

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

Optimizing Fault Tolerance of HDFS using Efficient Coding Technique

Supriya Datkhile¹, Prof. P. R. Ugale²,

M.E. Student, Department of Computer Engineering, SPCOE, Otur, India¹

Assistant Professor, Department of Computer Engineering, SPCOE, Otur, India²

ABSTRACT: Users of cloud storage usually assign different redundancy configurations (i.e., $\delta k; m; wP$) of erasure codes, depending on the desired balance between performance and fault tolerance. Our study finds that with very low probability, one coding scheme chosen by rules of thumb, for a given redundancy configuration performs best. To begin with, CaCo utilizes Cauchy matrix heuristics to deliver a matrix set. Then for each matrix in this set, CaCo utilizes XOR schedule heuristics to produce a progression of schedules. At long last, CaCo chooses the most limited one from all the delivered schedules. In such a way, CaCo can distinguish an ideal coding plan, inside the ability of the current best in class, for a self-assertive given excess arrangement. By leverage of CaCo's nature of ease to parallelize, we boost significantly the performance of the selection process with abundant computational resources in the cloud. We implement CaCo in hadoop distributed file system and calculate its performance compared with Hadoop-EC.

KEYWORDS: Sharing Data; Data Sharing; Collusion Attack; Key Transfer.

I. INTRODUCTION

Cloud computing is the biggest buzz in computer world now a days. Cloud computing is an internet based modern technique, which is capable of providing us various resources. It is providing excellent facilities by flexible infrastructure. Cloud computing is based on client-server architecture. Cloud computing is a hub of various server and many database to store data. The availability of these resources are very flexible in nature i.e. few are available to customers free of cost but some on a pay as use basis. Along with this the customer is also allowed to access information and can utilize computer resources from anywhere if having the internet access.

As storage systems have grown in size, scale and scope, the failure protection offered by the system is in many cases no longer sufficient. These existing systems protect to a maximum of two simultaneous disk or node failures. However, proliferation of components, the requirement of wide-area networking and enriched failure modes have led storage designers to tolerate larger numbers of failures. To solve this issues of storage space we incorporate CaCo that is Cauchy Coding approach implemented with hadoop distributed file system.

CaCo selects the globally optimal schedule from all the locally optimal schedules. This globally optimal schedule and its corresponding matrix will be stored and then used during data encoding and decoding. Incorporating all existing matrix and schedule heuristics, CaCo has the ability to identify an optimal coding scheme, within the capability of the current state of the art, for an arbitrary given redundancy configuration. CaCo provide a secure way to handle these failures which is occur during cloud storage space. To tolerate more failures than RAID, many storage systems employ Reed-Solomon (RS) codes for fault-tolerance. Reed-Solomon coding has been around for decades, and has a sound theoretical basis. As an erasure code, Reed-Solomon code is widely used in the field of data storage. Given k data blocks and a positive integer m , Reed-Solomon codes can encode the content of data blocks into m coding blocks, so that the storage system is resilient to any m disk failures. Reed-Solomon codes operate on binary words of data, and each word is composed of w bits, where $2w - k \leq m$. In the rest of this paper, we denote a combination of k , m , and w by a redundancy configuration $\delta k; m; wP$, where k data blocks are encoded into m coding blocks, with the coding unit of w -bit words. Reed-Solomon codes treat each word as a number between 0 and $2w - 1$, and employ Galois Field arithmetic over $GF(2w)$. In $GF(2w)$, addition is performed by bitwise exclusive-or (XOR), while multiplication is more complicated, typically implemented with

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

look-ups to logarithm tables. Thus, ReedSolomon codes are considered expensive. Cauchy Reed-Solomon (CRS) codes improve Reed-Solomon codes by using neat projection to convert Galois Field multiplications into XOR operations. Currently, CRS codes represent the best performing general purpose erasure codes for storage systems. In addition, CRS coding operates on entire strips across multiple storage devices instead of operating on single words. In particular, strips are partitioned into w packets, and these packets may be large. The redundancy configuration of the system is $k \frac{1}{4} 4$ and $m \frac{1}{4} 2$. With CRS codes, k data blocks are encoded into m coding blocks. In such a way, the system can tolerate any m disk failures without data loss. Note that those k data blocks and m coding blocks should be stored on different data nodes. Otherwise, the failure of one node may lead to multiple faults in the same group of $n \frac{1}{4} k \text{ } \text{ } m$ blocks.

Objectives:

1. Provide efficient approach for Cloud Storage System.
2. Avoid component failures, and fault tolerance which have been further increased. So, main aim is to maintain the failure protection of system.
3. Provide low cloud storage cost as compared to existing one.
4. Providing accuracy for performance of system.

II. RELATED WORK

D. Ford, F. Label [1] Highly available cloud storage is often implemented with complex, multi-tiered distributed systems built on top of clusters of commodity servers and disk drives. Sophisticated management, load balancing and recovery techniques are needed to achieve high performance and availability amidst an abundance of failure sources that include software, hardware, network connectivity, and power issues.

James S. Plank, Catherine D [2] presents Large scale, archival and wide-area storage systems use erasure codes to protect users from losing data due to the inevitable failures that occur. All but the most basic erasure codes employ bit-matrices so that encoding and decoding may be effected solely with the bitwise exclusiveOR (XOR) operation. There are CPU savings that can result from strategically scheduling these XOR operations so that fewer XOR's are performed. It is an open problem to derive a schedule from a bit-matrix that minimizes the number of XOR operations. This paper design a technique to attack this open problem, deriving two new heuristics called Uber-CHRS and X-Sets to schedule encoding and decoding bit-matrices with reduced XOR operations. Proposed system evaluate these heuristics in a variety of realistic erasure coding settings and demonstrate that they are a significant improvement over previously published heuristics.

Chuanyi Liu, Xiaojian Liu [4] define A policy based de-duplication proxy scheme is proposed in this paper. It suggests a policy-based de-duplication proxy scheme to enable different trust relations among cloud storage components, de-duplication related components and different security requirements. Further proposes a key management mechanism to access and decrypt the shared de-duplicated data chunks based on Proxy Re-encryption algorithms. This paper finally analyses the security of the scheme.

G. Xu et al[7]define various tries have been set out to get this point. At to begin with, individuals find that the thickness of a Cauchy matrix coordinates the amount of XORs. In light of this, a measure of work has attempted to plan codes with low thickness. Moreover, some lower limits have been construed on the thickness of MDS Cauchy lattices. Inside the bleeding edge best in class, the least demanding approach to discover most decreased thickness Cauchy cross sections is to distinguish most of the systems and pick the quality one.

Vibha V Dambal et al [10] This paper explores recovery solutions based on regenerating codes, which are shown to provide fault-tolerant storage and minimum recovery bandwidth. It presents specification of algorithm for write operation and how to implement it Cloud storage systems always use different redundancy configurations (i.e., $(k; m;w)$), depending on the desired balance between performance and fault tolerance. For different combinations of matrices and schedules, there is a large gap in the number of XOR operations, and no single combination performs best for all redundancy configurations.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

Drawbacks of Existing System:

1. Performance of system is poor.
2. Balance computation problem so need to update complexity for code
3. The cloud storage cost in existing WAS system is very high.

III. PROPOSED WORK

Many efforts have been devoted to achieve this goal. Initially, people discover that the density of a Cauchy matrix dictates the number of XORs. For this reason, an amount of work has sought to design codes with low density. Moreover, some lower bounds have been derived on the density of MDS Cauchy matrices. In the current state of the art, the only way to discover lowest-density Cauchy matrices is to enumerate all the matrices and select the best one. Given a redundancy configuration (k, m, w) our goal is to find a Cauchy matrix, whose schedule is desired to be the shortest. In this paper, we propose CaCo, a coding approach that incorporates all existing matrix and schedule heuristics, and therefore is able to discover an optimal solution for data coding in a cloud storage system, within the capability of the current state of the art.

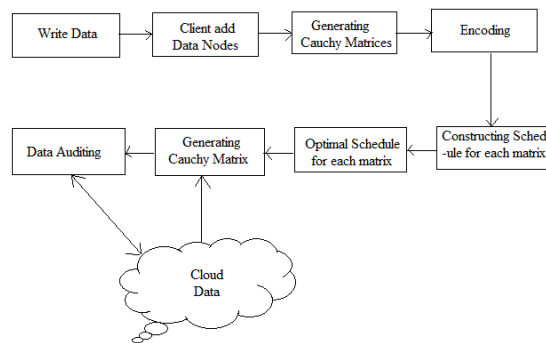


Fig. 1. Block Diagram of Proposed System

Generating Cauchy Matrices:

Selecting the best one from Cauchy matrices using the enumeration method is a combinatorial problem. Given redundancy configuration $(10; 6; 8)$, the magnitude of the matrices to be constructed can be up to 1029, and it is unrealistic to enumerate them. We cannot even determine which one of the matrices will produce better schedules. In the CaCo approach, we choose only a certain number of them for scheduling.

1. Constructing schedules for each matrix:

For each matrix $m_i (0 < i < p)$ in the set S_m , we pass the parameters including k, m, w and pointer of the matrix title function `do schedule (intk,intm,intw,int *matrix)` to perform q heuristics in the function, such as Uber -CSHR, X-Sets, and so on. In this manner, we get a set of schedules, denoted as $S_{s_i} = \{s_{0i}, s_{1i}, \dots, s_{q-1i}\}$. If there appears good heuristic for scheduling at a later date, we can add into the function `do schedule`.

2. Selecting the locally optimal schedule for each matrix:

For each matrix $m_i (0 < i < p)$ in the set S_m , we select the shortest schedule from the set S_{s_i} , denoted as s_i , so that we get a set of matrices and their shortest schedules, denoted as $S = \{(m_0, s_0); (m_1, s_1); \dots; (m_{p-1}, s_{p-1})\}$. For m_i in the set S_m , we can encode data in an order of XORs given by s_i . In this way, the times of XOR operations no longer have direct relationship with the density of the matrix. Therefore, scheduling excludes the influence of the lower limit of the number of ones in the matrix, so the performance improves significantly.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

3. Selecting the globally optimal solution:

From the collection of combinations of Cauchy matrix and schedule, namely $\{(m_0; s_0); (m_1; s_1); \dots; (m_{p-1}; s_{p-1})\}$, we choose the combinations with the shortest schedule. On this basis, for better performance, we tend to select the one containing the fewest ones in the matrix to be $(m_{best}; s_{best})$. Once selected, s_{best} can be used for encoding data.

Advantages of Proposed System:

1. Cloud storage systems always use different redundancy configurations.
2. There is a large gap in the number of XOR operations, and no single combination performs based for all redundancy configuration.
3. The performance of system is better as compare to existing one.

IV. EMPLOYING CaCo IN HDFS

To achieve data redundancy, HDFS requires triple replication. In CaCo, we use erasure codes instead of triple replication of data to guarantee the fault tolerance of the system. To implement CaCo, we should modify the architecture of HDFS to certain extent. For example, the illustration of write operation with CaCo is shown below and we can conclude the procedure to several steps as follows.

Algorithm 1: Write Operation with CaCo:

1. The Client sends a write request to the NameNode.
2. The NameNode allocates some DataNodes to the Client.
3. Write the data blocks into DataNodes.
4. Make a copy of data and put it into DataQueue.
5. Encode data with the schedule selected by CaCo.
6. Write the coding blocks into DataNodes.
7. Data encoding finishes.
8. Remove the copies of data from DataQueue.

The above algorithm for write operation can be shown in pictorial representation of how data will flow from name node to data node.

Modules of Proposed System:

A. Modules:

1. The Coordinator dispatches computational tasks
2. Parallel Deployment of CaCo in MPI
3. Performance Evolution
4. Running Time of CaCo
5. Report Generation.

B. Modules Description:

The Coordinator dispatches computational tasks:

In order to accelerate the selection of coding schemes, we perform CaCo in parallel by distributing the computational tasks to individual nodes in a cluster. For different heuristics, the calculations of Cauchy matrices and schedules are

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

completely independent. Therefore, it would be feasible and effective to distribute the tasks of generating matrices and schedules to different processors.

The Coordinator receives the parameters passed in, such as the redundancy configuration ($k; m; w$), and the number of Workers. Then the Coordinator sends some parameters to Workers, such as ($k; m; w$), HM (heuristic for generating a Cauchy matrix), and the other parameters required. In this procedure, we should concern about load balance among Workers.

Parallel Deployment of CaCo in MPI:

We deploy CaCo in MPICH with the release version 3.0.4, so that the selection of the best coding scheme can be accelerated in parallel over the distributed system. MPICH is a high-performance and widely portable implementation of the Message Passing Interface (MPI) standard. Our distributed environment is built up with multiple nodes, containing one node as Coordinator for dispatching computational tasks and assembling results, and other nodes as Workers for generating matrices and schedules.

To achieve data redundancy, HDFS requires triple replication. In CaCo, we use erasure codes instead of triple replication of data to guarantee the fault tolerance of the system. To implement CaCo, we should modify the architecture of HDFS to certain extent. With erasure coding, we reduce the number of data copies from three to one. So, when a block is written, we only allocate one Data Node to the Client.

Performance Evolution:

To evaluate CaCo's performance in every aspect, we do a set of experiments and analyses. First, while generating the combinations of matrix and schedule in every heuristic, we count the size of the generated schedules and the number of ones in the Cauchy matrices, to prove that CaCo can identify the optimal combination. Second, we run the framework with various redundancy configurations and accelerate it in MPI, to testify the feasibility of parallelizing in CaCo. Finally, we implement CaCo in HDFS and evaluate its performance of data encoding and decoding by comparing with Hadoop-EC. The test-bed used in these experiments consists of two nodes, as described below. The first node, installed with Linux kernel 2.6.35-22 generic, is equipped with Intel Xeon E5-2620 2.00GHz six-core processor and 32 GB of memory. The other node, installed with Linux kernel 2.6.32-33-generic, is equipped with Intel Xeon E5606 2.13GHz quad-core processor and 8 GB of memory.

Running Time of CaCo:

The running time of CaCo takes on an upward trend as increases along the x-axis. When one Worker is used to compute all the selection tasks, the running time increases from 43.26 s to 529.68 s. When four Workers are used to share the computing burden, the running time increases from 16.46 s to 169.24 s. In other words, the longest running time is 2.82 minutes. Second, for a given redundancy configuration, the running time of CaCo decreases significantly when four Workers are used. The speedup ratio varies between 2.63 and 3.15.

Report Generation:

Cloud storage systems always use different redundancy configurations (i.e., ($k; m; w$)), depending on the desired balance between performance and fault tolerance. For different combinations of matrices and schedules, there is a large gap in the number of XOR operations, and no single combination performs best for all redundancy configurations. We propose CaCo, a new approach that incorporates all existing matrix and schedule heuristics, and thus is able to identify an optimal coding scheme within the capability of the current state of the art for a given redundancy configuration. The selection process of CaCo has an acceptable complexity and can be accelerated by parallel computing. It should also be noticed that the selection process is once for all. The experimental results demonstrate that CaCo outperforms the "Hadoop-EC" approach by 26.68-40.18% in encoding time and by 38.4-52.83% in decoding time simultaneously.

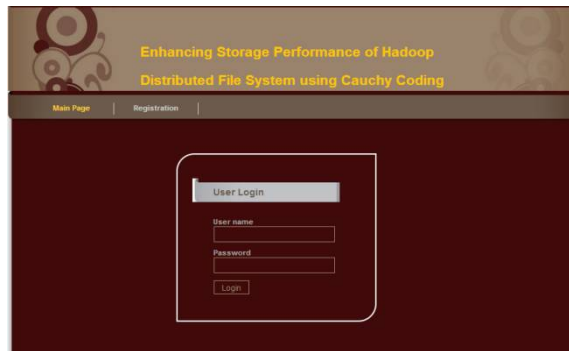
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

V. SNAPSHOTS



Enhancing Storage Performance of Hadoop
Distributed File System using Cauchy Coding

Main Page | Registration

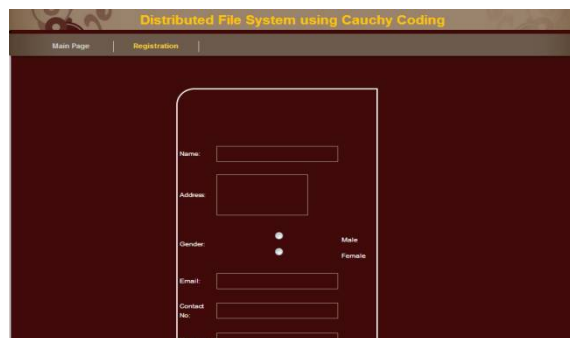
User Login

User name:

Password:

Log In

Fig. 1. User Login



Distributed File System using Cauchy Coding

Main Page | Registration

Name:

Address:

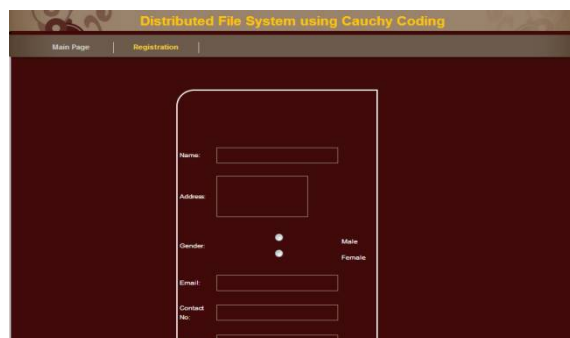
Gender: Male Female

Email:

Contact No:

Phone No:

Fig. 2. User Registration



Distributed File System using Cauchy Coding

Main Page | Registration

Name:

Address:

Gender: Male Female

Email:

Contact No:

Phone No:

Fig. 3. Registration Form

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017



Fig. 4. Main Page

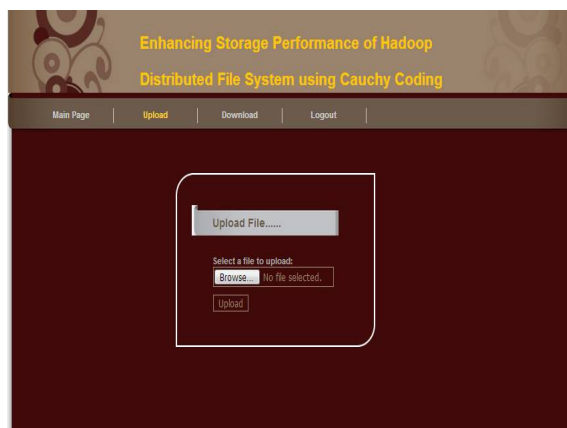


Fig. 5. Upload File

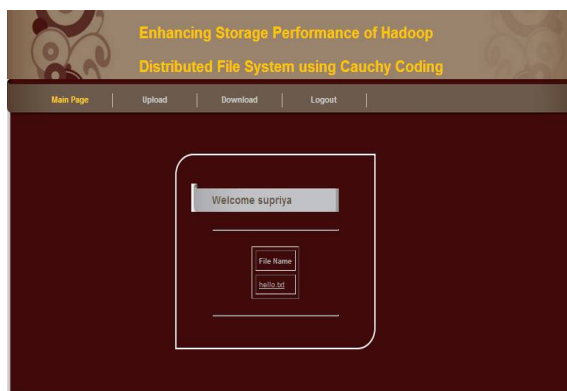


Fig. 6. Upload File Successfully

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

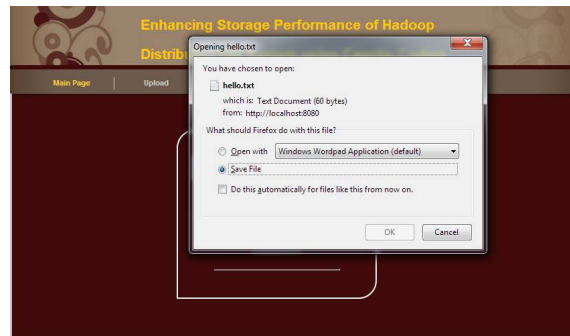


Fig. 7. Save File

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose CaCo, a Cauchy Coding approach which produce matrix set and schedule heuristics. CaCo avoid component failures, and fault tolerance. CaCo has the ability to identify an optimal coding scheme, within the capability of the current state of the art, for an arbitrary given redundancy configuration The experimental results indicate that CaCo outperforms the “Hadoop-EC” approach by 26.68-40.18% in encoding time and by 38.4-52.83% in decoding time simultaneously.

In future, the Cauchy’s rule can be incubated with bandwidth of performance with respect to time for series computation of delay in network node retiring. The delay time reduction under narrow network is still a bottle neck situation. Either of this can be improvised with technical reduction of data sets and its indexing.

REFERENCES

- [1]. Guangyan Zhang, Guiyong Wu, Shupeng Wang, Jiwu Shu, Weimin Zheng, and Keqin Li, “CaCo: An Efficient Cauchy Coding Approach for Cloud Storage Systems,” *IEEE Transactions On Computers*, Vol. 62, No. 11, November 2015.
- [2]. D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, “Availability in globally distributed storage systems,” in *Proc.9th USENIX Symp. Oper. Syst. Des. Implementation*, 2010, pp. 61–74.
- [3]. B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Adusumilli, M. McNett, S. Sankaran, K. Manivannan, and L. Rigas, “Windows Azure storage: A highly available cloud storage service with strong consistency,” in *Proc. 23rd ACM Symp. Oper. Syst. Principles, New York, NY, USA*, 2011, pp. 143–157.
- [4]. Osama Khan, Randal Burns, James Plank, William Pierce, “Rethinking Erasure Codes for Cloud File Systems : Minimizing I/O for Recovery and Degraded Reads,” *MSR-TR*, May 2013.
- [5]. J. S. Plank, “A tutorial on reed-solomon coding for fault-tolerance in raid-like systems,” *Software Practice Experience* , vol. 27, pp. 995-1012, Sept. 1997.
- [6]. B. Fan, W. Tantisiroj, L. Xiao, and G. Gibson, “Diskreduce: Raidfor data-intensive scalable computing,” in *Proc. 4th Annu. WorkshopPetascale Data Storage, New York, NY, USA*, 2009, pp. 6-10.
- [7]. K. D. Bowers, A. Juels, and A. Oprea, “Hail: A high-availabilityand integrity layer for cloud storage,” in *Proc. 16th ACM Conf.Comput. Commun. Security, New York, NY, USA*, 2009,pp. 187-198.
- [8]. G. Xu, F. Wang, H. Zhang, and J. Li, “Redundant data composition of peers in p2p streaming systems using CauchyReed-Solomon codes,” in *Proc. 6th Int. Conf. Fuzzy Syst. Knowl. Discovery-Volume 2, Piscataway, NJ, USA*, 2009, pp. 499-503.
- [9]. C. Huang, J. Li, and M. Chen, “On optimizing xor-based codes forfault-tolerant storage applications,” in *Proc. IEEE Inform. TheoryWorkshop*, 2007, pp. 218-223.
- [10]. J. Edmonds, “Paths, trees, and flowers,” *Can. J. Math.*, vol. 17, no.3, pp. 449-467, 1965.
- [11]. M. Nabeel, N. Shang, and E. Bertino, “Privacy preserving policy based content sharing in public clouds,” *IEEE Trans. Know. Data Eng.*, vol. 25, no. 11, pp. 2602– 2614, Nov. 2013.
- [12]. X. Liu, Y. Zhang, B. Wang, and J. Yang, “Mona: Secure multiowner data sharing for dynamic groups in the cloud,” *IEEE Trans.Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1182– 1191, Jun. 2013.
- [13]. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” *Commun. ACM*, vol. 53, no. 4,pp. 50–58, Apr. 2010.
- [14]. B. Dan and F. Matt, “Identity-based encryption from the weil pairing,” in *Proc. 21st Annu. Int. Cryptol. Conf. Adv. Cryptol.*, 2001, vol. 2139, pp. 213–229.