Who is More Reliable? An Evaluation Method for Distributed-memory Aggregation in the Internet

Hongyi Chen, Zhigang Sun, Bin Huang, Tao Li and Xilong Mao
School of Computer, National University of Defense Technology
Changsha, P. R. China
chychenhongyi@gmail.com, sunzhigang@263.net, huangbinnudt@gmail.com

ABSTRACT

Traditional disk-oriented storage systems can no longer meet the high requirements of applications in latency and bandwidth, for which RAMCloud [4], was proposed to replace the disk with RAM. We consider using the memory of end-users in existing Internet to achieve a similar memory storage system. However the end-users' behavior are not as controllable as the servers in data center, which would lead to a high data loss rate. So how to ensure the data's reliability is the biggest challenge. To improve the reliability, as well as the resource utilization, we present a thought of Contributor scoring, establishing a scoring model according to Contributors' behavior. This model will classify Contributors into certain levels. Moreover, we train the Model with genetic algorithms. Experimental results show that the optimized model can effectively divide the Contributor, ensure certain reliability.It has great significance to this user memory storage system for improving the reliability and resource utilization.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Miscellaneous

Keywords

RAMCloud, Contributor scoring, Genetic Algorithm

1. INTRODUCTION

Disk-oriented storage system's access latency and bandwidth cannot meet the needs; the authors of [4] proposed a random access memory storage model called RAMCloud in data center. We consider whether it's possible to achieve similar memory storage system in the Internet, arranging the end user's excess storage resources to provide storage services for P2P Cache [3], NDN, etc. Someone has done the research [1] that for the 2G-memory users in the edge network, there are almost 1G free spaces, so the end users have the ability to provide hundreds MB free memory to our memory storage system.

This user memory storage system as a distributed-memory aggregation storage model, just like RAMCloud, has a short data access delay and a high bandwidth. The biggest difference between the two is that RAMCloud deployed in a highly reliable data center, while the other deployed in the Internet. For the Internet end users' behavior is not as controllable as servers in data center, user memory storage system meet more challenges, such as how to prevent Contributors tampering with the data stored on their host, how to organize Contributors distributed-memory efficiently. From our perspective, improving the reliability of the system, reducing the possibility of data loss is the biggest problem faced in terms of usability. Reference [1] mentioned that distribute multi-copies to resolve this issue, but it greatly reduces Contributors resource utilization.

In this paper we attempt to resolve the utilization by Contributor scoring. According to the score, we can divide the Contributors into different grades to meet the different reliability level storage needs. Moreover, after train the scoring model with genetic algorithms, experiments show that it can make an effective division on Contributors.

2. PROBLEM DESCRIPTION

In this kind of user memory storage system, Contributors promised to contribute a certain period of time and a certain size of memory. While they may shut down their host for some reason without accomplishing their promising time, and it would cause a data loss and reduce system stability. So the system must ensure the appropriate level of failure rate and it's necessary for applications to specify the level of reliability for each request. When a high-level cache is stored in a high loss rate Contributor, the system must increase the backup number.

When caching data in this user memory storage system, the applications' data may be divided into multi-blocks (D_1, D_2,D_3,$\cdots$,D_N) and stored in different Contributors. The applications' data failure rate φ is directly related to Contributors' data loss rate. φ=f(θ_1, θ_2, θ_3,$\cdots$,θ_n), θ_i denotes the failure rate of block D_i. As block D_i may have multiple copies, stored on Contributors with the loss rate σ_j, then $\theta_i = \sigma_1\ \sigma_2 \cdots \sigma_k$.

In order to measure Contributors' loss rate, we proposed a idea, Contributor scoring, figuring out the score according to Contributor's behavior along with the deposit time and deposit memory size. Generally, a high score means a low loss rate.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CoNEXT Student'12, December 10, 2012, Nice, France.
Copyright 2012 ACM 978-1-4503-1779-5/12/12 ...$15.00.

3. GENETIC ALGORITHM ON CONTRIBUTOR SCORING

3.1 Contributor Scoring

Contributor scoring basic idea is: calculate a quantitative weighted score according to its behavior. Contributors will be divided into several levels; the loss rate would be almost the same when two contributors are at the same level. Firstly, we must analyze clearly what Contributor behavior may affect the score of the Contributor.

(1) For the Contributors achieving the promised time, give the Contributors' score appropriate enhancement.

(2) More memory and longer they promise, the complete contribution would enhance a higher score.

(3) With a lower average access delay, the Contributor's score are relatively higher.

Then we could conclude that Contributor scoring parameters are (contribution and promise to contribute time, average Memory size per contribution, average access latency).

However, it is a very difficult problem that how to get the weights of these parameters. We consider a genetic algorithm to solve this problem. For simplicity, the problem is simplified into only dividing the Contributors into two categories, Good Contributor and Bad Contributor..

3.2 Genetic Algorithm

Parameters which affect Contributors' score are defined as a vector $X = (\frac{servicetime}{promisetime}$, average Memory size per contribution, average access latency$) = (x_1, x_2, x_3)$; the vector $W = (w_1, w_2, w_3)$ denotes different weights for each attribute. Y represents a score calculated according to the equation (1). Comparing Y with the given value c, we would be able to determine which category the contributor belongs to according to equation (2)(3).

$$Y = W \cdot X^T = w_1x_1 + w_2x_2 - w_3x_3 \quad (1)$$

$$w_1x_1 + w_2x_2 - w_3x_3 \geq c(good) \quad (2)$$

$$w_1x_1 + w_2x_2 - w_3x_3 < c(bad) \quad (3)$$

By adopting the approach of [2] to this contributor scoring model, we should select the genes. We choose the the weights $W = (w_1, w_2, w_3)$ as a population's gene. Considering that x_1's value is relatively small, we define x_1 ranging from 0 to 100, while x_2, x_3 range from 0 to 10.

4. PRELIMINARY EXPERIMENT

We write a program, generating 30 data sets randomly, and obtain the value of constant c (33.0667) by adopting the approach of [2]. According to the training results Good Contributor and Bad Contributor judgment accuracy are 92.3% and 94.1% respectively, indicates that the scoring model can effectively classify the Contributors.

Figure 1 shows Contributors' loss rate and its corresponding score under the trained model. We can see that the higher the score, the lower the failure rate, which means the better reliability. The Contributor scores equal to c (33.0667), the loss rate would be 23.7%. It can be figured out that if data stored on Good Contributor, only one backup can achieve data loss rate less than 5.6%. However when backup data on Bad Contributors, 3-4 replicas would be needed to meet the reliability requirements. We also notice that the Contributor, whose ID is 15, scores higher than the

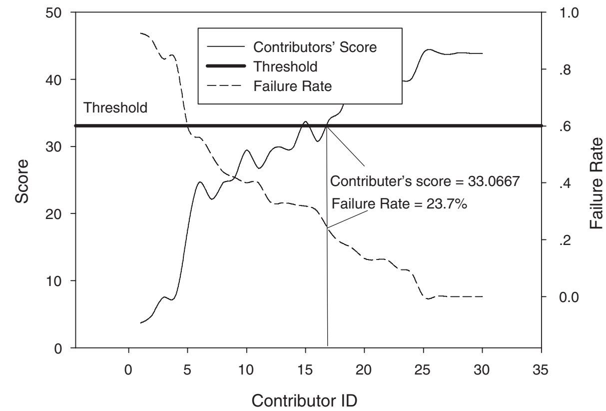

Figure 1: The Contributor's score and its corresponding loss rate

threshold c, judged as a good Contributor by our scoring model, but its loss rate is higher than 23.7%. This is because our scoring model gives a bonus to large memory Contributors. This also confirms that our model encourages Contributors contribute memory, the more memory they contribute, the higher score they will gain.

5. CONCLUSION AND FUTURE WORK

This paper presents a Contributor scoring model to solve the reliability problems of user memory storage system, experimentally derived scoring model can screen out the high loss rate Contributors. While, our model only have three parameters, future work should dig out more parameters to achieve a higher accuracy. In addition, how to perform a adaptive mechanism to the model parameters is also need to be considered in the future.

6. ACKNOWLEDGEMENT

This work was supported by the National Basic Research Program of China (No.2009CB320503, No.2012CB315906) and the National High Technology Research and Development Program of China (No.2011AA01A10).

7. REFERENCES

[1] H. Bin, S. Zhigang, C. Hongyi, and Z. Ziwen. Bufferbank: A distributed cache infrastructure for peer-to-peer application. *Peer-to-Peer Netw. Appl.*, 2012.

[2] W. Cai, X. Lin, X. Huang, and H. Zhong. A genetic algorithm model for personal credit scoring. In *Proceedings - CiSE 2009*, Wuhan, China, 2009.

[3] M. Hefeeda, C.-H. Hsu, and K. Mokhtarian. Design and evaluation of a proxy cache for peer-to-peer traffic. *IEEE Transactions on Computers*, 60(7):964 – 977, 2011.

[4] J. Ousterhout, P. Agrawal, D. Erickson, C. Kozyrakis, J. Leverich, D. Mazieres, S. Mitra, A. Narayanan, D. Ongaro, G. Parulkar, M. Rosenblum, S. M. Rumble, E. Stratmann, and R. Stutsman. The case for ramc loud. *Communications of the ACM*, 54(7):121 – 130, 2011.