementation is to write source code and internal

implementation. So that conformance of code to its specification can be easily verified, So that debugging, testing and modification are eased. The source is developed with clarity, simplicity and elegance.

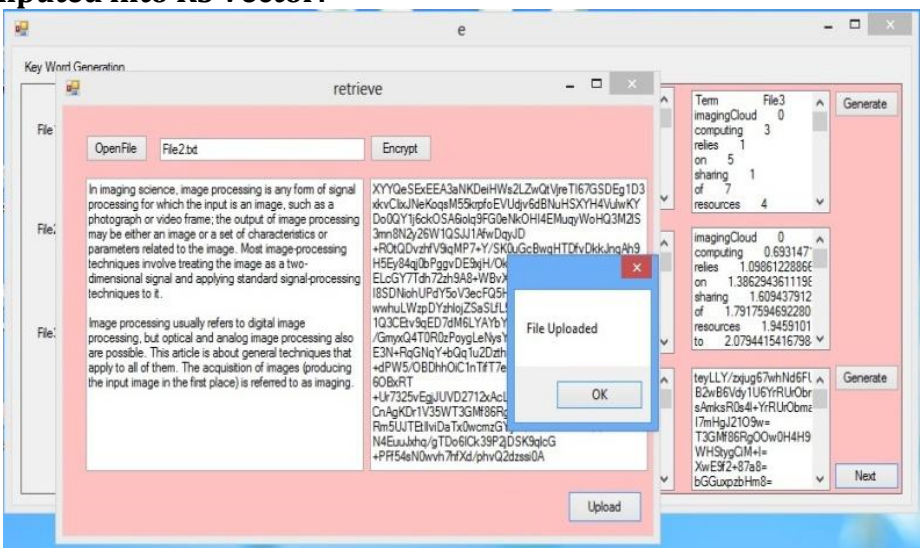
The coding is done in a modular fashion giving such importance even to the minute detail so, when hardware and storage procedures are changed or now data is added, rewriting of application programs is not necessary. To adapt or perfect use must determine new requirements, redesign generate code and test exiting software/hardware. Traditionally such task when they are applied to an existing program has been called maintenance.

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier. The active user must be aware of the benefits of using the system. Their confidence in the software built up. Proper guidance is impaired to the user so that he is comfortable in using the application. Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

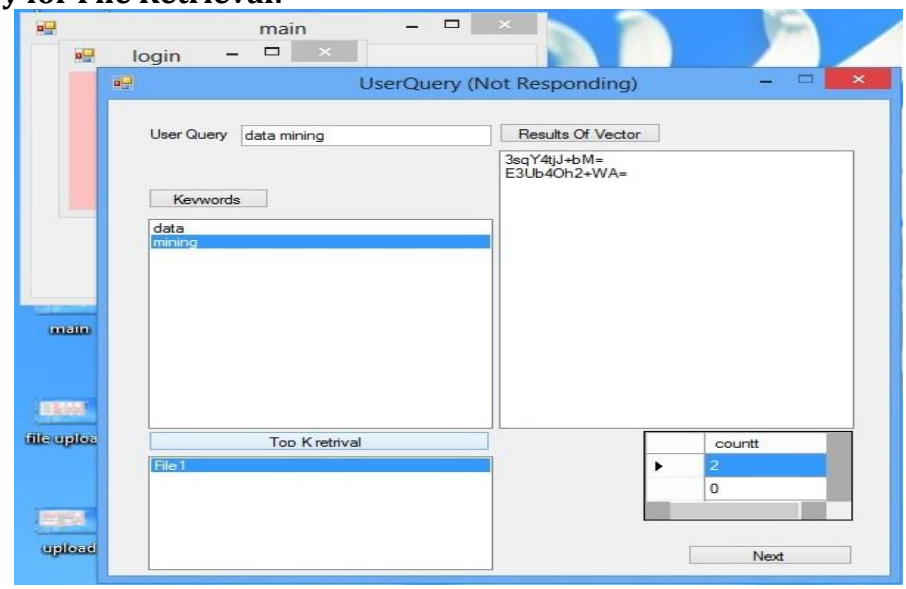
File Encrypte and Upload:



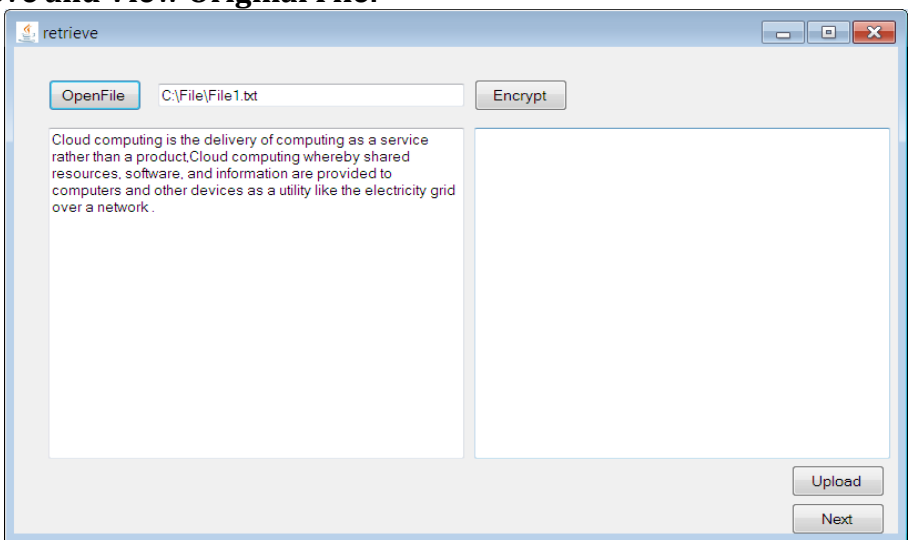
File is Computed into RS Vector:



User Query for File Retrieval:



File Retrieve and View Original File:



5. Conclusion and Future Enhancement:

In this paper, we motivate and solve the problem of secure multi-keyword top- k retrieval over encrypted cloud data. We define similarity relevance and scheme robustness. Based on OPE invisibly leaking sensitive information, we devise a server-side ranking SSE scheme. We then propose a MKSE scheme employing the fully homomorphic encryption, which fulfills the security requirements of multi-keyword top- k retrieval over the encrypted cloud data. By security analysis, we show that the proposed scheme guarantees data privacy. According to the efficiency evaluation of the proposed scheme over a real data set, extensive experimental results demonstrate that our scheme ensures practical efficiency.

In Cloud the Data is searched and received efficiently using MKSE but the data received is can't be proved whether it is correct are modified, and the data accessed by the user is also be proves as authorized not authenticated.

6. References:

1. R. Geambasu, T. Kohno, A. Levy, and H. M. Levy, "Vanish: Increasing data privacy with self-destructing data," in Proc. USENIX Security Symp., Montreal, Canada, Aug. 2009, pp. 299–315.

2. Shamir, "How to share a secret," Communication. ACM, vol. 22, no. 11, pp. 612–613, 1979.
3. S. Wolchok, O. S. Hofmann, N. Heninger, E. W. Felten, J. A. Halderman, C. J. Rossbach, B. Waters, and E. Witchel, "Defeating vanish with low-cost sybil attacks against large DHEs," in Proc. Network and Distributed System Security Symp., 2010.
4. L. Zeng, Z. Shi, S. Xu, and D. Feng, "Safevanish: An improved data self-destruction for protecting data privacy," in Proc. Second Int. Conf. Cloud Computing Technology and Science (CloudCom), Indianapolis, IN, USA, Dec. 2010, pp. 521–528.
5. L. Qin and D. Feng, "Active storage framework for object-based storage device," in Proc. IEEE 20th Int. Conf. Advanced Information Networking and Applications (AINA), 2006.
6. Y. Zhang and D. Feng, "An active storage system for high performance computing," in Proc. 22nd Int. Conf. Advanced Information Networking and Applications (AINA), 2008, pp. 644–651.
7. T. M. John, A. T. Ramani, and J. A. Chandy, "Active storage using object-based devices," in Proc. IEEE Int. Conf. Cluster Computing, 2008, pp. 472–478.
8. Devulapalli, I. T. Murugandi, D. Xu, and P. Wyckoff, 2009, Design of an intelligent object-based storage device [Online].
9. S. W. Son, S. Lang, P. Carns, R. Ross, R. Thakur, B. Ozisikyilmaz, W.-K. Liao, and A. Choudhary, "Enabling active storage on parallel I/O software stacks," in Proc. IEEE 26th Symp. Mass Storage Systems and Technologies (MSST), 2010.
10. Y. Xie, K.-K. Muniswamy-Reddy, D. Feng, D. D. E. Long, Y. Kang, Z. Niu, and Z. Tan, "Design and evaluation of oasis:An active storage framework based on t10 osd standard," in Proc. 27th IEEE Symp.Massive Storage Systems and Technologies (MSST), 2011.