01 Jan 2005

An Efficient Method for Supporting Multiple Types of Services on SMART Server

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

Yu-Hyeon Bak

Seung-Joe Bae

Jin-Mee Kim

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/864

An Efficient Method for supporting Multiple types of Services on SMART Server

Chang-Soo Kim, Seung-Jo Bae, Jin-Mee Kim, Yu-Hyeon Bak, Sang-Min Woo, Seung-Hyup Jeon, Won-Jae Lee, Hag-Young Kim
ETRI(Electronics and Telecommunications Research Institute)
{cskim7, sbae, jinmee, bakyh, smwoo, shjeon, russell, h0kim}@etri.re.kr

***Abstract* — The SMART server is a special purpose server for efficient multimedia streaming service over high speed networks. For sending multimedia streams at high speed, the SMART server adopts a special hardware called as NS card. The SMART system also has a special file system module and streaming module.**

In this paper, we focused on distribution of contents between global servers and local servers. Specially, we propose a mechanism that multiple types of services can be supported efficiently. With this mechanism, the SMART system can provide effectiveness in managing storage and distribution policy.

***Keywords* — SMART, streaming, Content Distribution, Global Server, Local Server, Service Type, Content Management**

1. Introduction

Fast growing of high speed networks such as VDSL, FTTH makes users to require high quality of multimedia services. However, WAN (Wide Area Network) bandwidth is very limited to provide high quality of multimedia services to end users through wide area network infrastructure. To resolve this problem, a service provider of multimedia contents adopts CDN (Contents Delivery Network) technique that locates many local servers or cache servers at various specific regional areas. In CDN infrastructure, multimedia streaming service is provided with end users through a nearest local server [3, 4, 5, 6].

In general architecture for multimedia contents service, as shown in Figure 1, a global server is equipped with contents library having very large shared storage. The contents library stores all contents required for streaming to end users. A local server is a system appropriate to multimedia streaming service and located at a specific regional area to support nearest end users. The local server has relatively small capacity of storage and maintains a local cache to maintain contents provided by a global server. The local server provides streaming service at regional community through LAN (Local Area Network) which has high speed network bandwidth.

In the case that none or only prefix portion of a content requested by a end user exists in the cache of a local server, the local server can bring full or the rest of the content from the global server through WAN. At this point, some contents in the cache of the local server can be purged in order to guarantee enough storage space.

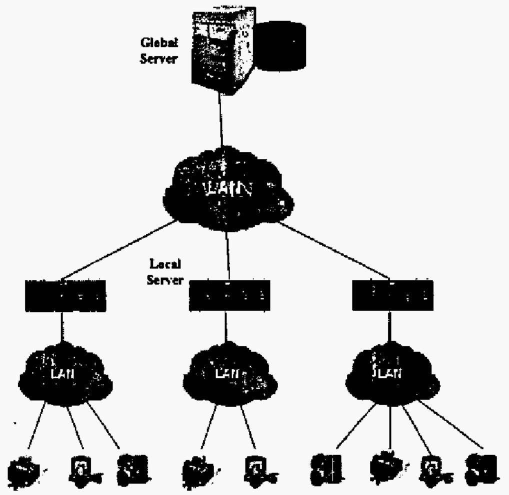

Figure 1. General architecture for multimedia contents service

In general CDN technique, the relationship between global servers and local servers is one versus many (1:N). In addition, each local server doesn't classify the types of contents to be serviced to end users and applies same rule for storage management. While the types of contents serviced to end users include video, education, news, and so on. Each content type has its own features, and so it is needed to deal separately.

The SMART (Server for Multimedia Applications in Residence community) system is a special purpose server system developed by ETRI (Electronics and Telecommunications Research Institute) which is designed for efficient streaming services over high speed networks in regional community[1, 2]. The SMART server plays a role as a local server.

In this paper, we focused on distribution of contents between global servers and local servers in SMART system. Specially, we propose a mechanism that multiple types of services can be supported efficiently. With this mechanism, we can remove the limitation that every local server can take only one global server, and the SMART server can provide effectiveness in managing storage and distribution policy. According to the type of contents to be serviced, the storage management including automatic content purging and automatic content placement can be varied.

The rest of this paper is organized as follows. In section 2, we describe overall SMART system itself, and in section 3, we explains the Content Distribution module's role and architecture. In section 4, we propose a mechanism to support multiple types of services, and finally show the conclusion with effectiveness of this mechanism and brief summary in section 5.

2. SMART server

The SMART server is planed to be located at regional community such as campus, hospital, apartment, and so on. It provides multimedia streaming service at regional community. One or more global servers will transfer some contents required by SMART servers located at a number of regional areas. The SMART server adopts special purpose hardware called as NS (Network Storage) card which combines NIC (Network Interface Card), memory and SCSI controller. The NS card can remove some overheads between storage access and network transmission, and accelerate the streaming service.

The overall hardware architecture of the SMART server is shown in figure 2.

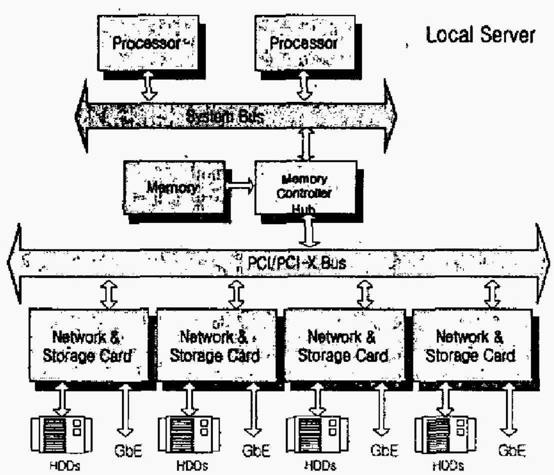

Figure 2. H/W Architecture of the SMART server

The SMART server has four NS cards. Each NS card has 2 Gbps Network Interface ports and several disks that store multimedia contents which can be accessed only through the specific network card.

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 called as CC (Content Container) to support NS card, large capacity of storage and efficient access to very large files. The CMM (Content Metadata Manager) maintains system configuration data, metadata for contents to be serviced and other service related data. The Content Streaming component is a middleware for transmitting some multimedia contents through CC interface, which communicates with NS module and OS interface, in order to provide streaming service to regional community at high speed bandwidth.

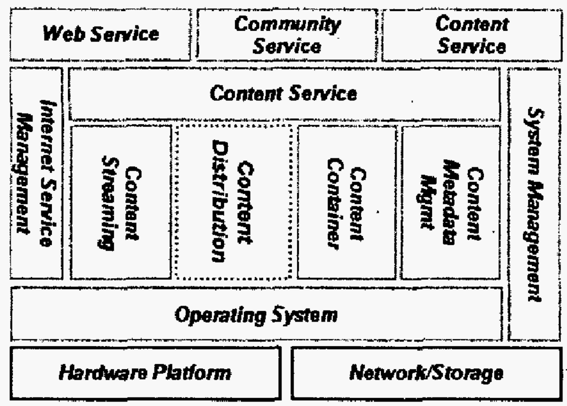

Figure 3. S/W Architecture of the SMART server

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

The SMART server can support up to 1000 clients and up to 200 concurrent clients with high-quality MPEG streaming service. Also, it can be re-started automatically after any abnormal shutdown. A SMART server has a good performance-price ratio.

3. Content Distribution

The Content Distribution (CD) component is mainly responsible to distribute required contents from a global server to the SMART server. The sub-modules of the CD are shown in figure 4.

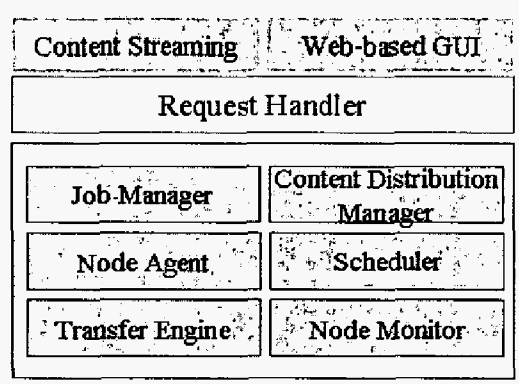

Figure 4. Sub-modules of the CD component

The CD has web-based GUI interface which runs an appropriate utility of various request handler command line utilities.

In the SMART system, a local server can be configured in cluster environments to support more end users. From this, the local server provides high scalability and high availability. The Job Manager which runs at a dispatcher node in the cluster, manages jobs submitted to the CD component from user or other modules of the CD, and distributes the jobs into appropriate sub-node. The Node Agent is run at every 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 Engine. The Transfer Engine is responsible to push or pull the requested content from/to a global server.

The Scheduler supports the single system image in cluster environment. If the SMART system is configured as cluster, the scheduler selects a sub node for best service when a user requests streaming service.

The Node Monitor collects various node statistics which is helpful to determine appropriate service node or content placement node.

The Content Distribution Manager maintains overall metadata required to manage the content distribution, and runs storage management, content purging, policy management, and so on.

4. An Efficient Method for supporting multiple types of Services

In this section, we propose an efficient method for supporting multiple types of services. For this mechanism, we extend the global infrastructure to contain several global servers and several local servers. In this infrastructure, each global server maintains a metadata table called as GCT. The GCT contains information about contents that the global server provides. The fields of GCT consist of GID (Global Server ID), CID (Content ID), Title of content, Global path, Release Date, the size of content, service type (video, education, news, and so on), Usage, bit rate, running time, and so on.

Also each local server maintains GCT table with same structure. But, at a local server, the GCT maintains only information of contents for the local server to service to its end user. Of course, a local server can maintains information from several global servers. If a local server plans to service all contents provided by all global servers, GCT of the local server maintains all combined information of those global servers.

Each local server also has LCT table to maintain information of contents that located at the local cache. The fields of LCT include GID, CID, NodeID (for cluster environment), Local Path, Content Status, Last Access Time, Stored Time, the number of total access, the number of current access, Original File Size, Current File Size, Service Type, and so on.

The global server provides some contents with local servers. At this point, each global server provider can select its profit policy such as fixed amount per month, transfer usage amount policy, and so on. Also a global server provider can select different profit policy according to the contents service type such as video, news, and so on.

While, each local server can select different policy in many local cache maintenance policies according to the contents service type. For example, it can select LRU policy for the video contents, and Release Date policy for the news service. Each local server has to select a proper policy with consideration of global server's profit policy and its own storage management policy. These policies have also to be applied to several content service types separately. In other words, each local server must maintain a different policy for each service type of content.

Service Name	Global ID	Service Type	Purge Policy	Placement Policy	Preloading Policy	Rate of Max Storage

Figure 5. The structure of CSPT

For this policy management, we maintain a CSPT (Content Storage Policy Table) table at each local server as shown in figure 5. The local server can apply appropriate policy for storage management (purge policy) according to the global server's profit policy and content service type.

In our system, each local server creates a directory for each content service type. All contents belong to a content service type are stored at the corresponding directory. Each directory has its own policies for purge, placement and preloading.

The CSPT has following fields. The 'Service Name' is identification of a service type determined by combination of the global server ID and content service type. The directory for a specific service name has a name same as the service name. If a service is newly added into the system, the system creates a directory named as the service name at each sub-node.

The purge policy indicates the selection method to remove some contents in order to ensure storage space for newly requested contents. Representative policies for the purge are LRU, USAGE, AGE, RANK, and so on. LRU selects content that is not serviced for longest duration; USAGE selects content that has lowest value for the field of 'the number of total accesses'; AGE selects content that has earliest Stored Date; and RANK selects content that has lowest rank value.

The placement policy is for selecting the location to store contents transferred from a global server. There can be STORAGE, CPU, NETWORK, CPU+NETWORK, and so on for this policy. Each policy for the placement selects a node that has lowest value for selected policy metrics.

When an end user requests a service for content that the local server has no any part of the content, the local server cannot support the request until at least the prefix part of the content exists at the local cache. This is no problem but cannot provide high quality of services. To prevent this situation, we run an automatic preloading daemon at the dispatcher node in the cluster. The automatic preloading daemon issues jobs to Job Manager periodically for preloading prefix parts of some contents satisfying some conditions.

The preloading policy of the CSPT indicates conditions for preloading some contents ahead by the automatic preloading daemon. The preloading policy can be such as RANK, NEWAGE, NEWAGE+RANK, and so on. The Rank selects contents that have highest value in the USAGE field of GCT. The NEWAGE policy selects contents that latest value in the Release Date field of GCT. The NEWAGE+RANK policy is combination of those two policies.

Finally, the 'Rate of Max Storage' of the CSPT indicates rate at which corresponding service type can occupy its underlying storage capacity. The SMART system maintains storage usage information for each node and each directory for a service type. The structure for this information is shown in figure 6.

Service Name	NodeID (for cluster)	Current	Reserve	Limit

Figure 6. The structure of SMT (Storage Management Table)

The 'Current' field of SMT indicates current amount of storage for each service at each node. The 'Reserve' field indicates the future amount of storage for each service at each node. The future amount of storage is sum of current amount of storage and the amount of storage occupied by currently transferred contents. The 'Limit' field contains a value computed by total storage capacity of the node * corresponding value of the 'Rate of max storage' in CSPT.

From above structures and mechanism for supporting multiple types of services, each local server can support several numbers of global servers, and each global server can provide its contents to several local servers. Also, each global server can adopt its own profit policy for each service type of contents. In addition, each local server can tailor its storage management policy considering corresponding global server's profit policy and service type of contents.

5. Conclusions

In this paper, we described overall hardware and software architecture of the SMART server. And then, we proposed an efficient method and needed structures for supporting multiple types of services. For the mechanism, we added some structures including GCT, LCT, CSPT and SMT. The GCT is maintained at both global server and local server. Others are maintained at only local server. We can apply various policies in CSPT for the storage management and preloading engine.

Through proposed mechanism, we can allow the relationship between global servers and local servers to be many to many. Due to the relationship, we can separate the global contents provider and streaming service provider at local community. This allows many contents provider and local streaming service provider to be activated with low cost. Also, we can apply best policies for each content service type and for profit policy of global content provider, and we can maximize the effectiveness of local content caches.

REFERENCES

[1] "Next Generation Internet Server Requirement Specifications V1.0," ETRI, September 2002.

[2] "Next Generation Internet Server: Content Distribution Subsystem Specifications Version 0.1," ETRI, December 2002.

[3] "Content Delivery Networks: An Architectural Overview," White paper, Networking Products Division, HCL Technologies Ltd. May 2002.

[4] S. Sen, J. Rexford, and D. Towsley, "Proxy Prefix Caching for Multimedia Streams", Proc. of IEEE NOSSDAV'99, Basking Ridge, NJ, June, 1999.

[5] R. Rejaie, H. Yu, M. Handley, and D. Estrin, "Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet," Proc. of IEEE INFORCOM'2000, Tel-Aviv, Israel, March 2000.

[6] Y. Wang, Z. L. Zhang, D. Du and D. Su, "A network-Conscious Approach to End-to-end Video Delivery over Wide Area Networks Using Proxy Servers," Proc. of IEEE Infocom'98, April 1998.