ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

De-duplication and Encryption in Cloud Storage

Joshi Vinay Kumar[1], V Ravi Shankar[2]

[1]PG Student, GITAM University, Hyderabad, India

[2]Assistant Professor, GITAM University, Hyderabad, India

ABSTRACT: De-duplication is one of the latest trend technologies in the current market because of its ability to reduce costs. In terms of cloud storage system a most concern issue is providing a definite encryption and security to redundant or duplicated data in cloud.**.** De-duplication can be applied to data on primary storage, backup storage, cloud storage, LAN and WAN transfers.Organizations frequently use de-duplication in backup and calamity recovery applications as well to avoid duplicate of similar data for savage of space in cloud.In this paperwe discuss about de-duplication technology, proposed security model, private cloud.

KEYWORDS: de-duplication technology, privatecloud,proposed security model.

I.INTRODUCTION

De-duplication is ideal for highly redundant operations like backup, which requires repeatedly copying and storing the same data set multiple times for recovery purposes over 30- to 90-day periods. As a result, enterprises of all sizes rely on backup and recovery with de-duplication for fast, reliable, and cost-effective backup and recovery. De-duplication segments an incoming data stream, uniquely identifies data segments, and then compares the segments to previously stored data. If the segment is unique, it's stored on disk. However, if an incoming data segment is a duplicate of what has already been stored, a reference is created to it and the segment isn't stored again.

For example, a file or volume that's backed up every week creates a significant amount of duplicate data. De-duplication algorithms analyze the data and store only the compressed, unique segments of a file. This process can provide an average of 10 to 30 times reduction in storage capacity requirements, with average backup retention policies on normal enterprise data. This means that companies can store 10 TB to 30 TB of backup data on 1 TB of physical disk capacity, which has huge economic benefits as follows.

- Eliminating redundant data can significantly shrink storage requirements and improve bandwidth efficiency. Because primary storage has gotten cheaper over time, enterprises typically store many versions of the same information so that new workers can reuse previously done work. Some operations like backup store extremely redundant information.
- De-duplication lowers storage costs as fewer disks are needed. It also improves disaster recovery since there's far less data to transfer. Backup and archive data usually includes a lot of duplicate data.
- The same data is stored over and over again, consuming unnecessary storage space on disk or tape, electricity to power and cool the disk or tape drives, and bandwidth for replication. This creates a chain of cost and resource inefficiencies within the organization.

1.2 De-duplication Technology

Data de-duplication[4][5] is a highly proprietary technology. De-duplication methods vary widely from vendor to vendor, and many of those methods are patented. For example, Microsoft has a patent on single instance storage. In

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

addition, Quantum owns a patent on variablelength de-duplication. Many other vendors also own patents related to de-duplication technology.

2. **De-duplication Implementation**

The process for implementing data de-duplication technology varies widely depending on the type of product and the vendor. For example, if de-duplication technology is included in a backup appliance or storage solution, the implementation process will be much different than for standalone de-duplication software. In general, de-duplication technology can be deployed in one of two basic ways: at the source or at the target. In source de-duplication, data copies are eliminated in primary storage before the data is sent to the backup system. The advantage of source de-duplication is that is reduces the bandwidth requirements and time necessary for backing up data. On the downside, source de-duplication consumes more processor resources, and it can be difficult to integrate with existing systems and applications. By contrast, target de-duplication takes place within the backup system and is often much easier to deploy. Target de-duplication comes in two types: in-line or post-process. In-line de-duplication takes place before the backup copy is written to disk or tape. The benefit of in-line de-duplication is that it requires less storage space than post-process de-duplication, but it can slow down the backup process. Post-process de-duplication takes place after the backup has been written, so it requires that organizations have a great deal of storage space available for the original backup. However, post-process de-duplication is usually faster than in-line de-duplication.

II.RELATED WORK

In this section we discuss about the present encryption algorithm used in cloud storage and its drawbacks.

- **Convergent encryption:**Convergent encryption is used to encrypt and decrypt file. User can derive the convergent key from each original data copy, then using that key encrypt data file. Also user derives tag for data copy to check duplicate data. Iftag is same then both files are same. Both convergent key and tag are independently derives. Convergent encryption [2],[3] also known as content hash keying, is used to produces identical cipher text from identical plaintext files. The simplest implementation of convergent encryption can be defined as: Alice derives the encryption key from her file F such that K = H(F), where H is a cryptographic hash function. Convergent encryption scheme can be defined with four primitive functions:
- KeyGenCE(M) -> K is the key generation algorithm that maps a data copy M to a convergent key K;
- EncCE(K, M) ->C is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs and then outputs a ciphertext C;
- DecCE (K, C) ->M is the decryption algorithm that takes both the ciphertext C and the convergent key K as inputs and then outputs the original data copy M ; and
- TagGen(M) -> T (M) is the tag generation algorithm that maps the original data copy M and outputs a tag T (M).
- **Proof of ownership:** proof of ownership (PoW)[1] is a protocol enables users to prove their ownership of data copies to the storage server.PoW is implemented as an interactive algorithm run by user and storage server act as prove and verifier. The verifier derives a short value $\phi(M)$ from a data copy M . To prove the ownership of the data copy M , the prover needs to send ϕ to the verifier such that $\phi = \phi(M)$.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

2.1 PROBLEMS

- **Confirmation attack:** A more fundamental problem with convergent encryption is the confirmation attack. Here an attacker can check if a given key H is in the associative array. If the attacker can do this, he can also check if a given plaintext X is in the associative array by checking the presence of

$$H=HB\ (E\ (HA(X),X)))$$

If no preventative measures are taken, this could allow an attacker to confirm if the user is in possession of a certain file, for example a banned book or a pirated movie.

- **Offline brute-force attack[9]:** In convergent encryption it is easy to recognize the correct key. The correct key K will satisfy the equation

$$K=HA\ (D\ (K,X'))$$

While theoretically interesting, offline brute-force attacks on conventional symmetric cyphers are already possible in practice. Plaintexts often contain easily recognizable structures such as file headers. This can then be used as an effective heuristic to check if the correct key is found. Such an attack will work on any cipher where keys are significantly shorter than the messages, which in practice means anything but the one-time pad.

- **Learn the remaining attack:**Perhaps the most important of the possible attacks is the learn the remaining attack. Suppose an attacker knows most of the file, for example if the file is a PDF form where the user needs to fill in sensitive information, says a PIN code. The attacker can now create all possible versions of the file X and check if it matches a cipher text X′ by encrypting and comparing or using the identity

$$X=D\ (HA(X),X'))$$

In fact, the attacker does not even need to have the value X′, it is sufficient to check if a given key H is in the associative array using the equation given above.This attack is possible when X is known to be a member of a small set. The set of possible X's can then be exhaustively tried at the small cost of three hashing and one encryption operation per try. Or in information theoretical terms: when the relative entropy of the plaintext relative to the attacker is low.

PRILIMINARIES

In this section we just give anflow chart representation of private cloud involving which is used for providing and performing secure storage encryption access using a token in cloud, and Proposed security model.

III.PROPOSED SECURITY MODEL

- We have a number of users who, before uploading to the Cloud, split data into blocks (the data chunking technique that best fits your data), encrypt blocks with convergent encryption and send to the server (or gateway) encrypted blocks together with their associated encrypted keys.
- A server/gateway that further encrypts blocks and keys with a set of unique and secret keys.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

- A metadata manager that updates the metadata (in order to rebuild the structure of each file) , stores encrypted block keys and performs de-duplication on encrypted blocks. Only those blocks that are not already stored are actually stored.
- A storage layer to store single blocks, which can be seen as files/objects of small size. Since our system is completely storage agnostic, we can implement the storage layer with any storage system/provider. For instance, we might use a cloud storage provider such as Amazon S3, a distributed storage, a local file system, etc.

PROPOSED SYSTEM ARCHITECTURE AND PRIVATE CLOUD

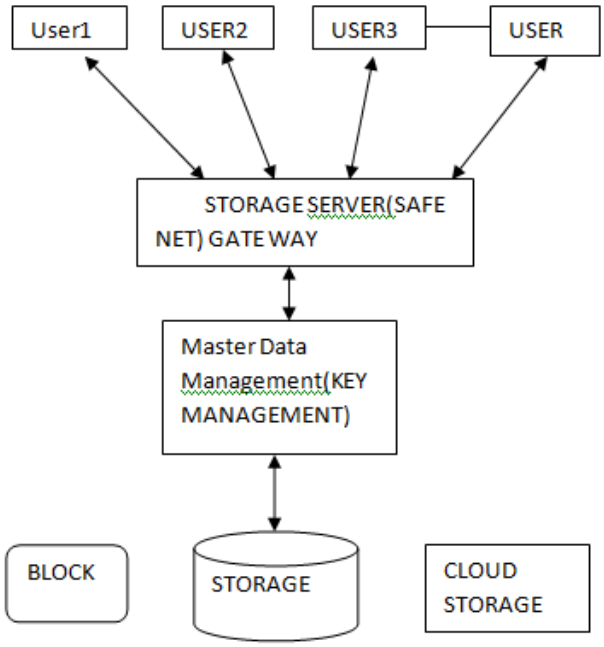

FIG 1

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

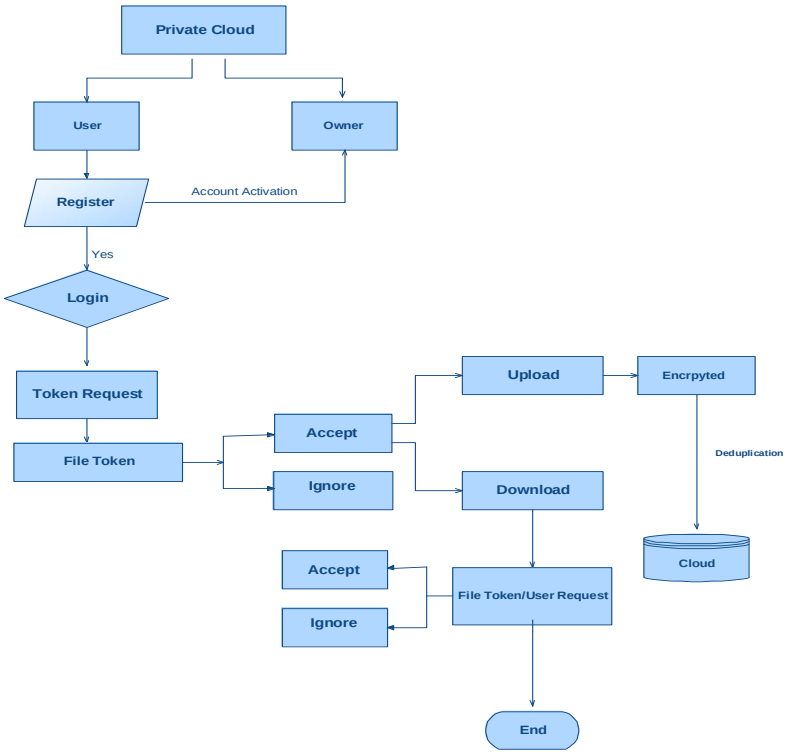

FIG 2. Flowchart

It's important to point out that thanks to our design**,**no single component has enough information to decrypt blocks or keys**.** Indeed**,** blocks and keys are encrypted by users and the server/gateway**.**While this solution might seem straightforward, it's surprising to see how effective it's and how well it fits for various use cases.
A typical use case might be the following: the employees of a big enterprise want to store their data in the Cloud while keeping data confidentiality and minimizing the amount of storage space consumed. In this context, the server/gateway can be deployed on the premises of the enterprise and the metadata manager can be deployed either locally or remotely (if we want to rely on a service provider for the de-duplication and key management services).

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

PERFORMANCE

According to the above Proposed system the following performance has been incurred shown in the following chart diagram.

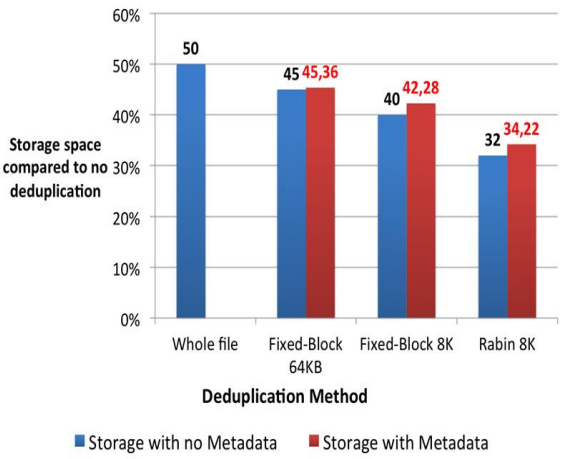

FIG 3

IV.IMPLEMENTATION AND RESULT

Finally, we are currently working (halfway done) on the full implementation of this system (metadata manager is based on REDIS) and the results are very promising. The storage space required for our metadata is really minimal and doesn't impact the gains of de-duplication. Also, from a computational point of view, de-duplication is very efficient (constant cost)! Following are the some of the screens obtained in our research.

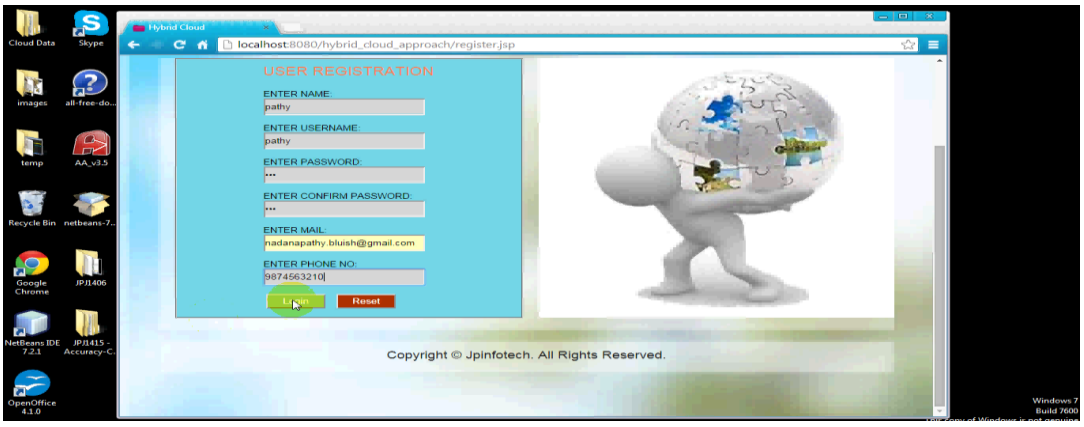

FIG4 USER REGISTRATION

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

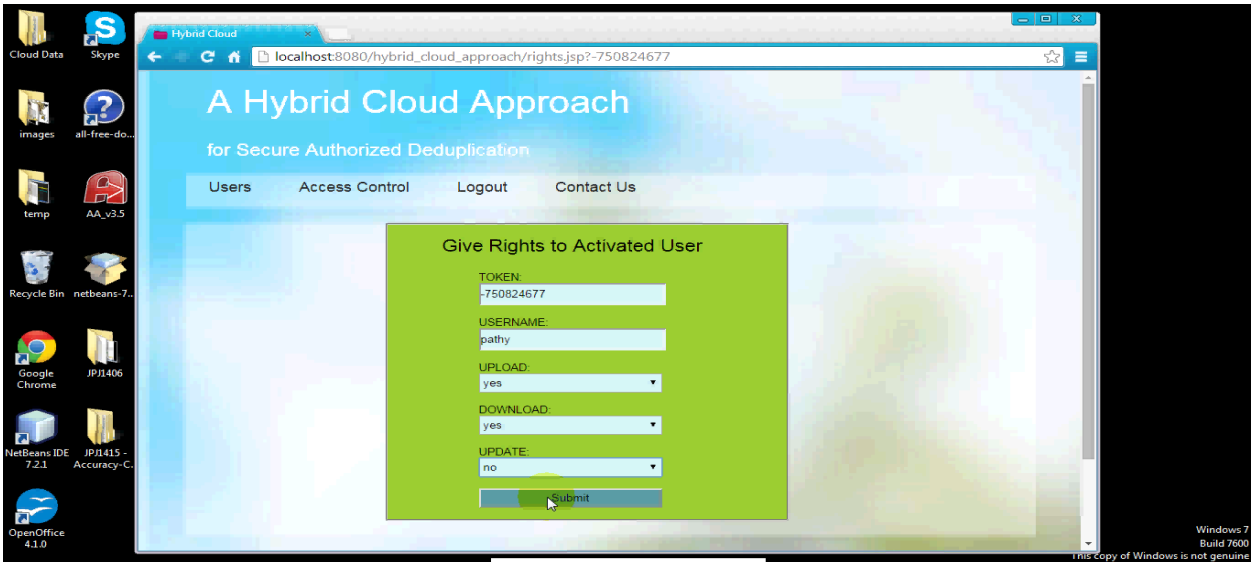

FIG 5 PRIVATE CLOUD TOKEN PRIVILAGES

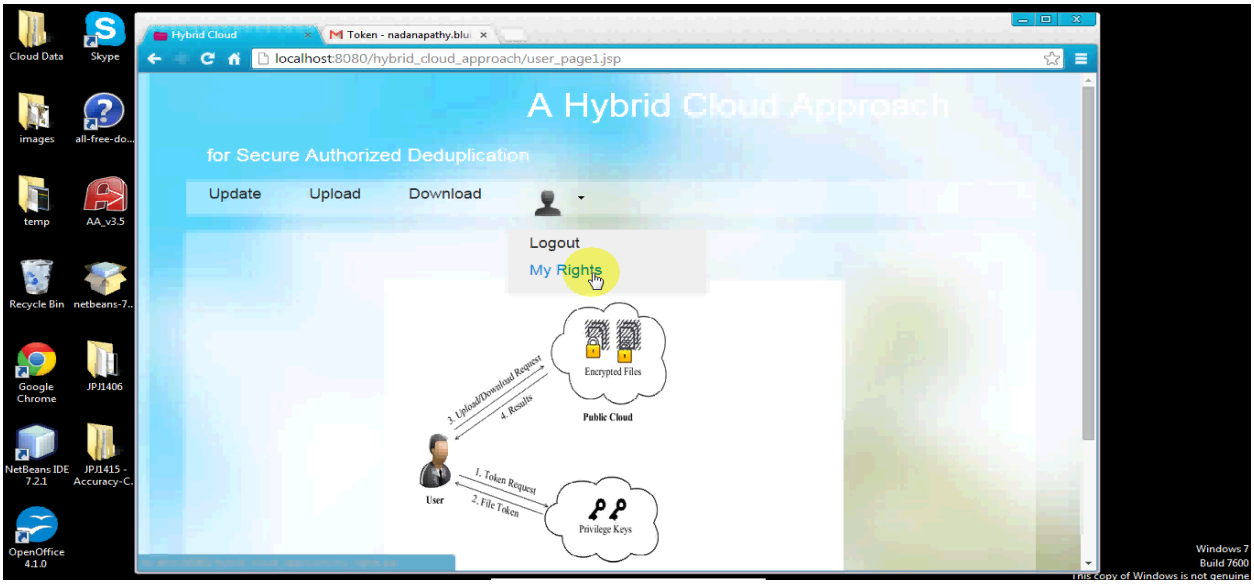

FIG 6 USER ACCESS PUBLIC CLOUD ARCHITECTURE

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

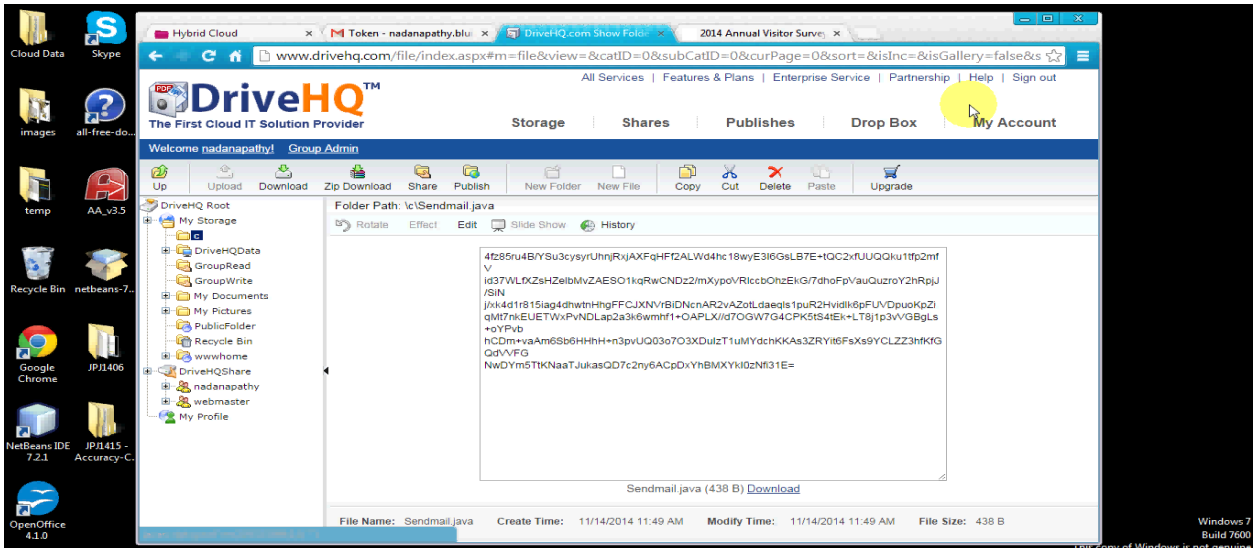

FIG 7 DRIVE HQ CLOUD STORAGE USED FOR ENCRYPTION

V.CONCLUSION AND FUTURE WORK

In this paper we discussed the problems existing in convergent encryption and We cope with this issue by adding one additional layer of deterministic and symmetric encryption on top of convergent encryption. This additional encryption can be added by a component placed between the user and the cloud storage provider such as a local server or a gateway**.** This component will take care of encrypting/decrypting data from/to users. In order to allow the cloud provider to detect duplicates, encryption and decryption are performed with one unique set of secret keys.

REFERENCES

[1]. S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg.Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis,and V. Shmatikov, editors, *ACM Conference on Computer andCommunications Security*, pages 491–500. ACM.
[2]. M. Bellare, S. Keelveedhi, and T. Ristenpart.Message-locked encryption and secure de-duplication. In *EUROCRYPT*, pages 296–312, 2013
[3]. J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer.Reclaiming space from duplicate files in a serverless distributed file system. In *ICDCS*, pages 617–624, 2002.
[4]. S. Quinlan and S. Dorward.Venti: a new approach to archival storage. In *Proc. USENIX FAST*, Jan2002.
[5] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure de-duplication with efficient and reliable convergent key management. In *IEEETransactions on Parallel and Distributed Systems*, 2013.