01 Jan 2006

Design and Implementation of a Storage Management Method for Content Distribution

Chang-Soo Kim
Missouri University of Science and Technology, ckim@mst.edu

Yu-Hyeon Bak

Sang-Min Woo

Won-Jae Lee

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/1530

Recommended Citation

C. Kim et al., "Design and Implementation of a Storage Management Method for Content Distribution," *Proceedings of the 8th International Conference Advanced Communication Technology, 2006*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2006.
The definitive version is available at https://doi.org/10.1109/ICACT.2006.206173

Design and Implementation of a Storage Management Method for Content Distribution

Chang Soo Kim, Yu Hyun Bak, Sang Min Woo, Won Jae Lee, Ok Gee Min, Hag Young Kim
ETRI (Electronics and Telecommunications Research Institute)
{cskim7, bakyh, smwoo, russell, ogmin, h0kim}@etri.re.kr

***Abstract* — The SMART system is a special purpose server developed by ETRI and designed for efficient streaming services over high speed networks. The SMART server has one or more special purpose NS (Network-Storage) card. The NS card has several disks that store multimedia contents. However all of the multimedia contents to be serviced cannot be stored at the server.**

In this paper, we will describe the storage management mechanism in design and implementation aspects. With this storage management mechanism, the SMART server can provide effectiveness in managing storage and distributing some contents from a source station to streaming service servers.

***Keywords* — SMART, Multimedia Streaming, Content Distribution, Storage Management**

1. Introduction

As computers and high speed networks have been popularized, the Internet becomes a main means of transferring information over the world in our lives. As the communication networks to home will be upgraded to 10 ~ 100Mbps bandwidth level in the near future, high quality multimedia services such as internet broadcast, remote medical services and internet video services will be more generalized [1]. For example, a FTTH (Fibre To The Home) village is planned to be constructed in Korea. Those high quality multimedia services require a system that can handle the multimedia streaming service over network fast.

In general, networking operation has high cost in computer system in that it requires several data copies from one place to another and processing steps whenever data passes from one protocol layer to another [2], [3]. In order to provide efficient multimedia streaming service over the network, we need to minimize the cost of networking operation.

Also, service providers should deliver multimedia contents to end users with stability and efficiency over network in real time [8], [9], [10], [11]. As service requests are increased, the traffic of a back-bone network increases, and as a result, the construction cost for the back-bone network increases. This problem should also be resolved [8],[9],[10],[11],[12],[13].

In order to support multimedia streaming services with high speed network infrastructure efficiently, it is needed to process high-cost network operations with high performance. Also, efficient management of infrastructure and distribution of service requests are required to reduce back-bone traffic and server loads respectively. Thus, we developed an efficient system including hardware and software components. We call this system as SMART (Server for Multimedia Applications in Residence community) server.

Figure 1 shows the service diagram of the system. Our system hierarchically distinguishes the services into global and local services that are processed separately to use network bandwidth efficiently. A global server provides overall multimedia contents and located at Internet Data Center (IDC), e-government and so on.

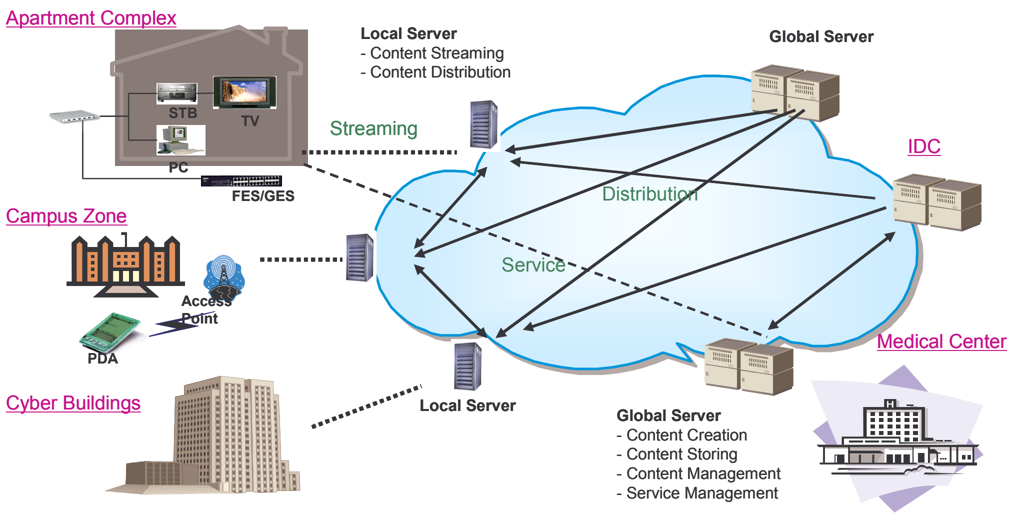

Figure 1. Service Diagram of the System

While a local server provides multimedia streaming services for local communities such as apartments, companies, universities that are based on local networks. It transparently communicates with a global server [12],[13].

As first phase, before overall system including the global server is designed, the SMART server focused on local server only. Therefore, in this paper, we use the term "SMART" for a local server. In this case, the role of the global server as contents provider must be simulated. For this role, we will introduce CSN (Contents Server Node) node in our system infrastructure.

The SMART server system contains special purpose NS cards for high performance networking and storage access operations, a special file system for NS card, multimedia streaming software and content distribution component.

In this paper, we focused on distribution of contents between CSN node and local servers. Specially, we propose a mechanism that manages the storage of SMART server correctly and efficiently. With this mechanism, the SMART server can provide effectiveness in managing storage and distributing contents from global server to local servers.

The remainder of this paper is organized as follows. In chapter 2, we describe overall SMART system itself, and in chapter 3, we explains the Content Distribution module's role and mechanism. In chapter 4, we propose a mechanism to support efficient storage management in terms of design and implementation. Finally, we show the results of performance evaluation and conclusion with brief summary in chapter 5 and 6.

2. SMART System

The SMART server is planed to be located at regional community such as campus, hospital, apartment, and so on. It provides multimedia streaming services at regional community. The SMART server consists of a conventional mother board with 2 processors and one or more special purpose PCI cards called as NS (Network Storage) card. The number of NS cards is ranged from one to four. The NS card aims to provide high-performance networking operation and storage access operation. It combines NIC (Network Interface Card), PCI memory with 512MB capacity and SCSI controller.

Figure 2 shows the architecture of NS card. The NS card can transfer multimedia contents from disks to network directly in a zero-copy manner without any intervention of host processor. Each NS card has its own disks that store multimedia contents and cannot access disks on other NS cards.

Various Software components are also included in the SMART system in order to support efficient streaming service at regional community. The overall software architecture of the SMART system is shown in figure 3.

The SMART system is operating on Linux OS, and has a special file system component to support NS card, large capacity of storage and efficient access to very large files. The special file system has two options. One is CC (Content Container) which is user-level application implementing file system functionalities.

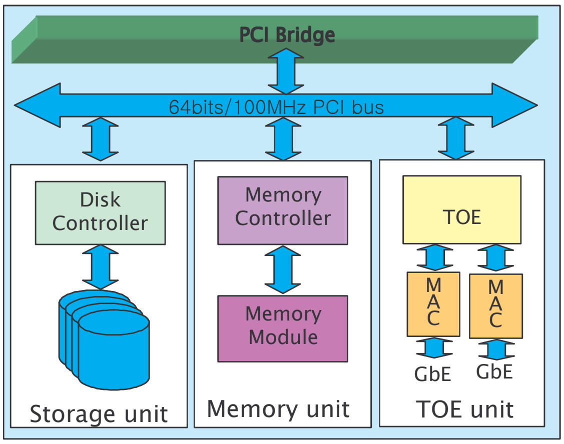

Figure 2. Architecture of NS card

The other is EXT3NS file system that extends the ext3 file system to accommodate features of the NS card.

The CMM (Content Metadata Manager) maintains system configuration data, metadata for contents to be serviced and other service related data. This can be replaced with any database-like component.

The Content Streaming component is a middleware for transmitting some multimedia contents through CC or EXT3NS interface in order to provide streaming service to regional community at high speed bandwidth.

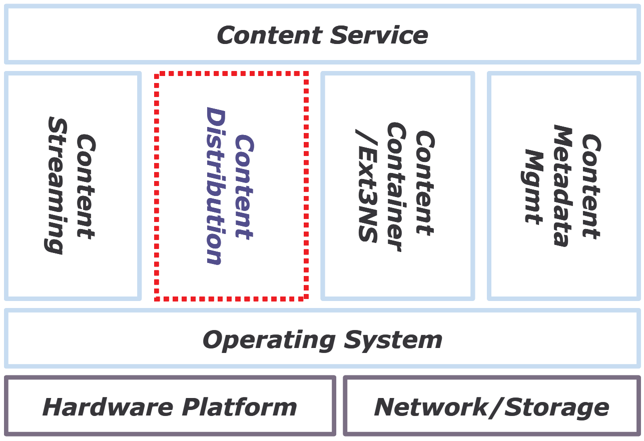

Figure 3. S/W Architecture of the SMART server

The Content Distribution component is mainly responsible to distribute required contents from a global server or CSN node to the SMART server. Other features of the Content Distribution component will be described in Section 3.

A SMART server can support up to 1000 clients and up to 200 concurrent clients with 20mbps high-quality MPEG streaming service.

3. Content Distribution

It is important to store the right content in the right place at the right time to satisfy client's requests. Also, when a requested content is not locally available, it should be obtained from the CSN node as fast and reliably as possible. In order to support these requirements, the server should provide an efficient content distribution mechanism. The more the server caches streaming media, the more the server can provide services to users. Our server system also adopts prefix caching mechanism for making more services to be available on the spot.

The content distribution software provides following features for versatile transfer and distribution methods, and content usage monitoring.

- Content transfer from CSN to local server
- Preloading and prefix caching
- Dynamic loading
- Content purge
- Storage management
- Content usage monitoring
- Clustering support
- Content placement for efficient and balanced resource consumption

The software architecture of the Content Distribution component is shown in figure 4.

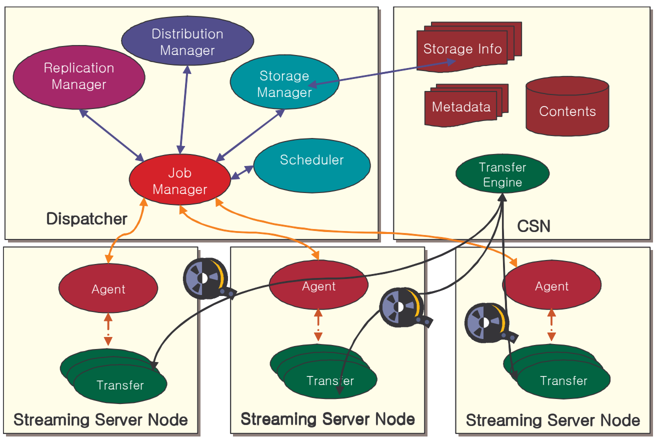

Figure 4. Software Architecture of Content Distribution

The Job Manager which runs at a dispatcher node in the cluster, manages jobs submitted from user or other components of the system, and distributes the jobs into appropriate servers. Jobs are managed in on-demand or scheduled fashion. Also, the Job Manager handles content placement based on various information including server load, storage status, and so on.

The Agent is run at every server node in the cluster, and executes jobs distributed from the Job Manager. Most of those jobs are content transfer requests. A transfer job is executed by invoking the Transfer client. The Transfer client connects to the Transfer Engine at the CSN node and pulls the requested content from the CSN. The Agent also notifies the status of streaming server node and NS cards in order to support efficient content placement and request dispatching. If the status of a component is not updated in given time period, the system considers it as failed component.

The Scheduler supports the single system image in cluster environment by dispatching all incoming service requests to optimal NS cards within streaming server nodes. It also activates dynamic loading when a requested content is not available completely. Additionally, the scheduler manages content usage information.

The Distribution Manager maintains overall metadata required to manage the content distribution, and provides utilities that control the content distribution job having on-demand, scheduled and/or prefix mode. The Storage Manager keeps track of storage usage of each NS card and if needed, activates content purging. The Replication Manager checks the necessity of replication of some contents periodically, and if needed, registers replication jobs. A replication job is processed in similar way to a transfer job.

4. Content Storage Management

Even though each NS card of SMART server provides large capacity of storage, all of the multimedia contents serviced to end users cannot be stored at the server. Also, the more the server caches streaming media, the more the server can provide services to users. From these facts, our system adopts prefix caching mechanism for making more services to be available. The prefix caching requires dynamic loading mechanism to support streaming service completely. As a result, there are many concurrent transfer jobs in service systems.

When a transfer job is run, it is needed to check the available storage. But the check procedure cannot be accomplished due to the concurrent transfer jobs. Therefore we need a mechanism for managing storage. The storage information is also helpful to determine content placement position.

For storage management, we maintain a SMT table at each local server as shown in figure 5.

Node_ID	NS_ID	Current	Reserve	Capacity

Figure 5. The Structure of SMT(Storage Management Table)

The 'Node_ID' field of SMT indicates a node in a cluster environment. The 'NS_ID' field is maintained for indicating a NS card in a node. The 'Current' field represents current amount of storage for each NS on each node in a cluster. While the 'Reserve' field indicates the future amount of storage which is sum of current amount of storage and the amount of storage occupied by currently transferred contents. The 'Capacity' field is for total capacity of each NS card on each node.

The storage management is executed in three cases. The first case is space reservation. After content placement position is determined using storage management information and other information such as node overhead, job manager has to dispatch the transfer job to determined server node. At this point, the job manager calls space reservation procedure with transferred content size. The space reservation procedure reserves space for content to be transferred. This makes the value of 'Reserve' field to be increased. If the value is greater than the Capacity value, purge operation is activated.

Second case is completion of a transfer job. If the transfer job is success, the ftp client notifies the initial size and final size of the content to the Agent. The Agent increases the value of 'Current' field for corresponding NS and node. The increased value is final size minus initial size. If the transfer job is failed, the ftp client notifies the initial size, final size and expected final size to the Agent. And then , the Agent increases the value of 'Current' field for corresponding NS and node by final size minus initial size, and decrease the 'Reserve' field by expected final size minus final size.

Final case is content purge job case. The purge command run at server node notifies whether or not the command is success to the Agent. Also it notifies NS and node information, and current content size. If the purge job is completed successfully, the Agent decreases the value of both 'Current' and 'Reserve' field by current size. Otherwise the job is failed, the Agent doesn't anything.

5. Performance Evaluation

We measured the performance of the proposed server by changing the type of contents format (MPEG-2, MPEG-4 and H.264), bit rate of contents and I/O block size, and so on. Also we tested the effect of clustering solution and prefix caching. The clustering and prefix caching utilize the storage management feature. All tests were run at TTA (Telecommunication Technology Association).

As shown in Figure 6, we evaluated the capacity of stream services with zero-loss by changing the number of NS cards and bit rate on MPEG-2 contents. From the results, we can know the fact that the proposed server supports 4.1Gbps for the 4Mbps content and 4.44 Gbps for the 10M and 20Mbps contents respectively. Other test results for single-node streaming server represent similar performance.

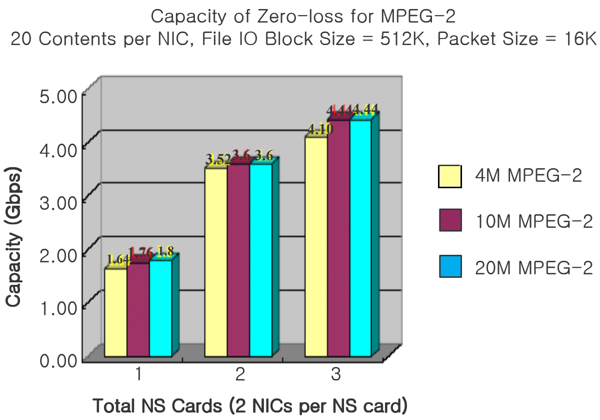

Figure 6. Zero-loss Server Capacity for MPEG-2

Figure 7 and Figure 8 shows the test results related to clustering and prefix caching features. The results tell us that clustering software distributes all incoming requests evenly to the participating streaming servers. Also, we can know that prefix caching doesn't affect the streaming performance, and from which we know the fact that prefix caching can help to give more chance to hit user requests.

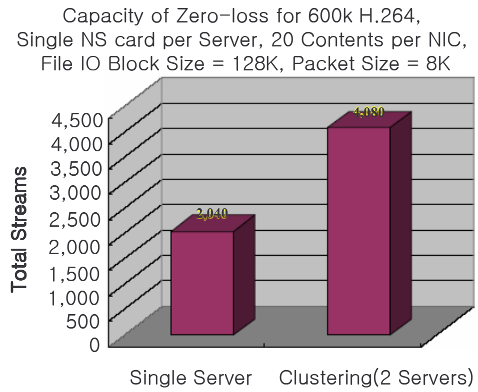

Figure 7. Zero-loss Server Capacity for Clustering

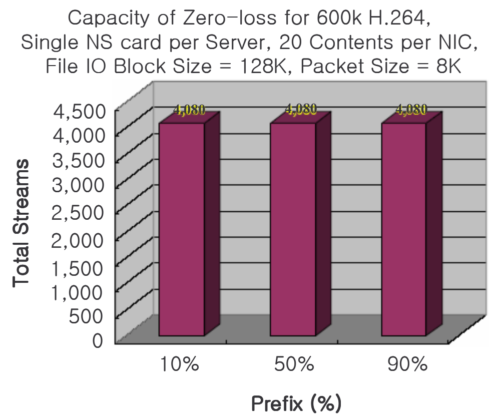

Figure 8. Zero-loss Server Capacity on Different Prefix Rates

6. Conclusions

In this paper, we described overall hardware and software architecture of the SMART server. And then, we proposed an efficient storage management mechanism in content distribution component. For the mechanism, we maintained SMT structure. The SMT has Node_ID, NS_ID, Current, Reserve, Capacity fields. The proposed mechanism processes the storage management in three cases, and at each case, different units of content distribution component are responsible for storage management.

Through proposed mechanism, prefix caching can be adopted in content distribution method. Also, the storage information can be utilized to determine position of content placement and request dispatching target.

We measured the performance on various situations and got desirable results. Specially, the clustering and prefix tests are related with storage management.

In order to provide total solution for multimedia streaming services, we have to design and implement the global server. Also the storage management method must be extended to global infrastructure.

REFERENCES

[1] Korea National Computerization Agency, Ministry of Information and Communication, 2002 Korea Internet White Paper, 2002
[2] Adaptec Company, “Advantages of a TCP/IP offload ASIC”, White paper
[3] Thomas Plagemann, Vera Goebel, Pal Halvorsen and Otto Anshus, “Operating System Support for Multimedia Systems”, The computer Communication Journal, Elsevier, Vol. 23, No. 3, 2000, pp. 267-289
[4] M.B. Abbott and L.L. Peterson, “Increasing network throughput by integrating protocol layers”, IEEE/ACM Transactions on Networking, Vol. 1, No. 5, 1993, pp. 600-610
[5] Gregory F. Pfisterl, In Search of Clusters, Prentice Hall PTR, 1998
[6] T. Schroeder, S. Goddard, and B. Ramamurthy, “Scalable Web Server Clustering Technologies”, IEEE Network, 14(3), 2000, pp. 38-45
[7] Wensong Zhang, “Linux Virtual Server for Scalable Network Services”, In Proc. of the Ottawa Linux Symposium, 2000
[8] Kasenna, “Video Content Distribution: A Hybrid DCS/Caching Architecture for Broadband Video”, White paper
[9] Subhabrata Sen, Jennifer Rexford, and Donald F. Towsley, “Proxy Prefix Caching for Multimedia Streams”, In Proc. of INFOCOM, 1999, pp. 1310-1319
[10] Stardust, “Content Networking and Edge Services: Leveraging the Internet for Profit”, White paper
[11] HCL Technologies Ltd, “Content Delivery Networks: An Architectural Overview”. White paper, May 2002
[12] “Next Generation Internet Server Requirement Specifications V1.0,” ETRI, September 2002.
[13] “Next Generation Internet Server: Content Distribution Subsystem Specifications Version 0.1,” ETRI, December 2002.