International Journal of Database Theory and Application
Vol.7, No.5 (2014), pp.1-16
http://dx.doi.org/10.14257/ijdta.2014.7.5.01

Improved Optimization for Data Disaster Recovery System over Low-Bandwidth Networks

Jian Wan[1,2], Xiaolong Hong[1,2] and Jinlin Zhang[1,2,3]

[1]*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China*
[2]*Key Laboratory of Complex System Modeling and Simulation, Ministry of Education*
[3]*Electric Engineering School, Zhejiang University, Hangzhou, China*
[1]*yongjian.ren@infocore.cn,*[2]*hongxiaolong.info@gmail.com,*[3]*jilin.zhang@hdu.edu.cn*

Abstract

Data generated by various fields are increasing exponentially and thus results in challenges for data performances in both scales of diversity and complexity. The problem how to solve the bottlenecks of low-bandwidth networks has been of fatal significance for all kinds of network status. We present a new approach on improved optimization for data disaster recovery system (DDRS) over low-bandwidth networks that not only aims to improve the defects and deficiencies of mainstream DDRS but also helps ensure the reliable network resources for operators to conduct multi-services. A novel bandwidth self-adaptive approach (BSAA) for data packing replication was essentially established to make contribution to the integral performance improvement. A Hidden Markov Model (HMM) for predicting network status was also built to ensure system availability and stability. Experiments showed that the DDRS over low-bandwidth networks named InfoDr can effectively optimize the workload with better performance and better application self-adaptability for multi-services.

Keywords: *Data Disaster Recovery System; Low-bandwidth network; Deduplication and delta compression; Hidden Markov Model*

1. Introduction

Research on integration technology optimization of data disaster recovery system (DDRS) deservedly requires a more rapid performance improvement for not only scientific research but also enterprise business to guarantee the data reliability against disasters, including nature disasters, hardware failures and operation errors. Data generated by various fields are increasing exponentially and thus results in challenges for data performances in both scales of diversity and complexity. Even the working capacity of local replication over high-bandwidth networks is also becoming limiting factors of the integral performance improvement. The problem how to solve the bottlenecks of low-bandwidth networks has been of fatal significance for all kinds of network status. Besides, the research also emerges to ensure the reliable network resources for operators to conduct multi-services.

In this paper, we investigate the issues of defects and deficiencies of existing DDRS. By using incremental backup, deduplication and delta compression, we can optimize storage format of the primary data in primary centers and effectively reduce the amount of data waiting to be replicated. By setting proper DDRS backup interval, we can achieve rational utilization of network resources, by transferring the explosive resource contention time of DDRS to idle time in dawn and thus ensure the reliable network resources for multi-services.

The above technologies can effectively optimize the integral performance improvement, but in the other side, the utilization ratio of transmission link will also increase rapidly with network delay and packet dropout. Therefore, research on integration technology optimization of DDRS over low-bandwidth network aims at the methods that can optimize the performance improvement of both primary center and transmission link.

Our research confirms and highlights the requirements of multi-services over low-bandwidth networks and investigate the issues of defects and deficiencies of existing DDRS [1]. A novel storage format named Time Marker was proposed with the Asynchronous Volume, which plays the most important role for performance compensation. To reduce the network loads, we presented a novel self-adaptive bandwidth control approach for data packing replication based on deduplication and delta compression. The idle network utilization at intermission can be optimized with HMM. By estimating and predicting the network status, we could replicate partial data packages to backup centers in advance so as to avoid the subsequent network congestions.

This paper is organized as follows: Section 2 introduces background and related work of researchers in recent years. Section 3 presents the systematic architecture design of InfoDr and the characteristic of the system — asynchronous volume, which redefines the storage format and data flow direction. In Section 4, a novel bandwidth self-adaptive approach for data packing replication base on DDRS is described, followed by the implementation of deduplication and delta compression. In Section 5, a Hidden Markov Model (HMM) for predicting network status is built for InfoDr, combining primary centers and low-bandwidth links together. In section 6, we design and describe experiments for InfoDr and compare the experiments with mainstream systems. Finally we conclude the paper in Section 7.

2. Related Work

In this paper, we are committed to the integral performance improvement, including tri-relative influence.

(1) Performance improvement in primary center.

Performance improvement in primary centers aims to manager contradictions between the exponential increase of data volume and the bottleneck effect of the transmission links, which have seriously hampered the development of enterprises level storages, especially in the practical applications of data disaster recovery technology [1]. Researchers proposed approaches for optimizing space utilization, improving the storage efficiency, and minimizing network workload, such as scaling [2], deduplication [3, 4], and delta compression [5]. Many other works focus on the storage services, like storage selection, security, storage backup *etc.*, such as [6, 7, 8].

(2) Performance improvement in transmission links

Performance improvement in transmission links aims to improve the integral performance by optimizing transmission efficiency and resolve bottlenecks. Lloyd *et al.*, [9] proposed an approach for effectively intelligent low-bandwidth output stream decreasing from a high-bandwidth input stream, providing model significance for us to reduce network congestion: either decrease traffic flow of thus much generated by special applications [10] or send partial data packages in advance. Based on HMM [11], we can infer a model of our study for predicting network status from available bandwidth seen by a sequence of probes sent from one end host to another end host.

(3) Performance improvement in backup center.

Performance improvement in backup centers is usually measured by two key parameters: recovery point objective (RPO) and recovery time objective (RTO) [1, 14]. Engineers protect the critical data from disasters mainly by some recovery technologies in backup center after periodical (daily or weekly) backups [12] and snapshots [12, 13].

3. Architecture

Numerous papers have explored architectures for DDRS and find that there are repeated components, which have been leveraged in systems to promote the integral performance improvement [15, 16, 17]. The essence of Data Disaster Recovery Technology behaves as three forms of data flow: "sequence", "branch", and "repeat", from one component to another. The research of our paper considers InfoDr as a system consists of three main components: primary center, backup center and transmission links between them.

As shown in Figure 1, not only the components but also the interior data stream in InfoDr shows similarities and differences from other mainstream DDRS. As we know, clients are usually regarded as the starting point of data flow while recovery volumes serve as ending point. Logic volumes mapping from heterogeneous storage pools of primary centers via iSCSI or FC channels are mounted for business requirements.

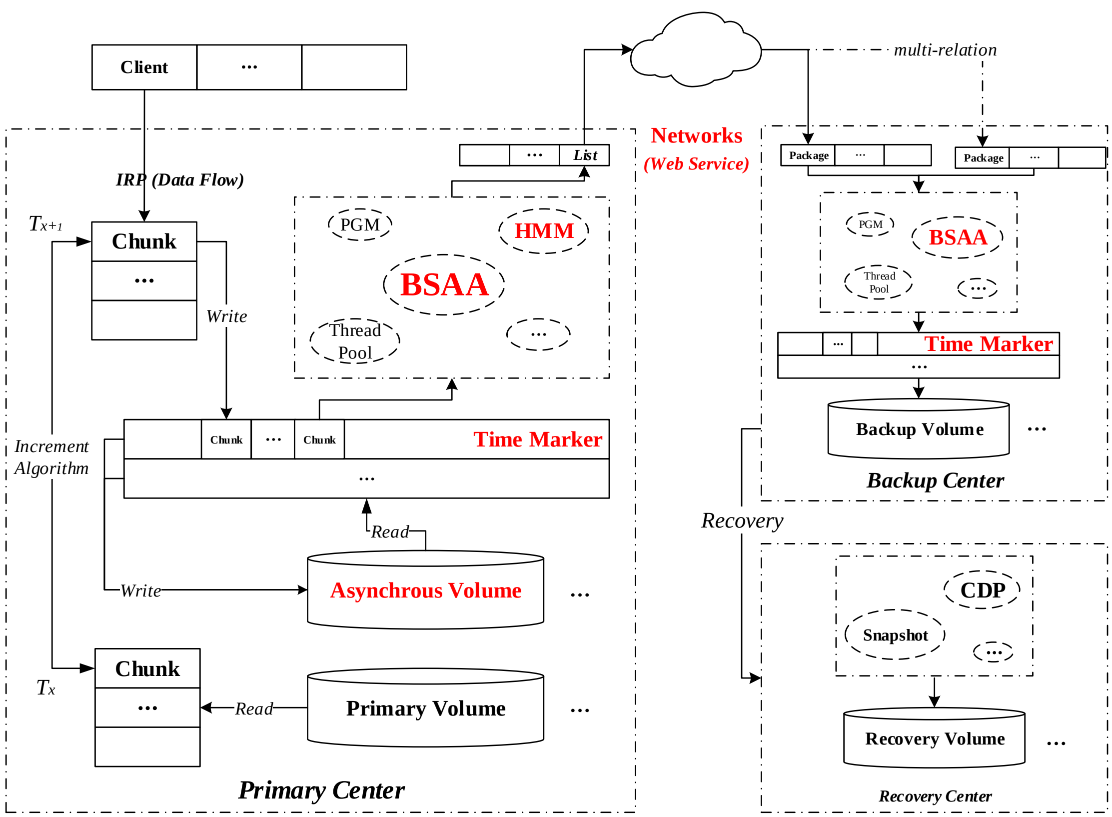

Figure 1. The Architecture of InfoDr

Primary data always comes from client computers connected with primary centers over iSCSI or FC channels as the starting point of data flow. After receiving and processing, InfoDr finally stores primary data flow in primary volumes and asynchronous volumes in some special formats. During the above week, service logic model determines that primary data must flow entirely through Data communication layer, Data processing layer, Data

Management layer and Data storage layer, realizing forms of “sequence” and “branch”. Further, once conditions are met, backups are immediately going to transfer data packages from primary centers to backup centers in a reverse direction, realizing another format of “repeat”. Different from the above description, antipodal cycles are actually adopted for recovery technologies in similar DDRS including InfoDr. Thus it is necessary to analysis the data cycles to help us understand the essence of InfoDr architecture.

For asynchronous replication, each write IO is sent to primary volume and intermediate volume named Asynchronous Volume. The data inside the IRP has its own unit of measurement named Chunk, and likewise, Asynchronous Volume has its own format, building an in-memory representation named Time Marker.

|--GH+TMDL--|--TMDC--|--TMDC--|--TMDC--|--TMDC------

1) GH: Global Header, the global control information of asynchronous volumes.

2) TMDL: Descriptions of Time Marker.

3) TMDC: Incremental data from Data Management layer are stored in chunks here, detailed as follows:

|--TMDC Header--|--IET--|------Data------|

a) IET: IO Entery Table.

b) Data: |--Chunk Data--|--Chunk Data--|-------

TMDC is set to be variable length, depending on data change between two time points.

For 1MB IET size, if each IET entry (need 8 bytes for original offset and offset on storage) represents 128KB (a chunk), a total of 1MB/8B=128K chunks (128KB*128K=16GB) can be stored. For first-time write into a chunk, initialize the chunk by primary data, and then just simply over-write. Replication will send the whole chunk over to remote site. To reduce the search time for looping over IET to find the right IO enrty, split IET into 8 sub IETs, with each sub IET 64KB size. Each IET will cover 64KB/8B=8K IOs. With sub IET, updating IET IOs onto volume shoule be 512B for each IET change, rather than whole IET change.

Asynchronous incremental mode, also known as Time Marker, further reduces the volume of primary data and becomes the best strategy for backups over low-bandwidth networks, balancing RTO and RPO [1]. Compared with caches, whose volatility may lead to immeasurable losses, Asynchronous Volume aims to ensure strict data consistency for the critical data. Different from mainstream DDRS, we combine Asynchronous Volume with a novel bandwidth self-adaptive approach for data packing replication and settle down to optimize efficiencies of both space and time.

4. A Novel Bandwidth Self-adaptive Approach

The calibration framework described in §3 directs that the forms of data flow: “sequence”, “branch”, and “repeat” show essential changes of primary data in the most literal sense: different paths in the whole InfoDr system, along which data throughputs increase or decrease as time goes by. With the development of information technology characterized with width of space, which can be optimized by the four layers of InfoDr with three forms, we settle down to optimize the depth of space. We proposed a novel bandwidth self-adaptive approach (BSAA) for data packing replication based on DDRS, which is essentially established to make contribution to the integral performance improvement.

4.1. Deduplication and Delta Compression

Besides spatial locality, there also exist a mass of duplicate data blocks compared with the similar blocks separated by time intervals. Backups request for appropriate sizes of data packages replicated to corresponding remote sites over low-bandwidth networks. We proposed a novel bandwidth self-adaptive approach (BSAA) for data packing replication based on DDRS, which is essentially established to make contribution to the integral performance improvement.

The process of BSAA is shown in Figure 2. In InfoDr, Asynchronous Volume receives write IO and stores incremental data between time points. Each driver layer provides service to the application layer above by interface style. BSAA reads and managers Time Marker to the final packages waiting replication. Before backups, BSAA settles down to optimize data space and data security, which are mainly concentrated on data collision in the period of fingerprint computing.

We try to balance time loss caused by read access, deduplication and delta compression by applying the thread pool and pre-reading from Time Marker in Asynchronous Volume. The detailed introduction of BSAA is given as follows.

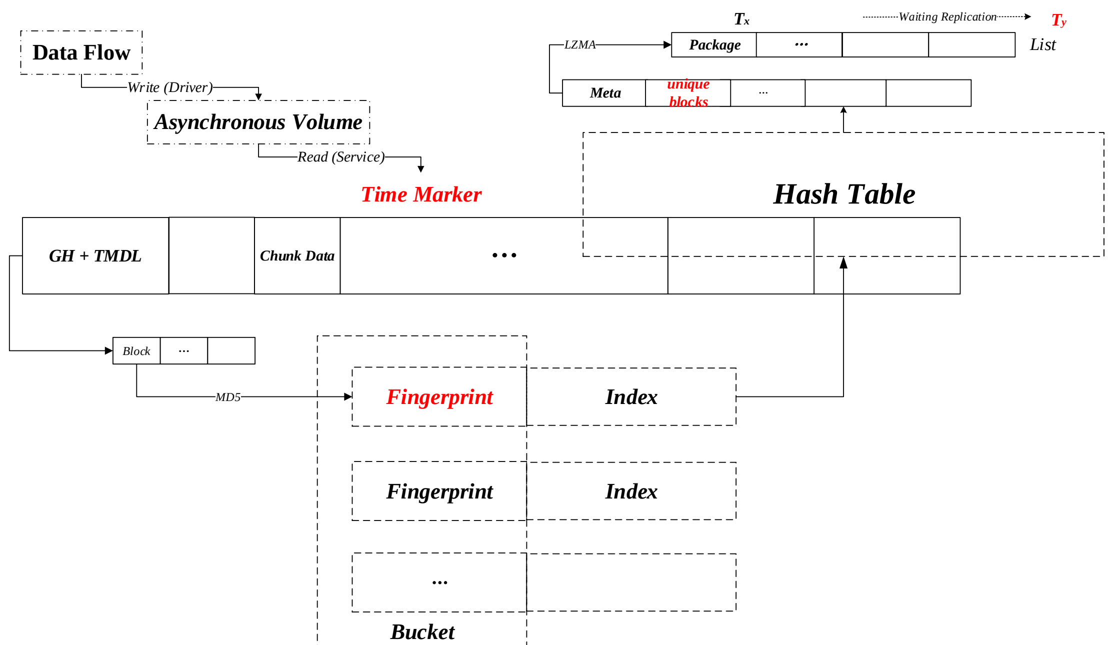

Figure 2. The Process of BSAA

Step 1 Data read from Time Marker will be split into a group of blocks, the quantity of which mainly depends on the output of the model in section 4. Each and all blocks need to calculate fingerprint for minutiae detection and matching based on the Hash algorithm.

Step 2 The indexes from the results of the Hash algorithm should be distinguished and applied by the bucket for the consistency policy to find unique blocks.

Step 3 Meta information records the indexes and offsets of the unique blocks and invariably ignores other duplicate blocks. In this way, we finally get a kind of logical representation, which is consist of metadata and unique blocks, corresponding to Time Marker.

Step 4 The LZMA algorithm is applied for the file-level compression and packages are finally inserted into lists for replication.

Step 5 Finish BSAA process, which can also serve as an inverse analysis method in backup centers, and write the final data to backup volumes.We measure deduplication and delta compression by deduplication ratio, compression ratio and performance. Performance of deduplication depends on the implementation technology while deduplication ratio depends on characteristic of primary data and applications generating data flow. Primary data provided in Time Marker is mainly dominated by duplicate data blocks over time intervals, and we manager to set the particle size of deduplication from 4K to 24K.

InfoDr is mainly designed for DDRS over low-bandwidth networks. The biggest mark of InfoDr is the bandwidth self-adaptive approach, which can dynamically adjust package sizes by means of predicting network state in advance. In the case of heavy network load, we always consider background traffics and packages with small size priorities to avoid network congestions. Only when network resources are sufficient for multi-services, big packages can be replicated from primary centers. In our study, we also make great efforts on fallback mechanism to abandon contentions in backup intervals, giving way to higher-priority traffics. Therefore, the role of asynchronous volumes become the protagonist for Data processing layer to pack primary data waiting deduplication and delta compression. We also try to balance time loss caused by read access, deduplication and delta compression using thread pool and pre-read from Time Markers in asynchronous volumes, which are regarded as buffers here.

4.2. Data Collision and Consistency Policy

Data security issues in §4 are mainly concentrated on data collision in the period of fingerprints computing. In our research, the md5 algorithm is used to get fingerprints corresponding to data blocks. Even though the collision of hash keys is considered as small probability events, the probability cannot be ignored because that it is bound to result in great economic loss with even any mistake. In InfoDr, we make efforts on methods based on md5 to optimize the data security of deduplication.

The issues imply that certain assumptions, while valid in simulation, may lead to unexpected behavior when the collisions happen in InfoDr. As we know, fingerprints result from the function – get_md5(block), corresponding to unique data blocks theoretically. If the collisions happen, fingerprints certainly can't serve as good criterions for deduplication. Some similar researches [18, 19, 20, 21], have presented methods to find real collisions quickly on a typical PC. Consistency policy of InfoDr requests two or more of the hash algorithm to guarantee the data security.

Actually, we can use the Link Method to solve collisions of hash keys. In our study, we need to store data blocks with the same hash keys in a special bucket. To distinguish unique blocks, we compare blocks bit by bit to another with the same hash key in bucket. After 2 times or more operations, we can finally get the right meta information.

As part of our framework, we offer a set of issues and steps to implement our method to solve collisions:

Step1. Firstly, assumptions about abstract models for the behavior of collisions are the foundation for inference methods used to interpret active measurements. Combined with data chunks (namely blocks in deduplication) in asynchronous volumes, we need to compute hash keys using the md5 algorithm, and record them before inserting into a hash table;

Step2. Secondly, we need to search the hash table defined in the first step to make sure the uniqueness of hash keys, which always represent unique blocks in deduplication files;

Step3. If the result in the second step shows no collisions have yet happened, we consider these hash keys as new fingerprints for unique blocks, otherwise, the process must continue in step 4;

Step4. There are already records same to results in step 2 in the hash table, we need more operations as follows:

Traversal bucket, which stores logical block numbers and offsets, and compare blocks bit by bit to another with the same hash key;

If the operations of A still show the same results, we ignore the current blocks because they are proved to be duplicate blocks;

If the operations of A show different results, we insert new records into the bucket and regard the current blocks as unique blocks.

InfoDr is mainly designed for DDRS over low-bandwidth networks. From the optimization problems, we also should consider background traffics and packages with small size priorities to avoid network congestions. In our study, we also make great efforts on fallback mechanism to abandon contentions in backup intervals, giving way to higher-priority traffics. Therefore, BSAA with Asynchronous Volume becomes the protagonist to optimize the workload with better performance and better application self-adaptability at both intermission and non-intermission.

5. A Hidden Markov Model Over Low-bandwidth Networks

In this paper, we built a Hidden Markov Model (HMM) [22] over low-bandwidth networks for InfoDr to evaluate network status series. Based on time series analysis, InfoDr gathers the observations by fixed time intervals from Probing Gap Models (PGM) [23], attempting to record and predict end-to-end available bandwidths of low-bandwidth networks before backups.

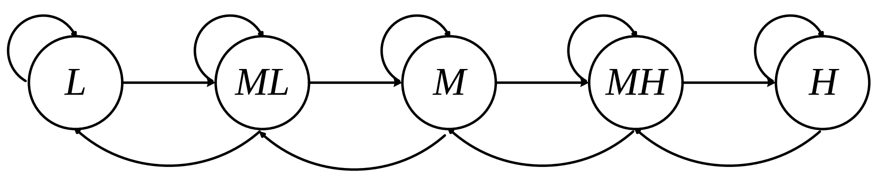

Figure 3. The State Transition Direction

An HMM is characterized by the individual states as $S = \{S_1, S_2, \dots, S_N\}$, each of which corresponding to a distinct hidden state and $V = \{V_1, V_2, \dots, V_M\}$, representing the observation states. Given appropriate parameters of the triple $\lambda = (\pi, A, B)$ for HMM, the state transition direction is shown in Figure 3. Each state variable represents a certain degree of bandwidth utilization range, corresponding to network load conditions. The sample values L to H are always in intervals from 0% to 100% as follows:

1) L (Low Load): 0% to 20%.
2) ML (Low and Medium Load): 20% to 40%.
3) M (Medium Load): 40% to 60%.
4) MH (Medium and High Load): 60% to 80%.
5) H (High Load): 80% to 100%.

(1) The Baum-Welch Algorithm.

The Learning Problem can be solved by the Baum-Welch algorithm [24], also known as Forward-Backward algorithm. Based on the probability estimates and expectations computed, the algorithm concentrates to find the local maximum and as such its usefulness depends upon the HMM being trained.

The basic process of the algorithm is as follows:

Step 1 Create an initial model H_0 with initial values for $\lambda = (\pi, A, B)$.

Step 2 Initial condition: A set of observation sequence $O = \{O_1, O_2, \dots, O_T\}$.

Step 3 Consider one such hidden state sequence $Q = (q_1, q_2, \dots, q_T)$ a forward variable defined as

$$\alpha_t(i) = P\left(O_1, O_2, \dots, O_t, q_t = S_i \middle| \lambda\right) \tag{1}$$

A backward variable defined as

$$\beta_t(i) = P\left(O_{t+1}, O_{t+2}, \dots, O_T \middle| q_t = S_i, \lambda\right) \tag{2}$$

Step 4 An estimated version $A' = (a'_{i,j})$ can be calculated by the auxiliary quantity with expected number of transitions from hidden state S_i to S_j stateand expected number of transitions from state S_i.

$$a'_{i,j} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{3}$$

$$\xi_t(i, j) = P\left(q_t = S_i, q_{t+1} = S_j \middle| O, \lambda\right) \tag{4}$$

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j) \tag{5}$$

Step 5 A set of reasonable re-estimation formulas for $B' = (b'_j(k))$ calculated with expected number of times in state j and observing symbol v_k and expected number of times in state j is as follows.

$$b'_j(k) = \frac{\sum_{\substack{t=1 \\ s.t. O_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{6}$$

Step 6 Re-estimate $\lambda = (\pi, A, B)$, learn and train a new model H from H_0.

Step 7 Determine the convergence, if not, go to step 2.

The Baum-Welch algorithm can be described as follows:

1 begin initialize: select arbitrary model parameters for $\lambda = (\pi, A, B)$

2 do $\lambda' := \lambda, S' := S$

3 Calculate $\alpha_t(i)$ using the Forward algorithm.

Calculate $\beta_t(i)$ using the Backward algorithm.
Calculate $A = (a'_{i,j})$ using (3).
Calculate $B' = \left(b'_j(k)\right)$ using (6).
while convergence criterion achieved
return $\lambda' = (\pi, A, B)$
end

(2) The Viterbi Algorithm.

To find the best hidden state sequence given observations and model, the Viterbi algorithm [25] considers status of transition probabilities with the maximal probabilities already derived for the preceding step. Formally, $\sigma_t(i)$ is the highest probability along a single path, at time t.

$$\sigma_t(i) = \max_{q_0, q_1, \dots, q_{t-1}} P\left(O_1, O_2, \dots, O_t, q_0, q_1, \dots, q_{t-1}, q_t = i \middle| \lambda\right) \tag{7}$$

Initialization:

$$\sigma_1(i) = \pi_i b_i(O_1), 1 \le \mathrm{i} \le \mathrm{N} \tag{8}$$

$$\chi_1(i) = 0 \left(\chi : an\ array\ keeps\ track\ of\ argument\ maximized\right) \tag{9}$$

Recursion:

$$\sigma_t(j) = \max_{1 \le i \le N} \left[\sigma_{t-1}(i)\, a_{i,j}\right] b_j(O_t), 2 \le \mathrm{t} \le \mathrm{T}, 1 \le j \le N \tag{10}$$

$$\chi_t(j) = \arg\max_{1 \le i \le N} \left[\sigma_{t-1}(i)\, a_{i,j}\right], 2 \le t \le T, 1 \le j \le N \tag{11}$$

Termination:

$$P^* = \max_{1 \le i \le N} \left[\sigma_T(i)\right] \tag{12}$$

$$q_T^* = \arg\max_{1 \le i \le N} \left[\sigma_T(i)\right] \tag{13}$$

Path (state sequence) backtracking:

$$q_t^* = \chi_{t+1}\left(q_{t+1}^*\right), t = T-1, T-2, \dots, 1 \tag{14}$$

The Viterbi algorithm can be described as follows:

```
begin initialize path χ_t
for state s_j do
   χ_1[i,1] := π_i b_i(O_1)
   χ_2[i,1] := 0
end for
for time step t := 1 to n do
  for state s_j := 1 to n do
    Calculate T_1[j, t] using (10)
    Calculate T_2[j, t] using (11)
  end for
end for
```

$qt_T := \arg\max_k \left(T_1[k,T] \right)$
$\chi_T := s_{qt_T}$
for i := T to 2 do
  $qt_{i-1} := T_2[qt_i, i]$
  $\chi_{i-1} := s_{qt_{i-1}}$
end for
return path
end

(3) The Probe Gap Model.

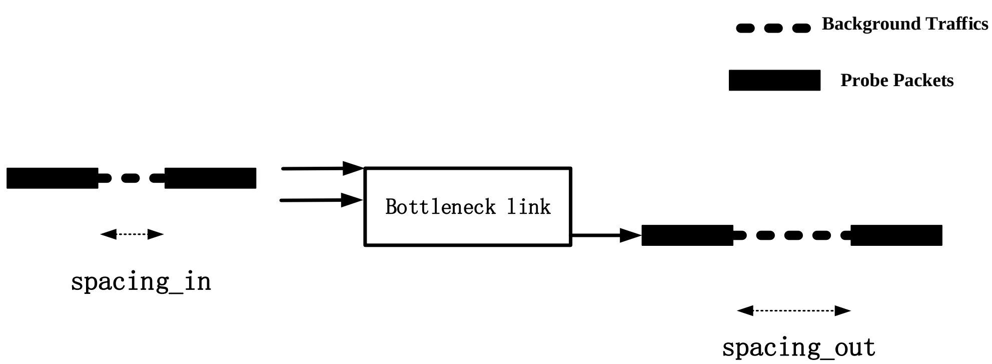

Figure 4. The Probe Gap Model (PGM)

The PGM in InfoDr is shown in Figure 4 and described as follows:

Step1. Initialization: input the real capacity of our networks and initialize related parameters.

Step2. Resource allocation: thread pools allocate and dispatch network resources, which finally come into being UDP packets queues and wait for disposing in intervals of spacing_in.

Step3. Session management: receivers receive the probe packets and maintain relationships between spacing_in and corresponding space_out, from which we can get results and feedback information to senders.

Step4. Output: background traffics (BT) can be calculated by formula as follows:

$$BT = \frac{spacing\_out - spacing\_in}{spacing\_in} * C \tag{15}$$

Where C represents known capacity values of link paths.

Available bandwidth (AB) series in InfoDr is described as follows:

$$AB = C\left(1 - \frac{spacing\_out - spacing\_in}{spacing\_in}\right) \tag{16}$$

6. Experimental Evaluation

The main objectives of the experiments include

1) Improved optimization of BSAA for data packing replication based on deduplication and delta compression before backups.

2) Improved optimization of the HMM over low-bandwidth networks for multi-services in InfoDr.

6.1. Experiment Environment

To measure environment effectiveness, we built transmission networks on a city level, placing data centers from labs to another with a few tens of kilometers distance. Cloud hosts provided by different cloud hosting vendors are correct and effective tactics to simulate extreme distance networks. The capacities of the IP networks between primary centers and backup centers are equals to 10M (WAN), while clients are directly connected to primary centers via Gigabit networks (LAN).

The experiment environment for InfoDr consists of three components: primary center, backup center and transmission links between them. The machine configurations are shown in Table 1. Primary volumes are mounted to clients for business requirements, mapping from primary centers via iSCSI.

Table 1. Experiment Environment

Experiment environment	Primary Center	Backup Center	Client
Machine Model	IBM System x3650 M3		Dell Inc. OptiPlex 380
CPU	8 cores Intel® Xeon® CPU E5620@2.4GHz		Pentium® Dual-Core CPU E5300@2.60GHz
Memory	4G		2G
Networks	10M(WAN)		1000M(LAN)
Storage	SAS300G+SAS300G	SAS300G	
Operating System	Microsoft® Windows® Server 2003, Enterprise Edition		
Version	5.2.3790 Service Pack 2 Build 3790		

6.2. Performance Evaluation

Because of the characteristics of InfoDr in real environments, replicating packages over low-bandwidth networks should make best use of the transmission links without congestions caused by intense competitions for network resources. Even though the performance improvements of transmission links are optimized by InfoDr in primary centers, the special network observations can be observed in usability evaluation. So, we determine to test the effects of deduplication and delta compression in performance evaluation separately.

Performance evaluation includes:

1) Data Set. A number of different versions of virtual machine images based on Linux and Windows as data set.

2) Experimental Results. Figure 5 shows the efficiency of BSAA. Notice that the storage spaces of the virtual machine images accumulated decrease in primary centers step by step after deduplication and delta compression. In comparison with the original sizes, BSAA helps reduce the amount of data by a large margin. From table 1 we can also find the deduplication rate and the compression rate, both of them are impacted by data sets, the higher the better. Therefore, InfoDr system can in some ways be proved performing efficiently with BSAA for low-bandwidth networks.

In the experiment for usability evaluation based on description of modeling assumption, backup intermission starts from 8:00 to 24:00. Packages will immediately be replicated at non-intermission, no matter the sizes or amounts. Time validity and data reliability are the key factors to meet the request of RTO and RPO, because of which large amounts of network bandwidths may be occupied for backups at intermission. Background traffics referring to random variations exist from 8:00 to 20:00, similar to real working hours. Workload pressure simulation of the low-bandwidth networks becomes bottlenecks for backups in InfoDr, reaching about 4GB/h from traffic generators based on web-service.

Table 2. Observation Results for BSAA

Test Data *(KB)*	Original	Deduplication	Compression	Deduplication Rate	Compressing Rate
CentOS6.3	5242880	970458	268949	0.8149	0.9487
Ubuntu13.04	5095670	3997304	948948	0.2155	0.7138
Windows7	20971520	5376335	1864017	0.7436	0.9111
Windows2008	7542400	5615994	1691807	0.2554	0.7757
Windows Xp	2254016	1679915	685699	0.2547	0.6958

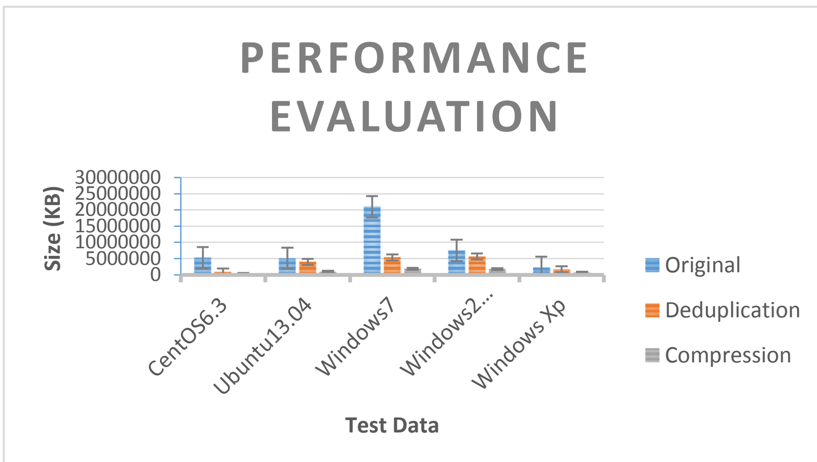

Figure 5. Performance Evaluation of BSAA

As we can see, Figure 6 to Figure 9 present different experimental results under different conditions. Once the background traffics (BT) with multi-services comprise at higher proportion, more network congestion problems arise much more easily. From the curves in Figure 6 and Figure 8, explosive resource contentions deeply impact on the whole system at intermission, mainly because of the characteristic and workload pressure of low-bandwidth networks. Multi-service systems request enough available bandwidths for communication applications on dynamic random time. In addition, it is also limited to RPO and data volume. In comparison with Figure 7 and Figure 9, we can learn that InfoDr with HMM can effectively help ensure the reliable network resources. There still exists a percentage of available bandwidths for multi-services. To some degree, we can also find that InfoDr helps save time for retransmission caused by network congestions.

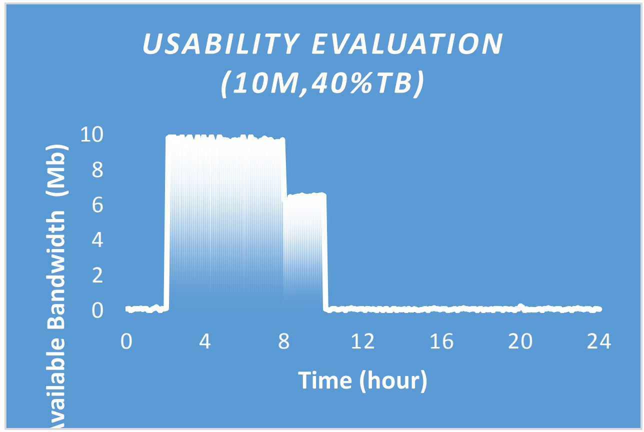

Figure 6. The Curves of Usability Evaluation for InfoDr (10M,40%TB)

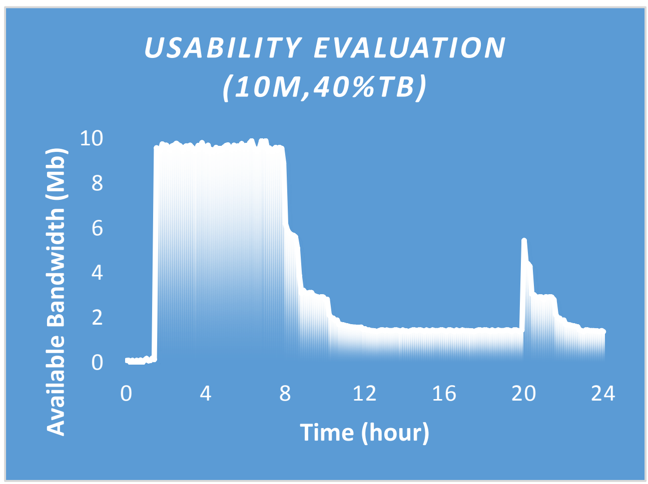

Figure 7. The Curves of Usability Evaluation for InfoDr (10M,40%TB,BSAA)

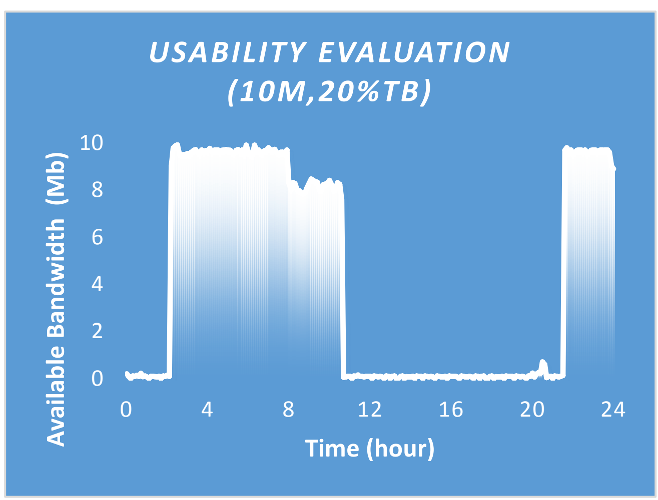

Figure 8. The Curves of Usability Evaluation for InfoDr (10M,20%TB,)

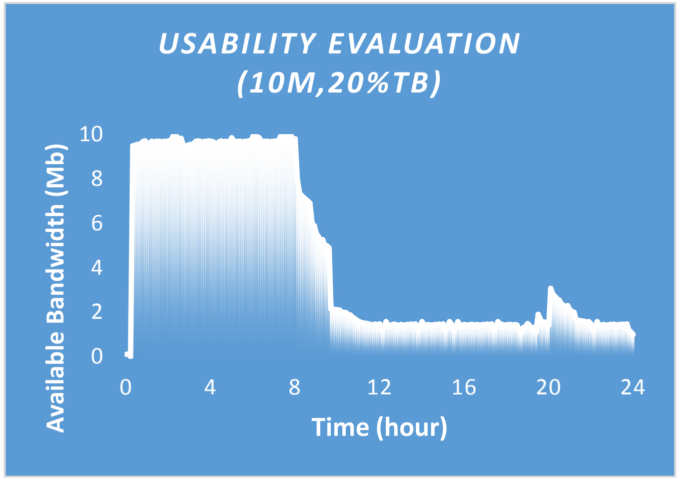

Figure 9. The Curves of Usability Evaluation for InfoDr (10M,20%TB,BSAA)

7. Conclusion

In this paper, we proposed a novel DDRS called InfoDr, which can be applied for core data protection over low-bandwidth networks. BSAA with Asynchronous Volume for data packing replication based on deduplication and delta compression is also presented to improve the performances and workloads for primary centers and transmission links. Then an HMM for predicting network status trade offing is given in detail, helping make use of idle available bandwidths at intermission for multi-services. Finally, we provided detailed theoretical modeling and analysis, and our experimental results confirmed that InfoDr can provide significant improved optimization for DDRS over low-bandwidth networks and is feasible for using in real world cloud storage systems.

Acknowledgement

This paper is granted by National Natural Science Foundation of China under Grant(No. 61202094), Zhejiang Provincial Natural Science Foundation(No.LY13F020045, LY13F020047), China Postdoctoral Science Foundation(No. 2013M541780) and The National Key Technology R&D Program under Grant(No.2012BAH24B04).

References

[1] IBM International Technical Support Organization San Jose Center, "International Technical Support Organization Planning for IBM Remote Copy", **(1995)**.

[2] C. Dubnicki, G. Leszek, H. Lukasz and M. Kaczmarczyk *et al*., "A scalable secondary storage, In Proceedings of the 7th conference on File and Storage Technologies", **(2009)**, pp. 197-210.

[3] M. Lillibridge, K. Eshghi, D. Bhagwat and V. Deolalikar, *et al*., "In Proceedings of the 7th Conference on File and Storage Technologies", **(2009)**, pp. 111-123.

[4] B. Zhu, K. Li and H. Patterson, "In Proceedings of the 6th Conference on File and Storage Technologies", **(2008)** February, pp. 269-282.

[5] P.Shilane, M. Huang, G. Wallace and W. Hsu, "WAN-optimized replication of backup datasets using stream-informed delta compression", "In Proceedings of the Tenth USENIX Conference on File and Storage Technologies", **(2012)**, pp. 1-26.

[6] A. Ruiz-Alvarez and M. Humphrey, "An automated approach to cloud storage service selection", "In Proceedings of the 2nd international workshop on Scientific cloud computing", **(2011)**, pp. 39-48.

[7] S. Kamara and K. Lauter, "Cryptographic cloud storage", "In Proceedings of Financial Cryptography and Data Security", **(2010)**, pp. 136-149.

[8] F. Douglis, D. Bhardwaj, H. Qian and Philip Shilane, "Content-aware load balancing for distributed backup", "In Proceedings of USENIX LISA", **(2011)**.
[9] L. Mize, R. H. Klenke and J. M. McCollum, "Snapshot capture from live high definition video stream for transmission over low-bandwidth data link", "In Proceedings of IEEE SoutheastCon", **(2010)**, pp. 344-347.
[10] J. Du, H. Yu and W. Zheng, "MassStore: A low bandwidth, high De-duplication efficiency network backup system", "In Proceedings of International Conference on Systems and Informatics", **(2012)**, pp. 886-890.
[11] S. Bhupinder, N. Kapur and P. Kaur, "Speech Recognition with Hidden Markov Model: A Review", "International Journal of Advanced Research in Computer Science and Software Engineering 2.3", **(2012)**.
[12] M. Rock and P. Poresky, "Shorten your backup windows storage", "In Proceedings of Special Issue on Managing the information that drives the enterprise", **(2005)**, pp. 28-34.
[13] G. Duzy, "Snaps to Apps", "In Proceedings of Storage, special issue on managing the information that drives the enterprise", **(2005)**, pp. 46-52.
[14] X. Weijun, J. Ren and Q. Yang, "A case for continuous data protection at block level in disk array storages", "Parallel and Distributed Systems, IEEE Transactions on 20.6", **(2009)**.
[15] K. Keeton, C. Santos, D. Beyer and J. Chase *et.al.*, "Designing for disasters", "In Proceedings of the 3rd USENIX Conference on File and Storage Technologies", **(2004)**, pp. 59-62.
[16] R. Cegiela, "Selecting technology for disaster recovery", In Proceedings of the International Conference on Dependability of Computer Systems, **(2006)**, pp. 160-167.
[17] V. Hristidis, V. Hristidis, T. Li, T. Li and Y. Deng, "Survey of data management and analysis in disaster situations",. In Proceedings of the Journal of Systems and Software, **(2010)**, pp. 1701-1714.
[18] X. Wang, Y. L. Yin and H. Yu, "Finding Collisions in the Full SHA-1", Cryptology–CRYPTO, **(2005)**, pp. 17-36.
[19] X. Y. Wang, D. G. Feng and X. Y. Yu, "The Collision Attack on Hash Function" HAVAL-128. In Chinese, Science in China, Series E, vol. 35, no. 4, **(2005)** April, pp. 405-416.
[20] X. Y. Wang and H. B. Yu, "How to Break MD5 and Other Hash Functions", Advances in Cryptology–Eurocrypt'05, Springer-Verlag, **(2005)** May, pp. 19-35.
[21] X. Y. Wang. X. J. Lai, D. G. Feng, H. Chen and X. Y. Yu, "Cryptanalysis for Hash Functions MD4 and RIPEMD", Advances in Cryptology–Eurocrypt'05, Springer-Verlag, **(2005)** May, pp.1-18.
[22] R, Lawrence, "A tutorial on hidden Markov models and selected applications in speech recognition", Proceedings of the IEEE 77.2, **(1989)**, pp. 257-286.
[23] H, Ningning and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques. Selected Areas in Communications", IEEE Journal on 21.6, **(2003)**, pp. 879-894.
[24] W. L. Rawrence, "Hidden Markov models and the Baum-Welch algorithm", IEEE Information Theory Society Newsletter 53.4 ,**(2003)**, pp. 10-13.
[25] F. G. D. Jr., "The viterbi algorithm. Proceedings of the IEEE 61.3", **(1973)**, pp. 268-278.

Authors

Yongjian Ren received the PhD degree in Computing Application Technology from Florida Atlantic University, in 1998. He is currently a professor in software engineering in Hangzhou Dianzi University, China. His research interests include Cloud Computing and Mass Storage.

Xiaolong Hong received the Bachelor Degree of Computer Science and Technology in Hangzhou Dianzi University, Zhejiang, China, in 2011. He is now studying the Master of Computer System Structure in Hangzhou Dianzi University, China. His research interest is Network Storage Technology.

Jilin Zhang received the PhD degree in Computer Application Technology from University of Science & Technology Beijing, Beijing, China, in 2009. He is currently a lecture in software engineering in Hangzhou Dianzi University, China. His research interests include High Performance Computing and Cloud Computing.

