Efficient Transmission of Geographic Information Using Streaming Technology

JaeJun Yoo, KyoungBok Sung, JungSook Kim, Republic of Korea

Key words: Geographic Data Transmission, Streaming

SUMMARY

Recently, as some services using geographic information or spatial data such as telematics or LBS (Location Based Service) have been being more diverse and complex and mobile devices such as PDA (Personal Digital Assistance) or several telematics terminals have been being used more widely, efficient transmission of geographic information is being more important, especially in wireless environment.

To make data transmission more efficient in such geographic services, several enhancements are proposed and are being researched. One of basic concepts which those enhancements are based on is streaming. Transmitting geographic information using streaming mechanism can be designed and implemented in several levels according to granularity of data to be transmitted. For example, a file or a specific region based on several criteria can be the granularity.

In this paper, we suggest a novel and efficient transmission method based on streaming mechanism of which transmission unit is a block or a page. To do this, we 1) introduce the concept of data streaming which is tightly coupled to operating system, 2) design and implement the streaming system and 3) conduct simple experiments to show the performance of the implemented system.

Efficient Transmission of Geographic Information Using Streaming Technology

JaeJun Yoo, Kyoung Bok Sung, JungSook Kim, Republic of Korea

1. INTRODUCTION

Recently, as some services using geographic information or spatial data such as telematics or LBS (Location Based Service) have been being more diverse and mobile devices such as PDA (Personal Digital Assistance) or several telematics terminals have been being used more widely, mobile geographic services are being an essential software in routine life [4, 6]. Also, these mobile services are being extended to include more various functions to be used frequently. Within many of such services or systems, efficient transmission of spatial or geographic information from a network entity – for example, a service server – to another – for instance, a client or a mobile device – is very important because communication cost is expensive especially in wireless environment.

To make data transmission more efficient in such geographic services, several enhancements are proposed and are being researched. One of basic concepts which those enhancements are based on is streaming. Streaming is a mechanism to transmit a part of data or information instead of whole part during some operations. Therefore, using streaming concepts, we can transmit only the requested part of data.

Data streaming can be designed and implemented in several levels according to some views which the streaming concept will be applied to. For example, when we transmit a video file which consists of a number of frames, to send frame by frame of the file in playing the file can be a kind of data streaming. Also, to send a software package which consists of a number of different files file by file can be a kind of data streaming. Based on the granularity of data to be streamed, each data streaming can have different effects on transmission. Equally, transmitting geographic information using streaming mechanism can be designed and implemented in several levels according to granularity of data to be transmitted. For example, a file or a specific region based on several criteria can be the granularity.

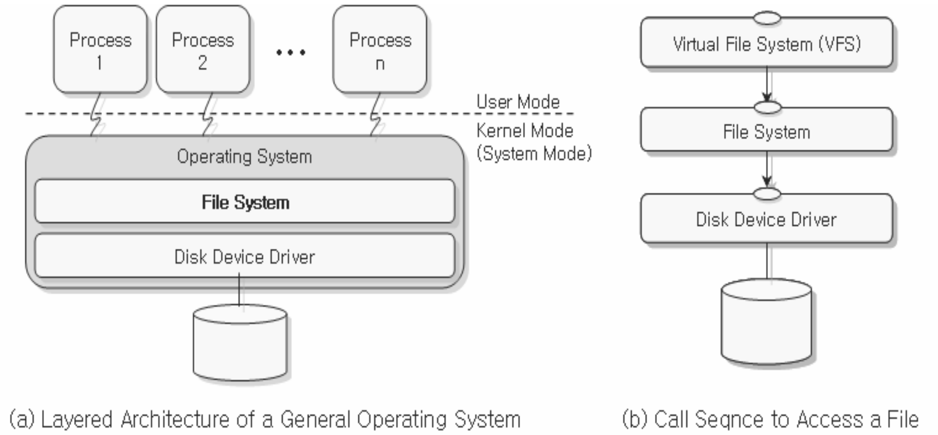
In this paper, we suggest a novel and efficient transmission method based on streaming mechanism of which transmission unit is a block or a page. (We will call the transmission based on streaming mechanism block by block or page by page to maximize transmission performance streaming the transmission tightly coupled to operating system.) To do this, we 1) introduce the concept of data streaming which is tightly coupled to operating system, 2) design and implement the streaming system and 3) conduct simple experiments to show the performance of the implemented system.

The structure of this paper is followings. In chapter 2, we explain the concept of data streaming of which transmission granularity is a block or a page. In chapter 3, we describe the design and implement of a streaming system for transmission of geographic information. In chapter 4, we describe a test platform to test the performance of the designed system by transmitting geographic information and shows results of experiments. Lastly, we conclude this paper in chapter 5.

2. TRANSMISSION USING STREAING WHICH IS TIGHTLY COUPLED TO OPERATING SYSTEM

In this paper, we introduce the concept of data streaming of which transmission granularity is a block or a page.

To explain streaming transmission tightly coupled to operating system, we have to mention general architecture of an operating system and call sequence to access a file in the operating system. Figure 1 (a) shows layered architecture generally used by an operating system. In such an operating system to access a file or data we have to call a sequence of functions or interfaces, in case of the operating system shown in figure1 firstly, the interface of Virtual File System, secondly the interface of File System for disk and lastly the interface of disk device driver, as shown in Figure 1(b). When accessing a file using the described call sequence, operating system moves data of a file in a certain granularity from lower layer to upper layer, which is a block or a page. (Generally, we use the term “block” when describing something in disk level and the term “page” when explaining something in memory level. However, in many cases, the block size and the page size is same [1].)



(a) Layered Architecture of a General Operating System (b) Call Seqnce to Access a File

Figure 1. General Architecture of an Operating System

Because when accessing a file we must use the function provided by operating system, if we adopt the transmission in a block-size unit or in a page-size unit which is being used in such an operating system to data streaming mechanism, we can implement a new kind of streaming transmission of which the transmission granularity is a block or a page. Different from existing data streaming mechanism, block or page based streaming is naturally minimize amount of transmitted data because it is the minimum size of data to be accessed using the operating system.

Figure 2 shows the concept of streaming mechanism which is tightly coupled to an operating system, that is, its transmission granularity is a block or a page which is adopted in this paper.

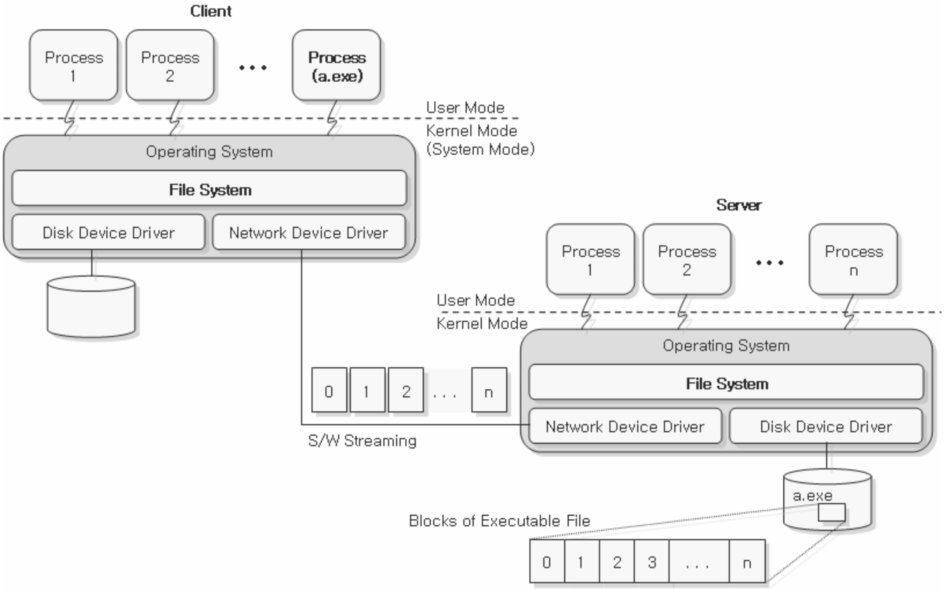


Figure 2. Streaming Tightly Coupled to Operating System

With the mentioned layered structure in an operating system, data streaming can be designed and implemented by reading the requested blocks or pages by operating systems from servers through network interface instead of reading them local storages. To implement this streaming concept, we need some supports of operating system to include our codes, that is, a kind of file systems, to operating system's file system level. Fortunately, the most of currently existing operating systems provide such concepts and related facilities. In case of LINUX, we can add our codes to operating system using VFS (Virtual File System) [5] and Stackable File System [2, 3] facilities.

To test the suggested streaming mechanism, we designed and implemented a file system called "StreamingFS" which is used to add S/W streaming mechanism to operating system and which is abbreviation of "Streaming File System".

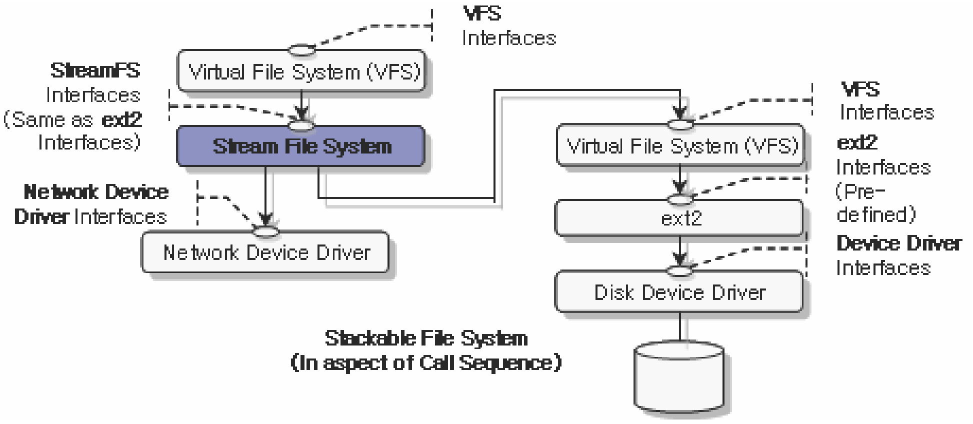

VFS Interfaces
StreamFS Interfaces (Same as ext2 Interfaces)
Virtual File System (VFS)
Stream File System
Network Device Driver Interfaces
Network Device Driver
VFS Interfaces
Virtual File System (VFS)
ext2 Interfaces (Pre-defined)
ext2
Device Driver Interfaces
Disk Device Driver
Stackable File System (In aspect of Call Sequence)

Figure 3. Call Sequence of File System Interfaces

Figure 3 shows call sequence of file system interfaces when we access an existing file using "StreamingFS". When we try to read a file using VFS interfaces, "StreamingFS" first check whether the requested blocks or pages were downloaded from server previously and stored in local storages or not. If the requested blocks are stored in local storages, they are read through traditional file system access interfaces and if not, they are read through network device driver from server.

3. SYSTEM ARHITECTURE

In this chapter, we describe the design and implement of a system for providing streaming transmission tightly coupled to operating system as mentioned above.

We designed the overall system architecture to support the descried streaming mechanism on Linux Platform in mobile communication environment. Open source and well known internal structures of LINUX are the reason why we selected LINUX as the main platform. Because the mechanism to access a certain file in an operating system is very different from the kind to the kind of operating system, if adopting this concept to another operating system environment such as Windows XP, we have to consider it a totally separated work

Figure 4 shows the whole structure of system architecture designed and implemented to provide the streaming transmission which is tightly coupled to operating system in functional block form. White blocks in Figure 4 mean pre-existing functional blocks in LINUX and Gray blocks represent new functional blocks to be implemented. .

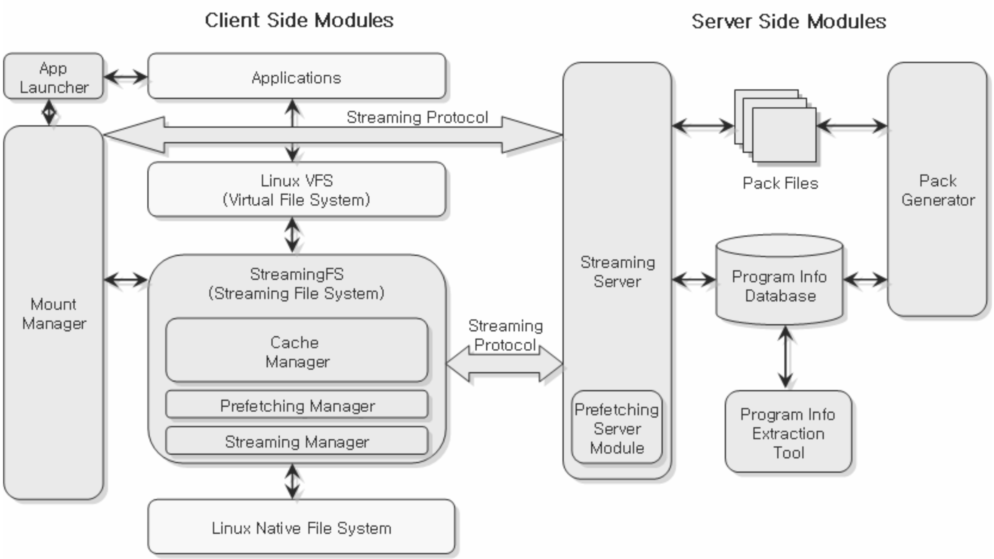


Figure 4. System Architecture of Streaming Transmission Which is Tightly Coupled to Operating System

On client side, following functional blocks are newly considered.

App Launcher It is a daemon process to execute the requested application after some pre-setting operations which are needed to provide S/W streaming for the application. Also, it passes some information about the selected application to Mount Manager.

Mount Manager It manages reference relationships among several file systems including StreamingFS, keeps some information about applications which are being executed in streaming fashion currently, and passes some detail data about executable applications to StreamingFS.

Streaming Protocol It is a predefined communication protocol specially designed for transferring geographic data to be accesses, information about the requested blocks or pages by operating system, and current status of client between a client and a server.

StreamingFS As the core part t of the streaming system, it provides file system interfaces letting operating system access itself, sends detailed information about application or general files to servers, and receives the requested blocks or pages as results. It has other three main components called Cache Manager and Streaming Manager, each of which is explained below.

Cache Manager To prevent retransmission of the request blocks or pages when the blocks are accessed again, it is necessary to save downloaded data or blocks in local storages. This is a kind of cache operation. This cache operation is supported by a functional block called Cache Manager.

Prefetching Manager To enhance performance of the streaming transmission, it needs to predict which blocks or pages will be requested and downloaded in the near future in advance. Although, if the prediction is mistaken, prefetching can lead communication and processing overhead, we can greatly improve system performance with accurate estimation of block usage. Moreover, to prefetch some blocks or pages can make data transfer more stable in mobile communication environment because accessing the current data can be proceed without any stop and system faults if some blocks or pages have been being cached previously. That it, stability of data communication between client and server can be higher.

To reduce overhead which can be occurred by miss prediction and to maximize system or data transmission performance, we need to forecast disk access pattern to be required as exactly as possibly. By figuring out the usage pattern, we can make transferring a geographic information faster and more stable. To describe extracting exact access pattern is out of the scope of this paper, and it will be explained in another related paper as soon as possible.

Streaming Manager It actually sends some information and data to server and receives the result from server. To communicate with servers, it interprets Streaming Protocol.

On server side, following functional blocks are newly introduced.

Streaming Server As the core functional block on server side to provide the designed streaming, it receives some information about the requested blocks or pages, or current client status and sends the requested blocks or pages as the result of processing.

Prefetching Server Module It is a specific sub-module dedicated to provide prefetching function. It communicates with Prefetching Manager on client side using subset of Streaming Protocol, which is specially designed for transferring efficient prefetching information such as access pattern of data accessed currently and some environment parameters related to prefetching, for example, size of transmission unit, time period of repeated requests and so on.

Pack Files To enhance of streaming mechanism, several files of data in streaming fashion are put together into a logical and physical package called “Pack File”. Generally, a “Pack File” can be seen as a general file. A “Pack File” contains additional index information such as what files consists of the application, what the relationships of the files are and so on.

Pack Generator It is a tool which creates Pack Files by analyzing dependencies among several kinds of data. Moreover, it analyzes what blocks or pages are repeatedly accessed.

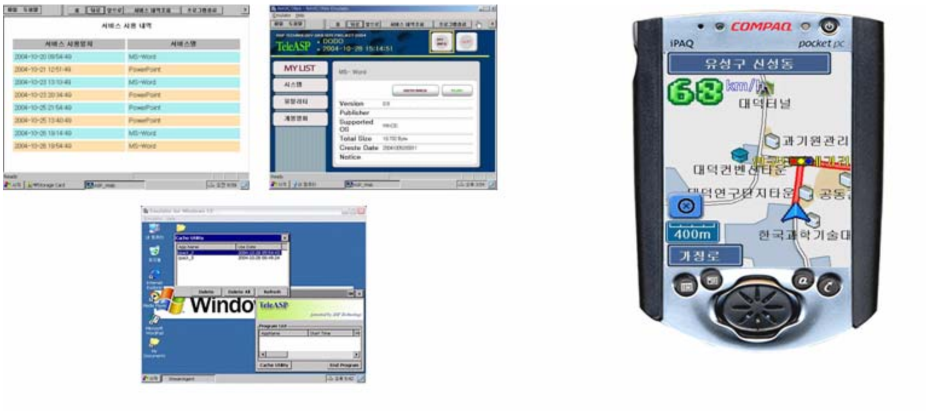
Data Info Database It is an information database which stores all kind of properties about streaming system and (geographic) data to be accessed. For example, information about data to be transferred to clients in streaming fashion, file size, creation data, and version number can be stored and queried.

Data Info Extraction Tool It is a small tool which can extract all kind of information about (geographic) data to be streamed. The extracted information is stored in “Data Info Database”

4. EXPERIMENTS ON THE DESIGNED STREAMING SYSTEM

In this paper, we implemented the server system to transfer geographic information to clients on Windows XP platform and the several clients on LINUX Platform, to test the designed and implemented streaming system, using Redhat Embedded Linux mobile version on HP iPAQ PDA.

User interfaces of the system designed to transfer some geographic information in streaming transmission which is tightly coupled to operating system are shown in Figure 5. Figure 5.(a) shows the user interfaces of servers, and Figure 5.(b) represents the user interfaces of a mobile geographic information program being executed with streaming on client side.



(a) Server User Interfaces

(b) Client Program User Interfaces

Figure 5. User Interface

The sample geographic information program on client was implemented as small and simple as possible. Main functions of the sample geographic information system is loading and unloading some geographic information, that is, map data, and displaying and searching the loaded data. Because to make efficient program is not our intention, we didn’t use complex and efficient internal approaches to manage geographic data such as spatial indexing, colorful display, and so on. We implemented the client program using combination of Tcl/Tk and C++ because the programming languages provide some functional toolkits or primitives for us to implement some prototypes in short time.

To examine performance of the designed system in this paper, we conducted some experiments using the implemented system and sample geographic information software. As in real life, we used the implemented prototype of geographic information software in mobile environment, that is, we downloaded it from server while moving and executed it repeatedly.
We used CDMA (Code Division Multiple Access) as mobile communication which is the most popular mobile environment because of cellular phones. The bandwidth of CDMA data transmission is about 9.6Kbps ~ 14.4Kbps.

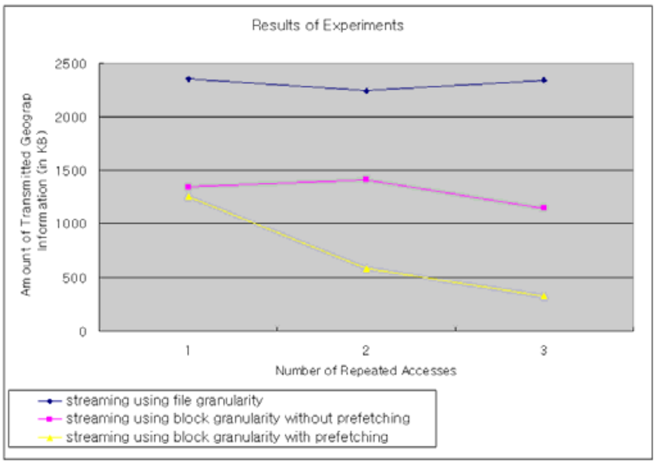


Figure 6. The Result of Experiments

Figure 6 shows the results of experiments. We downloaded, executed the sample geographic information program, and moved to a site to access geographic information of a specific region repeatedly using three approaches, 1) streaming with file level transmission granularity, 2) streaming with block level transmission granularity without prefetching function, and 3) streaming with block level transmission granularity with prefetching function.

In case of streaming with block level granularity without prefetching, the required and transmitted geographic information to be accessed can be smaller than that of streaming with file level granularity because only the requested blocks of data, not the whole data, are transferred to clients. In case of adopting prefetching function with block level streaming, transmitted data can be reduced much more because repeated access to the same blocks didn't make block transmission. Moreover, if the sample data was accessed several times, amount of transmitted geographic data is smaller because previously downloaded blocks or pages remain in client storage.

One defect of this experiment is that data transmission rate using CDMA is not constant. That is, by some external causes such as some problems of a communication company which provides CDMA environments, the data transmission rate can vary slightly. We couldn't control this factor in this experiment. However, the execution time difference between the approach that prefetching is added and the approach that prefetching is not adopted is so considerable that there is no problem when we show the effectiveness of prefetching mechanism in data streamin.g

Another consideration we have to keep focus with is pattern in which geographic information is stored. That is, According to the pattern and the relationship of blocks which have the geographic information, for example physical locations of blocks on disks, the results of experiments can have some variations. As mentioned previously, to extract block access patterns or to organize storage of accessed geographic information efficiently can be another interesting research topics to be investigated.

5. CONCLUSIONS

In this paper, we proposed advanced streaming method to enhance data transmission of geographic information. By adopting the designed streaming transmission which is tightly coupled to operating system with prefetching concepts, we could improve transmission performance and make the data transmission more stable. Using the suggested mechanism, we designed, implemented prototype system and conducts some experiments. The result of the experiments showed effectiveness of the designed system.

As following works, we will research some approaches to extract disk access pattern to enhance the performance of data transmission when accessing geographic information.

REFERENCES

[1] David A. Rusling, “Memory Management,”
http://www.tldp.org/LDP/tlk/mm/memory.html

[2] Erez Zadok, Ion Badulescu, “A Stackable File System Interface For Linux,”
http://www.filesystems.org/docs/linux-stacking/

[3] Erez Zadok, Jason Nieh, “FiST: A Language for Stackable File Systems,” USENIX Annual Technical Conference, San Diego, USA, 2002.

[4] JaeJun Yoo, In-Hak Joo, Kwang-Woo Nam, Jong-Hun Lee, “The Design and Implementation of A Video Geographic Information System,” Proceeding of Korean Information Science Society Fall Conference, Korea, vol. 29, No. 2. Sept. 2002.

[5] Neil Brown, “The Linux Virtual File-system Layer”,
http://www.cse.unsw.edu.au/~neilb/oss/linux-commentary/vfs.html

[6] Softbank Research Magazine, “LBS, Now & Future”, Strategy Report, Softbank Research, Korea, pp. 9-23, July, 2002.

BIOGRAPHICAL NOTES

- About 5 years research fields related to GIS (Geographic Information System), LBS (Location Based Services), and Telematics.
- Major Projects
  1. Development of Telematics Application Service Technology Based on USN Infrastructure, March. 2006. ~ Feb. 2008.
  2. Development of USN Middleware Technology, March. 2006. ~ Feb. 2008.
  3. Development of Advanced Streaming Technology for Telematics Environment, Jan. 2004. ~ Dec. 2005.
  4. Development of 4S-Integration Technology, Jan. 2002. ~ Dec. 2004.

CONTACTS:

JaeJun Yoo, KyoungBok Sung, JungSook Kim
Electronics and Telecommunications Research Institute
161, Gajeong-Dong, Electronics and Telecommunications Research Institute (ETRI)
Daejeon
Korea, Republic of (ROK)
Tel. : +82-42-869-1625
Fax : +82-42-869-1608
Email: {jjryu, kbsung, jungsook96}@etri.re.kr
Web site: None.