Yo-Sung Ho
Hyoung Joong Kim (Eds.)

LNCS 3768

Advances in Multimedia Information Processing – PCM 2005

6th Pacific-Rim Conference on Multimedia
Jeju Island, Korea, November 2005
Proceedings, Part II

2 Part II

Springer

Table of Contents – II

An Effective Failure Recovery Mechanism with Pipeline Computing in Clustered-Based VOD Servers

Dongmahn Seo, Joahyoung Lee, Dongkook Kim,
Yoon Kim, and Inbum Jung

Department of Computer Information & Telecommunication Engineering,
Kangwon National University, Hyoja-Dong, Chunchon,
Kangwon, 200-701, Korea
{dmseo, jhlee, dkkim, yooni, ibjung}@snslab.kangwon.ac.kr

Abstract. For the actual Video-On-Demand (VOD) service environment, we implement a cluster-based VOD server composed of general PCs and adopt the parallel processing for MPEG movies. For the implemented VOD server, a video block recovery mechanism is designed on the RAID-3 and the RAID-4 algorithms. However, without considering the architecture of cluster-based VOD server, the application of these basic RAID techniques causes the performance bottleneck of the internal network for recovery. To solve these problems, the new failure recovery mechanism based on the pipeline computing concept is proposed. The proposed method distributes the network traffics invoked by recovery operations and utilizes the available CPU computing power of cluster nodes.

1 Introduction

Recent advanced computer and communication technologies have provide economically feasible multimedia services such as VOD, digital library and Education-On-Demand (EOD). Among them, the VOD service is the most prominent multimedia application. It provides online clients with the video data of streaming level by guaranteeing the Quality of Service (QoS) metric [1].

In contrary to traditional file servers, VOD servers are subject to real-time constraints while storing, retrieving and delivering the movie data into the network. Since the ceasing and jittering streaming videos are unmeaningful for VOD clients, the streaming media should be supplied within the QoS metric to each client. To support the QoS, servers must be able to continuously deliver video data at a constant interval to VOD clients. And also, even in the failure of server components, the streaming service should be re-continued within the human acceptable Mean Time To Repair (MTTR) value [2,3].

A cluster server architecture has been exploited in the areas of Internet Web, database, game and VOD server [4]. It has an advantage of the ratio of performance to cost and is easily extended from the general PC equipment. The cluster server architecture usually consists of a front-end node and multiple backend

Y.-S. Ho and H.J. Kim (Eds.): PCM 2005, Part II, LNCS 3768, pp. 12–23, 2005.

nodes. Since the video data are distributed into several backend nodes, the performance scalability including the storage devices could be achieved accordingly as the number of backend nodes increased. However, even if the cluster server can be scaled by just adding new backend nodes, the probability of the failure of nodes also increases in proportion to the number of backend nodes.

The fault of nodes causes not only the stop of all streaming service but also the loss of the position information of current playing movies. Since the VOD server has to guarantee QoS streams to all clients even in the failure of nodes, the recover mechanisms are necessary for dealing with a realistic VOD service. In this paper, the recovery mechanisms in cluster-based VOD servers are studied to support QoS streams while a backend node is in failure state.

To study the failure events during the actual VOD service, we implement the cluster-based VOD server composed of general PCs and adopts parallel processing for MPEG media to support large scale clients. From the implemented VOD server, it is evaluated that a basic recovery system is composed of the advantages of RAID-3 and RAID-4 algorithms. From experiments, it is found that the basic recovery system causes the performance bottleneck on the input network of the recovery node that consume a few computing resource of CPU. To solve these issues, the new failure recovery system based on pipeline computing is proposed over all survived backend nodes. The proposed system distributes the network traffics across all backend nodes. All survived backend nodes are participated in the recovery operations so that the proposed method provides the improved performance of cluster-based VOD servers as well as the unceasing streaming service even in the failure state of a backend node.

The rest of this paper is organized as follows. Sect. 2 explains the implemented cluster-based VOD server and the management of video blocks in the cluster architecture. Sect. 3 suggests the basic recovery system mixed the advantages of RAID-3 and RAID-4 levels and a new recovery mechanism based on pipeline computing to utilize the resources of backend nodes. In Sect. 4, performances of two recovery systems are measured and discussed. Sect. 5 concludes the paper.

2 Implemented Cluster-Based VOD Server

For large scale VOD services, we implement a cluster-based VOD server called as Video On Demand on Clustering Architecture (VODCA). The VODCA consists of a front-end node named as Head-end Server (HS) and several backend nodes known as Media Management Server (MMS). Throughout the internal network path between a HS node and MMS nodes, they exchange the working states and internal commands each other. The HS node not only receives clients' requests but also manages MMS nodes to support QoS. When new MPEG movies are enrolled, they are split by HS and distributed into each MMS node. To perform these administrative functions, the HS consists of striping module, monitoring module, service_control module and main_daemon module. The MMS nodes transmit their stored movie fragments to clients under the supervision of the HS node. Each MMS node sends the current working status to the HS

node periodically. This message operates as a heartbeat protocol between MMS nodes and the HS node. Each MMS node consists of media_management module, media_service module, resource_management module and main_daemon module.

To apply parallel processing for MPEG movies, movie files are striped according to the defined granularity policy. To exploit MPEG media characteristics in parallel processing, a GOP size is used as a striping unit, since each GOP has approximately equal running time in MPEG streams. The MPEG movies are split into GOPs and distributed into each node with their sequence number and size.

3 Proposed Recovery Systems

From the implemented VOD server of previous section, a video block recovery mechanism is designed on the RAID-3 and the RAID-4 algorithms. However, without considering the architecture of cluster-based VOD server, the application of these basic RAID techniques causes the performance bottleneck of the internal network for recovery. To solve these problems, the new failure recovery mechanism based on the pipeline computing concept is proposed.

3.1 Recovery System on Basic RAID Mechanisms

Fig. 1 shows the architecture of the recovery system based on basic RAID-3 and RAID-4 mechanisms. We denote this recovery model as Recover System based on Basic RAID Mechanisms (RS-BRM). This system is implemented on VODCA sever described in Sect. 2. As shown in Fig. 1, two network paths exist: one is used for connecting between the MMS nodes and the VOD clients, and

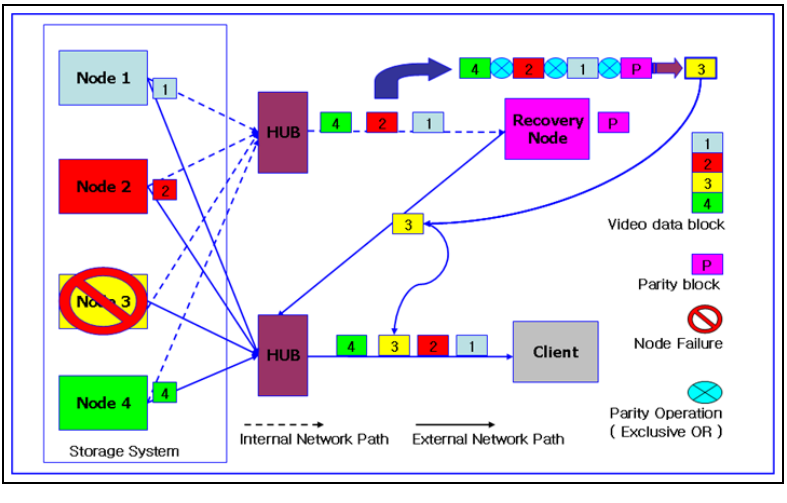

Fig. 1. Architecture and video block flows in RS-BRM

the other is an internal network path installed between all MMS nodes and a recovery node. When a MMS node fails, the video blocks should be transferred to the recovery node. These blocks are transferred on the isolated internal network path. Therefore, the external network path fully focuses on the QoS streams for clients without interference events.

When all MMS nodes are working normally, all MMS nodes transmit their stored video blocks to clients directly through the external network path. On the other hand, when a MMS node fails, the survived MMS nodes send the video blocks to both the clients and the recovery node. Using the video blocks received from the MMS nodes and the parity blocks stored in its own disks, the recovery node regenerates the failed video blocks. Since both MMS nodes and the recovery node use their internal network path for recovery operations, the external network bandwidth can support QoS streams to the VOD clients. For the cluster-based VOD server architecture, we introduce the RAID-4 level to improve the data retrieving performance, and apply the RAID-3 level to the cases of data transferring and recovering operations. Since the RAID-3 level can support smaller stripping units, all video blocks are gradually aggregated in the recovery node so that the abrupt memory shortage could be avoided. By tailoring the advantage of the RAID-3 and RAID-4 mechanisms, this mixed approach improves the performance of recovery system by utilizing the characteristics of individual hardware components. For example, as shown in Fig. 1, when the MMS 3 node fails, the recovery node regenerates the video block 3 by calculating the exclusive OR operation with the received video blocks 1, 2, 4 and its own parity block. Since the regenerated video block 3 is sent to the corresponding client via the external network path, the streaming media service is unceased even in the failure state.

3.2 Recovery System Based on Pipeline Computing

The performance of RS-BRM suffers from the bottleneck of input network on the recovery node. It has been restricted by the number of MMS nodes. To address this problem, the new recovery system based on the pipeline computing is proposed. It is denoted as Recovery System based on Pipeline Computing Mechanism (RS-PCM). The proposed method distributes the network traffics for recovery operations into all survived MMS nodes and utilizes the available CPU computing capacity of MMS nodes.

The exclusive OR operations for video blocks are a major role of parity based RAID algorithms. To rebuild the video blocks stored in the failure MMS node, sequential several exclusive OR stages are necessary. For each stage, the two blocks are needed to compute exclusive OR operation at a time. Based on this characteristic, the stages are distributed into MMS nodes so that the network saturation in the recovery node is solved. In addition, the CPU computing power of MMS nodes can be utilized without impairing the QoS streams.

Fig. 2 shows the architecture of RS-PCM and the flow of video blocks in the VODCA server. The basic algorithm for data recovery is based on the RAID-4,

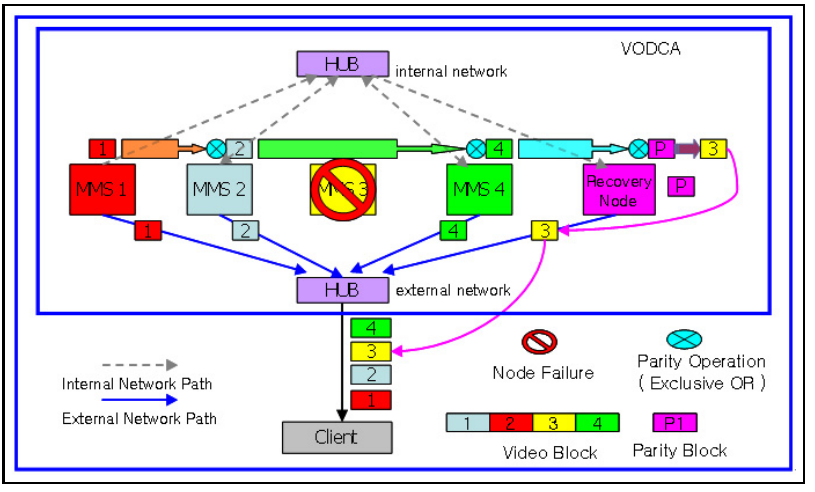

Fig. 2. Architecture and the flow of video blocks in RS-PCM

3 algorithms. As shown in Fig. 2, the RS-PCM distributes the network traffics for recovery processes, and spreads the exclusive OR operations over all MMS nodes.

When a MMS node fails, survived MMS nodes do not send their video blocks to the recovery node directly but transmit the original video block or their own exclusive OR result block to their neighbor MMS node. Each MMS node performs its own fraction of exclusive OR operation with both the video block retrieved from its local disk and the block received from its neighbor MMS node. The blocks received from its neighbor MMS node may be an original video block stored in the disk or the result of exclusive OR operation processed on the neighbor MMS node. The results are sent to the neighbor MMS node successively such as the pipeline process in the instruction level [5].

Finally, the recovery node performs the last exclusive OR operation with its parity block and the aggregate result of all MMS nodes so that the video block of the failure MMS node is rebuilt. After that, the regenerated video block is transmitted to the client through the external network path. For example, as shown in the Fig. 2, when the MMS node 3 fails, the MMS 1 node sends the video block 1 to the MMS 2 node. The MMS 2 node performs the exclusive OR operations with both the video block 1 and the block 2. After that, the result is sent to the MMS 4 node to perform the exclusive OR operation with the video block 4. Finally, after the exclusive OR operations for all survived video blocks are finished, the result is sent to the recovery node. The recovery node regenerates the video block 3 throughout the exclusive OR operation with the parity block.

Fig. 3 shows the recovery operations according to the pipeline concept of the RS-PCM. As a CPU unit executes a step of pipeline to perform an operation in a cycle, each MMS node executes a step of recovery operation using its idle CPU in order to recover a failed movie block in a cycle [5]. This parallel processing for recovering the failed blocks makes a good performance in proposed RS-PCM.

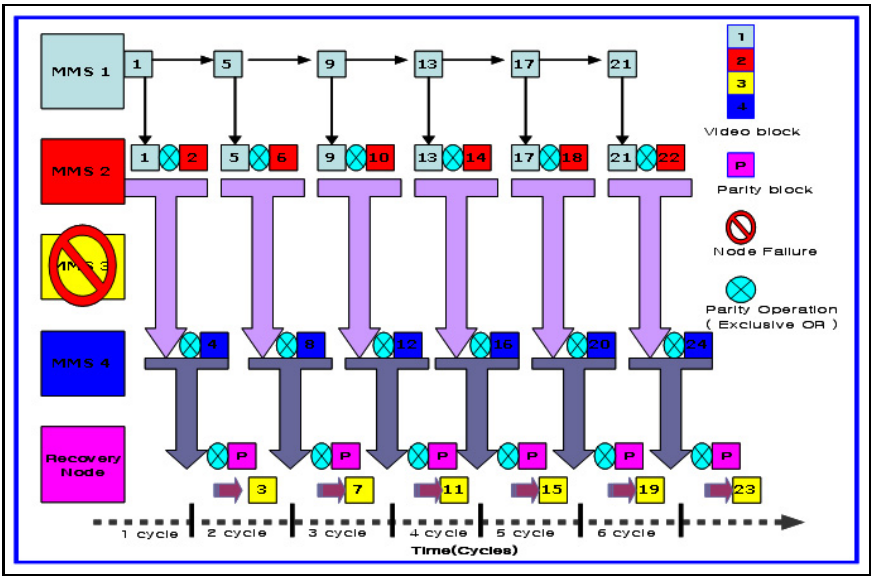

Fig. 3. Recovery steps based on pipeline concept in RS-PCM

As shown in Fig. 3, the failed MMS 3 node has video block 3, 7, 11, 15, 19, 23. These blocks are regenerated in the recovery node every cycle according to the pipeline computing.

4 Performances of Proposed Recovery Systems

The VODCA server for experiments consists of a HS node, 4 MMS nodes and a recovery node. Each node operates on the Linux operating system. The MMS nodes, HS node and clients are connected via a 100 Mbps ethernet switch. All MMS nodes and the recovery node are also connected via the internal network path constructed by a 100 Mbps ethernet switch. The yardstick program is used to measure the performance of the implemented cluster-based VOD server [6]. The yardstick program consists of the virtual load generator and the virtual client daemon. The virtual load generator is located in the HS node and generates client requests based on the Poisson distribution with $\lambda = 0.25$ [7,8]. These requests are sent to each MMS nodes. After that, all MMS nodes concurrently begin streaming media services for satisfying the client's demand.

4.1 Performances of RS-BRM

Fig. 4 shows the amounts of output network traffics transmitted from a MMS node to all clients in the RS-BRM. The results are the averages of the network traffics of each MMS node. As shown in Fig 4, the load generator generates six loads individually. The VODCA server guarantees 1.5 Mbps transmitting rates

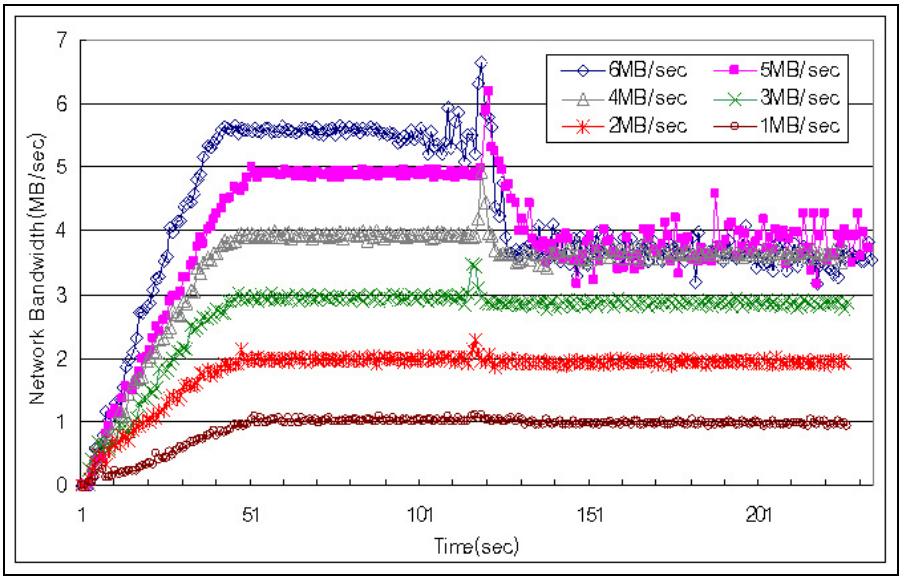

Fig. 4. Output network traffic from a MMS node to all clients

for each QoS stream. The output network traffic of 6 MB/s means that 4 MMS nodes provide 128 clients with QoS stream.

The failure of a MMS node takes place at 120 second of time line. Under the 1 MB/s, 2 MB/s and 3 MB/s traffics, the variations of network traffics are minimal after the failing event. However, in the 6 MB/s, 5 MB/s and 4 MB/s traffics, the fluctuations of network traffics continuously appear in the time line from the 120 second position. In particular, the severely reduced network traffics are incurred in the 6 MB/s and 5 MB/s cases. The reason is that the MMS node can not send the video blocks fully to clients only if the recovery node can not receive more video blocks due to its input network bottleneck. Under the 5MB/s and the 6 MB/s loads, the network traffics from 3 MMS nodes to the recovery node reach to 15 MB/s and 18 MB/s respectively. Since the input network capacity of the recovery node is limited to 12 MB/s, the recovery node suffers from the bottleneck phenomenon of input network path.

From the experiments, we also observe that the CPU usage of MMS nodes is minimal. Since the MMS nodes simply perform the retrieving and transmitting of their own video blocks, the average CPU utilization is measured below 10 %.

Fig. 5 shows the reading times of one GOP in the client side while the streaming service is in progress. Although the failure time of a MMS node is the 120 second of the time line, the fluctuation of reading times in the client side appears in the 156 second of the time line. Since the network delay exists from the VOD server to clients, the time delay takes place due to the buffering mechanism in the client side.

As shown in Fig. 5, when all MMS nodes work normally, the average reading time is about 0.65 seconds and it keeps steady state. However, after a MMS node

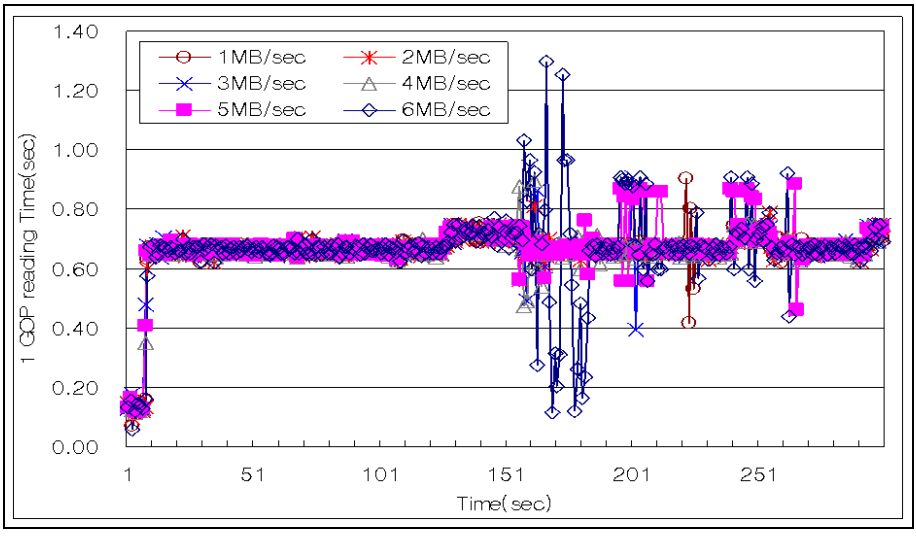

Fig. 5. GOP reading time in the client side under RS-BRM

fails, the reading times vary. These variations are due to the packet data loss, the initial setup time of the recovery node and the data congestion phenomenon of the recovery node. In particular, the fluctuation rates are high at the 5 MB/s and 6 MB/s load. In these work loads, the unsteady state of the reading times comes out between the 156 seconds and the 266 seconds. The difference between the maximum reading time and the minimum reading time is about 1.18 second. After the fluctuation period pass through, the recovery node works normally and the reading times converge into the 0.65 second level again. The MTTR value is 110 seconds [2,3]. It can be regarded as impatient period to VOD clients.

4.2 Performances of RS-PCM

Fig. 6 shows the network traffics in a MMS node and the recovery node when the 12 MB/s network traffic is loaded in the RS-PCM. The failure takes place at the 120 second of the time line. As shown in Fig 6, after the failure occurs, the network traffics from a MMS node to clients decrease from 12 MB/s rates to 9 MB/s rates. The reason is that the video blocks transmitted from the neighbor MMS node occupy the main memory of the MMS node. If many clients are serviced, the video blocks from its neighbor MMS node take a great part of memory. The shortage of memory causes memory swapping overheads. However, the output traffic to clients in a MMS node is over twofold compared with the RS-BRM. In Fig. 4, the RS-BRM shows the maximum 4 MB/s traffics due to the input network bottleneck of the recovery node. In the RS-BRM, even though the 12 MB/s load is generated, the output traffics to clients is 8MB/s rates. This experimental result proves that the RS-PCM provides more clients with the QoS streams than the RS-BSM.

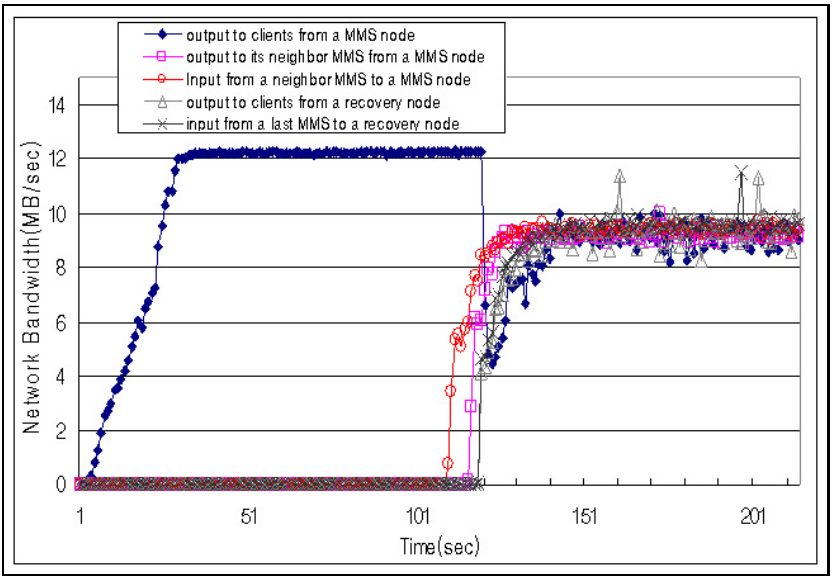

Fig. 6. Network traffic of a MMS node and a recovery node under 12 MB/s load

The square legend mark of this figure represents the amount of output traffic toward the neighbor MMS node. In the RS-PCM, if the current MMS node is not the last MMS node, it transmits its own video blocks or the result blocks of exclusive OR operation to its neighbor MMS. From the circle legend mark, it is found that the amount of input traffics from the neighbor MMS node is almost equal to that of its own output traffics. The amount of input traffics from the last MMS node reaches the 9 MB/s rates so that the recovery node also can rebuild the video blocks as much as 9 MB/s rates. After that, the recovery node transmits them to clients. According to the triangle legend mark of this figure, the output traffics of rebuilt blocks in the recovery node get to the 9 MB/s rates.

When compared with the RS-BRM, the RS-PCM has better performance as much as double streams in the same working environment. The memory swapping problem in the RS-PCM could be simply solved by adding memory units. From additional experiments, after extending the memory capacity, it is confirmed that the output network traffics from a MMS node reached the maximum 12 MB/s. However, even if the amount of memory units increase, the internal network bottleneck of the RS-BRM can not be avoided. Since the RS-PCM utilizes the available CPU resources of MMS nodes and all MMS nodes are participated in the total recovery procedures, it provides the improved performance of cluster-based VOD servers as well as the unceasing streaming services under the failure of a MMS node.

Fig. 7 represents the reading times of one GOP in the client side. The experiments are performed on between the 7 MB/s and 12 MB/s loads. The RS-PCM can support these network traffic loads. The failure of a MMS node takes place in the 120 seconds position. Since there is the network delay between the server

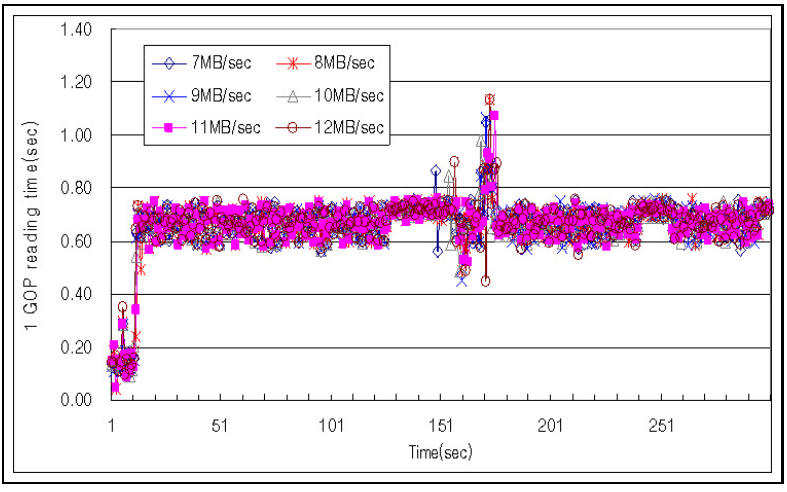

Fig. 7. GOP reading time in the clients side under RS-PCM

and clients, the fluctuations of reading time in the client side begin at the 148 seconds and end at the 176 seconds. After the agitation state, the reading times promptly converge into the steady state with 0.65 seconds levels. The fluctuation period is 28 seconds.

When compared with the RS-BRM, the period of the fluctuation is very short. Since the recovery operations are distributed into all MMS nodes, the recovery node can transmit the rebuilt video blocks in the relatively short time. As shown in the Fig. 5, after a MMS failure happen, the RS-BRM need 110 seconds to return to the steady state. The fluctuation period of RS-PCM is 4 times shorter than the RS-BRM.

Furthermore, as shown in Fig. 7, the difference between the maximum reading time and the minimum time is 0.68 seconds. This result is the half of the difference in the RS-BRM. In the RS-PCM, both the period of fluctuations and the amplitudes of vibration are shorter than those of the RS-BRM. From these results, the RS-PCM results in much better MTTR value than the RS-BRM [2,3].

5 Conclusions

To study the recovery system in the actual VOD service, we implemented the cluster-based VOD servers composed of general PCs and the internal network path. From the implemented VOD server, the RS-BRM was designed with the advantage of RAID-4 in disk retrieving speed and the advantage of RAID-3 in effective memory usage. However, in the RS-BRM, it was found that the input network path of a recovery node was easily saturated with the video blocks transmitted from the survived MMS nodes.

To address these issues, the RS-PCM based on the pipeline computing was proposed over the MMS nodes and a recovery node. In the RS-PCM, the recovery

node generated a rebuilt video block and sent it to the client just one time for each cycle. This mechanism is similar to the pipeline process of instructions. The RS-PCM made an efficient use of the available CPU resource of MMS nodes since all survived MMS nodes were participated in the recovery procedures to rebuild the impaired video blocks. Based on this pipeline computing, the RS-PCM distributed not only the computation load for exclusive OR operation but also the network traffics across all MMS nodes. From the experiments, we observed the network traffics across all MMS nodes. Even in the failure state of a MMS node, the RS-PCM showed the improved performance by providing at least twice unceasing QoS streams compared to the RS-BRM.

One of the important characteristics in VOD service is that the streaming media with ceasing, jittering and out of ordered frames are not meaningful. This requirement is deserved even in the partial failure state of VOD server. To satisfy this characteristic, after a failure takes place, the fluctuation period should be short. In the GOP reading times in the client side, the RS-PCM showed the 4 times shorter fluctuation period than the RS-BRM. Due to the relatively short fluctuation period, the streaming media service quickly converged into the normal steady state. As a result, the RS-PCM resulted in much better MTTR value than RS-BSM.

In future work, we plan to evaluate the effectiveness of RS-PCM in the failure of a portion of disks in a MMS node. In that case, since the impaired MMS node can send its heart beat, it is difficult to detect the abnormal MMS node from the point of view of the HS node. And also, there are several issues that the MMS node with the partly failed disks will be participated in the recovery operation. We will investigate the method to detect the partial disk failure from MMS nodes and apply the RS-PCM to the imperfect cluster-based VOD server.

References

1. Dinkar Sitaram, Asit Dan: Multimedia Servers: Applications, Environments, and Design. Morgan Kaufmann Publishers, 2000
2. Armando Fox, David Patterson: Approaches to Recovery Oriented Computing. IEEE Internet Computing, Vol. 9, No. 2, pp.14–16, 2005
3. Dong Tang, Ji Zhu, Roy Andrada: Automatic Generation of Availability Models in RAScard. IEEE International Conference of Dependable Systems and Networks, pp.488–494, June 2002
4. http://www.ieeetfcc.org/
5. David A. Patterson and John L. Hennessy: Computer Organization & Design. Morgan Kaufmann, pp.392–490, 1998
6. Brian K. Schmidt, Monica S. Lam, J. Duane Northcutt: The interactive performance of SLIM: a stateless, thin-client architecture. ACM SOSP'99, pp.31–47, 1999
7. W.C. Feng and M. Lie: Critical Bandwidth Allocation Techniques for Stored Video Delivery Across Best-Effort Networks. 20th International Conference on Distributed Computing Systems, April, pp.201–207, 2000
8. Jung-Min Choi, Seung-Won Lee, Ki-Dong Chung: A Muticast Delivery Scheme for VCR Operations in a Large VOD System. 8th IEEE International Conference on Parallel and Distributed Systems, pp.555–561, June 2001

9. http://www.mpeg.org/
10. T. Chang, S. Shim, and D. Du: The Designs of RAID with XOR Engines on Disks for Mass Storage Systems. IEEE Mass Storage Conference, pp.181–186, March 1998
11. Prashant J. Shenoy, Harrick M. Vin: Failure recovery algorithms for multimedia servers. Multimedia Systems, 8, Springer-Verlag, pp.1–19, 2000
12. Leana Golubchik, Richard R. Muntz, Cheng-Fu Chou, Steven Berson: Design of Fault-Tolerant Large-Scale VOD Servers: With Emphasis on High-Performance and Low-Cost. IEEE Transactions on PARALLEL AND DISTRIBUTED SYSTEMS, Vol.12, No.4, pp.363–386, 2001