Disaster Recovery in the Multi-cloud Model

Fang Cui [a], Yong Liu

Henan University of Science and Technology, Luoyang 471000, China

[a]cuifang007@126.com

Abstract

With the rapid development of cloud computing model, the use of cloud resources for disaster recovery has become an attractive method.However,adoption of disaster recovery remains limited because of this model is not stable.In this paper,we presents a method: using multi-cloud and pipelined synchronous replication as an antidote to there problems. A number of cloud service providers can cooperate with each other, but different cloud service providers have a certain geographical distance, so it will produce a certain replication latency on performance. Pipelined synchrony addresses the impact of replication latency on performance, by efficiently overlapping replication with application processing for multi-tier servers. Multi-cloud proposes multiple optimization scheduling strategies to balance the disaster recovery objectives which are also transparent to the customers, such as high data reliability, low backup cost, and short recovery time.By tracking the disk recovery site continued to modify the results, all the way indicating the message to customers, in the event of a disaster, application to achieve forward progress while maintaining consistent client visible state guarantee.Experiment shows that the multi-cloud can effectively cooperate with the cloud service providers with various parameters and pipelined synchrony can improve the throughput of an order of magnitude, and the synchronous replication reduces the response time while guaranteeing zero data loss.

Keywords

Multi-Cloud; Disaster Recovery; Pipelined Synchronous Replication.

1. Introduction

In current society, more and more dependent on the key computer system, means that even a very short period of downtime may cause serious damage to property or social problems, in some cases, even a threat to human life. Many commercial and government services use disaster recovery systems to minimize downtime caused by catastrophic system failures. Such as financial service and health service. As a result, many important business and public service use disaster recovery mechanism, in order to protect the key data, and try to reduce downtime caused by a catastrophic system failure. By reducing the time and data loss recovery due to disaster, a disaster recovery service can also provide business continuity, but usually the cost is higher. Because Greenberg [1] et al has shown that the cost of a data center includes purchasing servers and infrastructure, maintaining facilities, and employing human resource. And there is no difference in cost no matter whether the service is standby or in use. So, if a service provider chooses to build its own disaster recovery data center, it needs huge investments, and generates huge idle resources, resulting in a waste of resources.

Over the past ten years, cloud computing has become more and more popular as a new service model. At present, a large number of services are built on the cloud platform. Wood [2] et al. Has demonstrated that the cost of using existing cloud resources for data disaster recovery is far less than the cost of building and maintaining disaster recovery data centers for private data centers. In addition, the cloud platform has the ability to quickly activate the required resources after the disaster, and to minimize the cost of recovery. The on-demand of cloud computing has greatly reduced the computational cost, and its peak demand is much higher than the average demand. However, data disaster recovery is one of the highest reliability requirements for data services. It is still a challenge how to make use of the cloud computing model for data disaster recovery services, and to maximize the reliability of the data and to reduce the cost. There is no difference from other computer systems, cloud computing systems may also encounter certain risks, such as software errors, hardware failures, network intrusions, man-made damage, and natural disasters and so on. All of these risks can lead to cloud services interruption, in some cases even cause data loss. In order to ensure the reliability of the data, cloud service providers have been deployed a number of data protection strategies. However, once the data center crashes, the data may still be lost. To avoid this problem, some cloud service providers use geographic data distributed technology to protect the most critical data, and data center in different regions have a mostly similar software stack, infrastructure purchased in bulk, operating mechanism and working team cloud service providers. In a period of time, there are still risks of failure of multiple data centers across the data center because of some common reasons. In addition, the number of data centers owned by a cloud service provider is limited. When some of these data centers cannot be connected, the other data centers may be not applicable to certain customers because of the long geopolitical distance and the network delay in the emergency cases. Therefore, no matter how many preventive measures have been taken, the possibility of interruption of the reliability data in the cloud cannot be ignored. According to some public reports [3, 4], even the most advanced cloud services also encountered several outages, and led to many public service disruptions.

So, we believe that the use of multiple data centers in different cloud service providers to carry out disaster recovery services is the best solution. In this scheme. A cloud service provider that provides data disaster recovery service to the public can be rented from other cloud service resources in the pay-as-you go mode. Therefore, data disaster recovery is no longer confined to one data center of cloud service provider.A disaster recovery service provider can selectively back up data in its own data center or other cloud service provider's data centers [5].By using the appropriate backup data scheduling strategy, disaster recovery service providers can achieve better recovery effect, that can improve the quality of service and also can reduce the cost.

In this paper, we propose a system model based on cloud disaster recovery service. Multi-cloud uses a number of cloud service provider resources. Customers only need to deal with multi-cloud, using very common and unified service interface. Due to the presence of a certain geographical distance, a number of cloud service providers have a certain network delay. Consequently, we solve this problem by effectively overlapping replication with the application for muli-tier servers.

2. Related Work

Cumulus [6] can back up a file system to cloud storage, using a least-common-denominator cloud interface, thus support many kinds of cloud services. It only use a cloud to maintain a backup, and focus on the mechanism in the local file system, rather than the cloud platform. Wood [2] et al proposed a new cloud service model, i.e., disaster recovery as a cloud service, which leverages the virtual platforms in cloud computing to provide data disaster recovery service. They created a disaster recovery cloud model for a Web site application, which explains the data backup in the cloud resource, and it can greatly reduce the cost of disaster recovery for enterprise data. But they did not study how to use a number of clouds to further improve the quality of service. Some researchers focus on how to backup data in cloud computing environment, such as Bajpai [7] et al. and Zhang [8] and Zhang. CABdedupe [9] employs the deduplication technology to remove redundant data from transmission during backup and recovery, which reduces time and network traffic consumption. Li [10] et al. proposed a data replication strategy which uses an incremental replication method to reduce network and storage cost. R-ADMAD [11] exploits ECC codes to encode data chunks, then distributes them among storage nodes within a data center, and uses a distributed dynamic restoring process to handle data recovery. Compared with the scheme based replication, it greatly reduces the memory footprint, while ensuring a relatively high reliability of the data. Pipe Cloud is a pipe lined synchronous replication based disaster recovery system, designed to increase throughput and reduce response time while providing zero data loss consistency guarantees [12]. But it only use a copy of a cloud storage data. Nguyen [13] et al. proposed a differentiated replication strategy that can handle customers' different requirements. This strategy provides a reliable guarantee for the reliability of the data and a differentiated backup plan for different service type, and it also improves the utilization of cloud resources. Cachin [14] et al. analyzed the data integrity and data security in the intercloud mode. In a number of cloud platforms, they use fault-tolerant protocol and secure access control to ensure data integrity and confidentiality. Metadata [15] storage focuses on the data consistency and communication latency data copies among multiple cloud providers. They set up a data access priority queue to reduce the communication latency between the clouds, and speculate that the smaller the file, the better to ensure data consistency.

The method mentioned above does not put forward a practical one, which can make use of multiple cloud resources to optimize the backup cost and recovery time, and to ensure the high reliability of the data disaster recovery service model.

3. Model of multi-cloud

3.1 Multi-cloud Philosophy

On the basis of the above analysis, we can see that the multi-cloud model is a kind of method which is suitable for the data disaster recovery service.But it also brings some challenges to the new design of a practical system model, and a summary is as follows:

a. The model should be able to make use of a variety of cloud platforms. At present, different cloud service providers provide different types of infrastructure, service interfaces, storage and network resources and charging model [16], such as Google, Amazon and Microsoft. By supporting all kinds of cloud platforms as much as possible, the disaster recovery service providers can take advantage of more resources and accommodate more customers in a wide geographic area.

b. In order to reduce the service barrier, the model must be transparent to end users. That is, from the users' point of view, the use of this model and the use of a common cloud model is no different. Customers do not need to consider the internal details of multiple clouds. Negotiating technical and financial parameters with other cloud service providers and implementation of data disaster recovery tasks are the responsibility of the service provider.

c. The model should provide sufficient space for optimization to achieve sufficient advantage. In this case, this means it can effectively choose to store the backup data from multiple cloud providers of resources such that the data disaster recovery costs as low as possible, and when a disaster occurs the recovery time as short as possible, while ensuring high data reliability.

In order to meet all these challenges, we designed a multi-cloud based data disaster recovery service model.

By using a least-common-denominator cloud storage interface, in other words, get and put of complete files, multi-cloud allow a service provider to use other cloud service providers' resources to establish data disaster recovery services to meet more customers' need, and improve its market reputation and company's revenue. In order to ensure the high reliability of the data, multi-cloud adopts 3-replicas data redundancy mechanism. The disaster recovery provider will intelligently select a resource from a number of cloud service providers based on a certain scheduling policy, including its own. This decision is based on storage costs, network communication costs and data recovery speed, so that it can meet two basic optimization objectives: as far as possible to reduce the cost of data backup and as far as possible to shorten the time of data recovery. All internal procedures are transparent to the customers. In this mode, whether it is a person or company or other cloud service providers can use multi-cloud.

3.2 Multi-cloud model

The overall structure of the multi-cloud is shown in Fig.1 and Fig.2, which are the data backup model and recovery model.

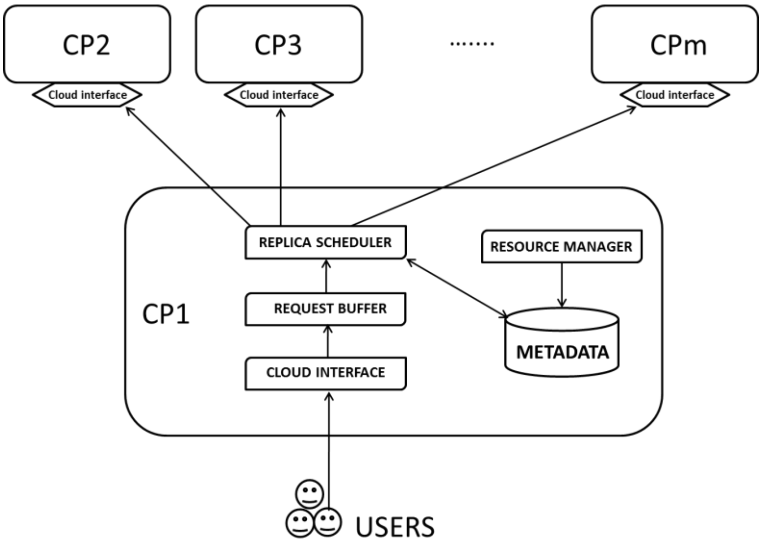

Fig.1 Data backup model

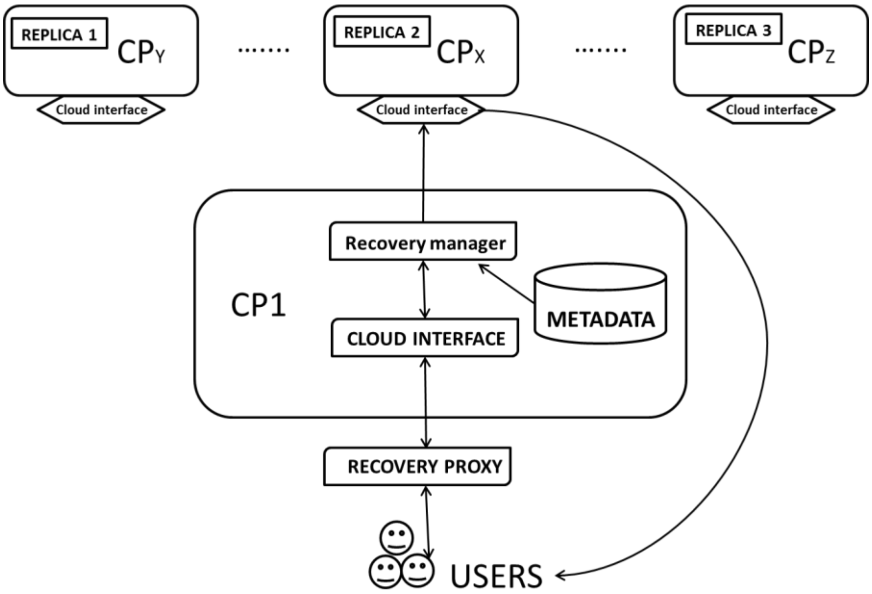

Fig.2 Data recovery model

In general, multi-cloud is comprised of data disaster recovery service customers and clouds service providers. In Fig.1, the CP1 represents the data disaster recovery service provider. All customers are the users of CP1, who can be a person, business and even other cloud service providers. They have the right account and CP1 privilege.CP2-CPm are other cloud service providers that provide common cloud resources for CP1. Every CP cloud interface is the lest-common-denominator cloud storage interface, which receives/sends data from/to the users.CP1 request buffer holds a data backup request to arrive at a time period, it will be arranged. Replica scheduler from the request buffer to read the request, so that each of the three copies of a set of copies distributed to those CPs. Resource manager is responsible for monitoring the changes in the use of all CPs resources. Metadata is a database that contains information about the location of the copy and the use of CPs resource.

In Fig.2, CPX, CPY and CPZ represent a cloud service providers that contain a copy of the data recovery request. The CP1 recovery manager receives and checks the recovery request, then choose a suitable CP which contains at least one copy of the request. Recovery proxy is the agent installed to the client, that is responsible for the recovery of data from the CPs.

3.3 Data backup model

The execution flow chart of the Cloud data backup program, as shown in Fig.1, is described as follows:

a. Customers will be asked to back up their data, including data storage and storage time parameters, through the CP1 cloud interface to send to CP1.

b. CP1 receives these requests and puts them in the request buffer.

c. CP1 checks client accounts and privileges, and will refuse illegal requests.

d. In each unit time or when there are certain number of requests in the request buffer,the replica scheduler triggers its scheduling policy.

e. The scheduling policy of replica scheduler reads the data size and storage time parameters of all requests from the request buffer, makes three copies for each data, and then intelligently determines the storage location of each copy. Then generating an overall data backup plan.

f. According to the scheme,replica scheduler sent to their destination, the destination may be CP1's own data center or other CPs data center.

g. Other CPs record CP1's occupancy of their resources.

h. When the data in the program is completed,CP1 to submit the location information of all data copies to the metadata database.

i. CP1 delete all requests to complete the request,then record the information of these requests to the customer charges.

j. CP1 send the received message to the client to return the appropriate data processing that can be used to recover certain data from CP1.

From the above description, the data backed up is sent to CP1 first, and then copy three copies, CP1 sent them to the final destination. As a matter of fact, there is another process can be selected: Firstly, only the backup data parameters are sent to CP1, then CP1 generates a backup plan and it is returned to the customers. Finally, according to the program customers to send their data to the corresponding CPs. In this process, the customer needs to send three copies of his data to multiple CPs. It can reduce throughput and increase the customer's application response time, and consume more customer bandwidth. In addition, due to the failure of data transmission, this process will make the location of the actual data and the consistency of the CP1 metadata database become more complex. So we don't think this is a good design choice.

From the customer's point of view, the data backup program in multi-cloud is very similar to the common data stored procedure in a cloud mode. Customers do not need to know the details of the multi-cloud process, but also do not need to record the actual data location. This reduces the multi-cloud service barriers.

3.4 Data recovery model

The implementation process of data recovery program in cloud computing, as shown in Fig.2, described as follows:

a. Before processing the corresponding data of CP1, the customer sends the data recovery request to CP1 through the cloud interface of CP1.

b. CP1 receives the request, checks the client's account and permissions, and if it's illegal, rejects the request.

c. The CP1 recovery manager looks for data processing in the metadata database to find the location of the backup data.

d. CP1 compares the location of the three copies of the data, and then selects the fastest location, that is, the one with the highest bandwidth. In the Fig.2, assumed the position is CPX.

e. Recovery manager notifies the CPX to establish a temporary access right, which can only be used to read the copy, and can only be used by the client to restore the agent.

f. CPX establishes the authorization, and then notifies the CP1.

g. CP1 sends the location of the data and authorization information to the recovery proxy of client.

h. Recovery proxy removes the data from the CPX, and then notifies the CP1.

i. CP1 notifies CPX to destroy the temporary access authorization and then record the information of the recovery request. According to the customer charges.

j. CPX records recovery request for network traffic consumption. And then according to this for charging.

In addition, if the CPX is CP1 itself, there is no temporary access needs to be set by the CP1, the recovery proxy will directly use CP1 customer accounts.

From the above description, the data is directly recovered from the CPX without repeating CP1 to reduce the recovery time when the disaster occurs. Due to the recovery time is particularly important we think this is a better choice. We know that even a very short system downtime may cause a huge economic losses. So disaster recovery service should to shorten the recovery time as much as possible. This process is very important for the use of temporary access authorization mechanisms. Because this mechanism can ensures data security in multi-cloud. Although CP1 has a permanent account of other CPs, but there accounts cannot lend CP1's customers, otherwise it will lead to security risks. In addition, the interface that the recovery proxy exposes to the client is the same as the common cloud storage data interface. The internal process of recovering data from multiple cloud is transparent to the customer by processing the CPs with the recovery proxy. It's like a data recovery program in a cloud model.

4. Multi-cloud scheduling strategy

The internal cloud architecture in multi-cloud, CP1 copy of the scheduler scheduling strategy is the most important, it determines the effect of multi-cloud. The responsibility of scheduling strategy in multi-cloud is to reduce the cost of data backup and shorten the data recovery time as far as possible.

We think scheduling strategy should be a one-time scheduling mode. That is to say, a copy of the data stays on one computer until it expired, during its lifetime, a copy of data do not be transmitted to another computer unless the data copy belongs to the computer is a permanent fault. When some of the better storage resources are released, migrating a copy consume additional network traffic. In fact, the subsequent backup request can also be effectively utilized, and the waste gain compared to the network resource can be ignored.

In multi-cloud, for CP1 data disaster recovery service, all CPs have their own storage resource constraints, and this may not be their total storage resource limitations for a variety of services. As a result, the backup request will compete with storage resource that have advantage. Due to the diversity of the request parameters, including data size and storage duration, make good scheduling decisions is a typical multi-bojective combinatorial optimization problem.

Therefore, we formulate the multi-cloud scheduling strategy as the optimization goal is the combination of two factors: data recovery cost and recovery time. It should be emphasized that the cost of data backup costs CP1, not the customer. So the lower the cost of CP1 backup, then the more CP1 profits.

In order to further discuss the scheduling problem, we use the following formula to carry out a formal description:

Assume that there are X cloud service providers:CP1、CP2......CPX. The i-th CP's price of storage space, price of network traffic, data recovery bandwidth, and storage resource limitation respectively are SPi (dollar/(GB ·h)), TPi (dollar/GB), BWi (Gbit/s), and SLi (GB).

In a certain scheduling cycle, there are Y data backup task, that is a valid requests, to go through CP1 processing:T1、T2......TN. The j -th task's data size and store duration respectively are Sj (GB), and Dj (h).

The cost of data backup includes data storage cost and network communication cost. Therefore, in the scheduling cycle, the backup cost of all tasks can be represented by COST:

$$COST = \sum_{j=1}^{Y}\sum_{i=1}^{X}(Sj \times Dj \times Rij \times SPi + Sj \times Eij \times TPi) \quad .$$

In this formula. Rij represents the count of data copy stored in CPi for task Tj. And Eij is a boolean variable. When CPi contains task Tj's data copy it should be 1. For example, when Rij is equal to or greater than 1, otherwise should be 0.

We can know: $\forall j \in [1, N], \sum_{i=1}^{X} Rij = 3$. $\forall i \in [1, X]$ and $\forall j \in [1, Y], 0 \le Rij \le 2$.

The recovery time, RTO, represents the length of time that a client sends a recovery request to CP1 until the customer gets all the data from a CP. Its main components include the recovery transmission start-up delay and transmission time. The rest part is a very small constant. So we know:

$$RTO = \sum_{j=1}^{Y} \underset{i=1}{\overset{X}{MIN}}(Sj \times Eij / BWi + Li \times Eij)$$

Now we have developed the optimization objectives: cost and recovery time objectives (RTO). And then we can quantitatively discuss scheduling strategies: (We use CPE to represent during the scheduling time a CP has enough storage space to contain)

(1)Random strategy: For each new task in the request buffer, send three copies randomly one by one to CPEs selected.

(2)Cost priority strategy: For each new task in the request buffer, send three copies one by one to select the CPE with the lowest backup cost. If the first and second copy are in the same CPE, then the third copy is sent to the other backup costs as low as possible to the CPE.

(3)RTO priority strategy: For each new task in the request buffer, send the first copy to CPE that has the shortest recovery time, if you can choose then send the second and third copies to the other CPEs randomly.

(4)COST/RTO trade off strategy: For each new task in the request buffer, if CP1 has enough storage space to contain two copies of the task, then store first and second copies in CP1,and send the third copy to the shortest recovery time of the CPE; if CP1 can contain a copy of the task, then send a second copy to the CPE with the shortest recovery time, and send a third copy to the CPE with the lowest backup cost; if CP1 can not contain an arbitrary copy task, then send the first copy to the CPE with the shortest recovery time, and send the second and third copy to the CPE with the lowest backup cost.

(5)Dynamic optimization strategy: All replicas created by CP1 in a scheduling cycle, using a multi-objective combinatorial optimization algorithm, and consider all the problems, then generate a sub-optimal backup plan, and then schedule the replicas according to the schedule. For the sake of generality, we use the MOPSO [17] algorithm.

As shown above, the first and the third strategy are simple and intuitive. The second strategy and the third strategy respectively tend to optimize cost and RTO. If possible, they are trying to make sure that three copies of each task are at least in the two CPs.

5. Pipelined Synchronous Replication

Multi-cloud model use multiple cloud service providers to help customers recovery data. But this model would have a certain network latency because of the geographical distance between the cloud service providers, so we refer to the pipelined synchronous replication technology to reduce latency.

Pipelined synchronous replication is defined as blocking on an externally visible event until all writes resulting from the computation [18].Pipelined synchronous replication allows for an overlap of calculation and remote writes when processing a request. However, when an externally visible event is generated, the event must be blocked,but not released to the client until all pending writes are completed.

In essence, through the pipelining computation and remote writes,our approach can reduce the performance penalties associated with speed-of-light delays while ensuring that the same relationship between the client view and the auxiliary view is replicated as synchronous.

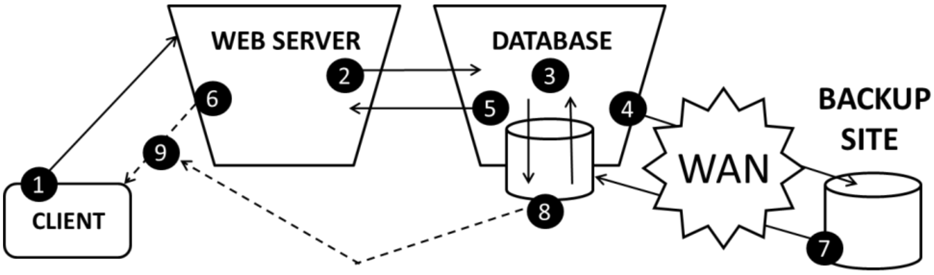

Fig.3 pipelined synchronous replication

Compared the pipelined synchronous replication with existing replication strategy, the illustrative example in Fig.3. For example, there is a client who buys a train ticket from a website, and he submits his credit card information in step 1. As well as we know the client interacts with a frontend web server which may perform some processing before forwarding the request on to a backend database to record her purchase in step 2. In step 3, the database writes the dealing. Because this is the key state of the application in step 4, the disk writes are also replicated to the backup site over the WAN link for retention. The type of replication used determines the behavior of the system. The synchronous replication is that the system continues to process the next replication before a copy is completed (step 5), and responds to the client (step 6), for the acknowledgement from the remote site in step 7. After the local database is written to success, the transfer in step 4 will be executed immediately. Eventually, the Web layer generates a reply to the client in step 6. In the case of the pipelined synchrony, this reply cannot be returned to the customer until the database to write it on the bases of the stick to the remote site. Only after step 7 completed and the remote server has been recognized to write complete can the reply to the customer be released (step 8) and then returned to client's web browser to show her purchase confirmation(step 9).

6. Experiment result and analysis

We use multi-cloud and synchronous replication technology to carry on the simulation experiment. Fig.4 and Fig.5 show the comparison of the average values of normalized COST and RTO. The difference between Fig.4 and Fig.5 is that Fig. 4 used different sets of 500 task, and Fig. 5 used different sets of 1000 tasks. From these two graphs, we can see that the first strategy is the worst, COST and RTO are unacceptable. The second strategy's COST is the lowest, but RTO is the longest. The third strategy's RTO is the shortest, and the COST is relatively higher than others except the first strategy.

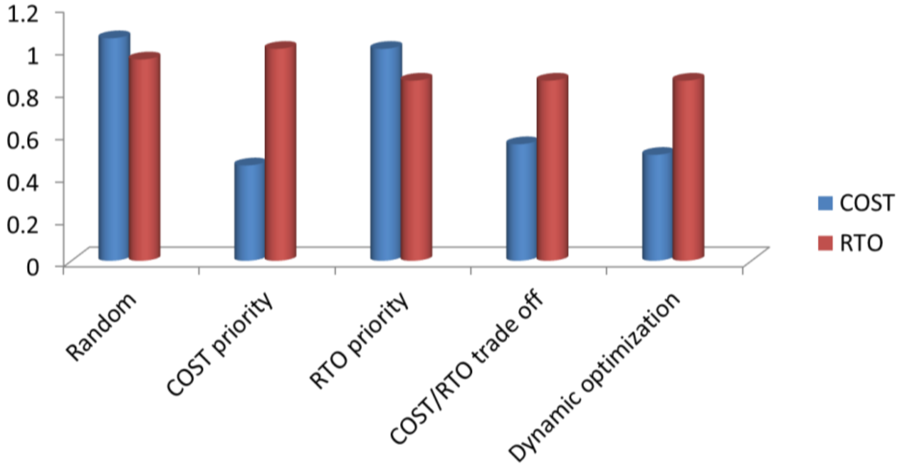

Fig.4 Comparison of the average values of normalized COST and RTO (500 tasks)

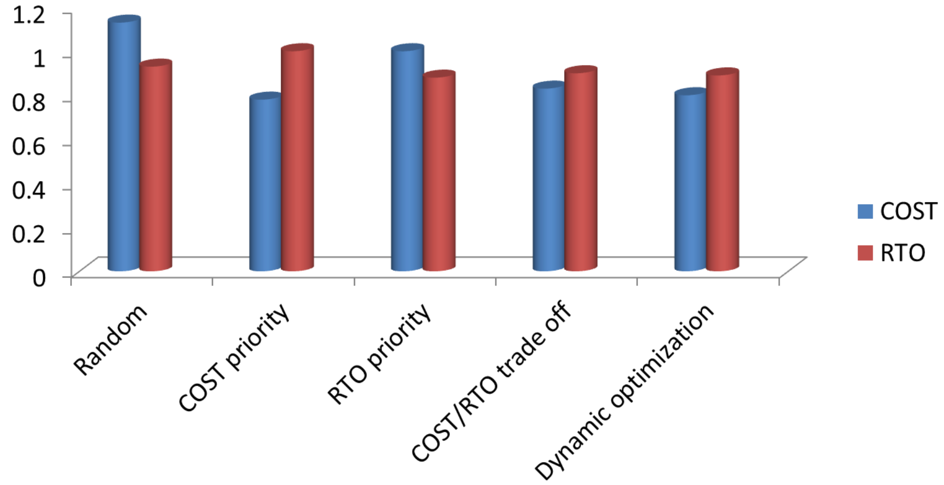

Fig.5 Comparison of the average values of normalized COST and RTO (1000 tasks)

The COST and RTO of the fourth and fifth strategies are relatively balanced. They can achieve relatively approximate COST of the second strategy which is COST preferred and RTO of the third strategy which is RTO preferred. In addition, the fifth strategy can produce better results than the fourth strategy.

Therefore, we can conclude that the multi-cloud model can effectively work with multiple cloud service providers, different cloud computing scheduling strategies have different characteristics. In practice, the first strategy is not appropriate. According to COST and RTO the second strategy and the third strategy maybe suitable for some special scenarios, they have a high demand for COST and RTO, but for other factors are relaxed. Both to achieve the balance of COST and RTO of the fourth and fifth strategies, so they can be applied to the general scene. The fifth strategy has a more optimal result than the fourth strategy, although it is computationally more complex, and it works in most cases, except for those with a large number of cloud providers.

7. Conclusion

Aiming at the problem of disaster recovery, this paper presents a solution of multiple cloud service providers' combination and pipelined synchronous replication. The model enhances data security by backing up and restoring data across multiple clouds, and reducing network latency with WAN replication through pipelined synchronous replication. Customers only need to deal with a cloud service provider, and using a very simple and unified service interface that does not involve an internal process between heterogeneous clouds. This model uses different scheduling strategies to implement low-cost, fast, and secure data disaster recovery solutions that are suitable for different types of disaster recovery.

References

[1] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, The cost of a cloud: Research problems in data center networks, ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 68-73, 2008.

[2] T. Wood, E. Cecchet, K. K. Ramakrishnan, P. Shenoy,J. Van der Merwe, and A. Venkataramani, Disaster recovery as a cloud service: Economic benefits & deployment challenges, in 2nd USENIX Workshop on Hot Topics in Cloud Computing, Boston, USA, 2010.

[3] J. R. Raphael, The 10 worst cloud outages (and what we can learn from them), http: //www. infoworld.com/d/cloudcomputing /the-10- worst-cloud-outages-and-what-we-canlearn-them- 902, 2011.

[4] J. R. Raphael, The worst cloud outages of 2013 (so far), http://www.i nfoworld.com/ slideshow /107783/ theworst-cloud-outages-of-2013-so-far-221831, 2013.

[5] Yu Gu. DR-Cloud Multi-Cloud based disaster recovery service [J].Tsinghua Science and Technology, 2014, 01:13-23.

[6] M. Vrable, S. Savage, and G. M. Voelker, Cumulus: Filesystem backup to the cloud, ACM Transactions on Storage (TOS), vol. 5, no. 4, p. 14, 2009.

[7] A. Bajpai, P. Rana, and S. Maitrey, Remote mirroring: A disaster recovery technique in cloud computing, International Journal of Advance Research in Science and Engineering, vol. 2, no. 8, 2013.

[8] J. Zhang and N. Zhang, cloud computing-based data storage and disaster recovery, in IEEE International Conference on Future Computer Science and Education (ICFCSE), 2011, pp. 629-632.

[9] Y. Tan, H. Jiang, D. Feng, L. Tian, and Z. Yan, CABdedupe: A causality-based deduplication performance booster for cloud backup services, in Parallel & Distributed Processing Symposium (IPDPS), 2011, pp. 1266-1277.

[10] W. Li, Y. Yang, and D. Yuan, A novel cost-effective dynamic data replication strategy for reliability in cloud data centers, in IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), 2011, pp. 496-502.

[11] C. Liu, Y. Gu, L. Sun, B. Yan, and D. Wang, RADMAD: High reliability provision for large-scale deduplication archival storage systems, in Proceedings of the 23rd International Conference on Supercomputing, 2009, pp. 370-379.

[12] T. Wood, H. A. Lagar-Cavilla, K. K. Ramakrishnan, P. Shenoy, and J. Van der Merwe, PipeCloud: Using causality to overcome speed-of-light delays in cloudbased disaster recovery, in Proceedings of the 2nd ACM Symposium on Cloud Computing, 2011, p. 17.

[13] T. Nguyen, A. Cutway, and W. Shi, Differentiated replication strategy in data centers, in Proc. The IFIP International Conference on Network and Parallel Computing, Guangzhou, China, 2010, pp. 277-288.

[14] C. Cachin, R. Haas, and M. Vukolic, Dependable storage in the Intercloud, IBM Research, vol. 3783, pp. 1-6, 2010.

[15] D. Bermbach, M. Klems, S. Tai, and M. Menzel, Metastorage: A federated cloud storage system to manage consistency-latency tradeoffs, in IEEE International Conference on Cloud Computing (CLOUD), 2011, pp. 452-459.

[16] H. Wang, Q. Jing, R. Chen, B. He, Z. Qian, and L. Zhou, Distributed systems meet economics: Pricing in the cloud, in Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, Boston, USA, 2010, p. 6.

[17] J. Kennedy and R. Eberhart, Particle swarm optimization, in Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942-1948.

[18] Timothy Wood. PipeCloud: Using Causality to Overcome Speed-of-Light Delays in Cloud-Based Disaster Recovery[C].Proceedings of the 2nd ACM Symposium on Cloud Computing. ACM, 2011:17.