

---

# 데이터 지역성 메커니즘을 이용한 지역 스토리지 용량 관리 방법

김바울\* · 구민오\* · 민덕기\*

\*건국대학교

A Method to Manage Local Storage Capacity Using Data Locality Mechanism

Baul Kim\* · Mino Ku\* · Dugki Min\*

\*Konkuk University

E-mail : {bangwol\*, happykus\*, dkmin\*}@konkuk.ac.kr

## 요 약

최근 클라우드 컴퓨팅 기술의 발달로 인해 클라우드 기반의 컴퓨팅 자원의 활용은 다양한 분야에서 실생활에 적용되고 있다. 특히, 스마트 기기의 기술과 네트워크 인프라의 발달은 지역 스마트 디바이스와 클라우드 저장소간의 파일 공유에 대한 필요성을 더욱 촉진 시켰다. 하지만, 스마트 기기의 저장 공간 제약 사항으로 인해 클라우드 저장소에 파일을 저장하는 것은 지역 저장소의 저장 공간 기아 문제를 촉발 시킨다. 이것은 클라우드 저장소 서비스에서 대용량의 파일 저장 공간을 제공하더라도 사용자는 저장소 부족 문제에 직면 할 수 있음을 의미한다. 본 연구에서는 스마트 기기와 클라우드 저장소간의 파일 관리 방법을 제시한다. 본 연구에서 제시하는 지역 스토리지 용량 관리 방법은 사용자의 최근 사용한 날짜를 기반으로 파일 사용 패턴을 계산하고 이를 기반으로 지역에 저장된 파일들 중 마이그레이션 되어야 할 대상들을 선정하는 방법을 사용하고 있다. 더욱이 본 연구에서 제시하는 방법을 통하여 대규모 빅 데이터 저장소와 저장 공간의 제약을 가지고 있는 지역의 소형 클라이언트 장치간의 파일 동기화에도 활용될 수 있다.

## ABSTRACT

Recently, due to evolving cloud computing technology, we can easily and transparently utilize both local computing resource and remote computing resource in real life. Especially, enhancing smart device technologies and network infrastructures promote an increase of needs to share files between local smart devices and cloud storages. However, since smart devices have a limited storage space, storing files on cloud storage causes a starvation problem of local storage. It means that users can face a storage-lack problem even a cloud storage service provide a huge file storing space. In this research, we propose a method to manage files between smart devices and cloud storages. Our approach calculate file usage pattern based on recently used date, and then this approach determines local files being migrated. As a result, our approach is sufficient for handling data synchronization between big data storage farm and local thin client which contains limited storage space.

## 키워드

스토리지, 클라우드, 파일 관리, 파일 동기화

## 1. 서 론

클라우드 컴퓨팅 분야의 발전에 따라서 다양한 클라우드 서비스가 제공되고 있다. 사람들에게 잘 알려진 SaaS(Software as a Service) 클라우드 서비스의 하나인 파일 동기화 시스템은 원격지의 서비스 업체가 제공하는 스토리지를 바탕으로 내 디바이스의 파일이나 정보들을 백업하고 다른 디바이스에서도 사용이 가능하도록 만드는 기능을 제공하고 있다.

본 논문에서는 한정적인 스마트 디바이스의 지역 스토리지 자원을 효율적으로 사용하기 위해 사용자의 파일 사용 패턴을 분석하고 동기화하는 방법을 제시한다. 본 논문에서 제안하는 지역 스토리지 용량 관리의 특징은 무조건적인 파일 동기화가 아니라, 운영체제 페이지 교체 알고리즘 중 LRU(Least Recently Used) 알고리즘을 응용하여, 사용자가 필요한 파일을 사용할 수 있도록 마지막으로 사용한 최신 파일을 위주로 동기화를 하도록 설계하였다. 이 설계를 통하여 스마트 기기의 지역 스토리지 자원의 낭비를 막고 최소한으로 필요한 것만을 동기화하고자 하는 것을 목표로 한다. 오랫동안 사용되지 않는 파일은 클라이언트의 스토리지에서 제거하고 서버에만 저장하여 스토리지의 가용공간을 늘리도록 하였다. 파일을 전부 동기화 하는 시스템의 경우, 모든 파일을 동기화하기 위한 스토리지 자원의 낭비가 심하고, 선택적으로 특정 폴더에 대해 동기화를 제공하는 시스템의 경우 사용자가 일일이 설정하게 되는 번거로움이 나타나고, 그럼에도 불구하고 다시 낭비하게 되는 자원이 발생하게 되는 문제가 나타나게 된다.

특히 스마트 폰에 대해 사용자는 여전히 저장 공간에 대해 부족함을 느낀다. 본 논문에서 제안하는 방법을 이용하여 스마트 폰이 만들어내고 있는 데이터를 서버에서 관리하여 지역 스토리지의 한계에서 자유로워질 수 있게 한다. 스마트 폰이 만들어내고 있는 사진의 데이터는 1장당 1~3MB로 동영상의 경우 1초당 3MB 정도이다. 또 스마트폰에서 볼 수 있는 다양한 문서들도 수 MB에서 수 십MB등 저마다 용량을 차지한다. 어플리케이션의 경우에는 종류에 따라 다르지만 수 십MB에서 수 백MB까지 다양하다. 사용자에게 해서 매일, 매 주마다 발생하는 수많은 사진과 동영상들 또 여러 데이터들로 인해 스마트폰의 지역 스토리지는 점차 사용하게 되어 늘 부족한 용량에 시달리게 된다. 전 세계에서 만들어지는 빅 데이터 중 많은 용량을 차지하고 있는 사진, 동영상 데이터만 스마트 폰에서 줄이더라도 충분한 용량의 확보가 가능하다. 데이터 지역성 메커니즘을 이용한 용량 관리로 스마트 폰에서 확인을 안하고 잊어버린 오래된 사진이나 동영상, 각 종 문서파일 심지어는 어플리케이션까지 관리하고 자동으로 서버에 동기화를 한다. 동기화된 데이터의 경우 스마트 폰에서 제거가 되기 때문에 지역 스

토리지에 대한 용량을 차지하지 않아 해당하는 만큼 용량을 확보할 수 있다. 스마트 폰에서 제거할 수 있게 된다. 따라서 귀중한 사용자의 데이터를 버리지 않고 보관할 수 있게 된다.

본 논문에서는 최적화된 알고리즘을 통해 효과적으로 파일을 동기화 하는 과정보다는 스마트 기기의 저장용량을 효율적으로 사용하기 위한 시스템 설계방안에 초점을 맞추고 있다. 본 논문의 구성은 다음과 같다. 2장에서는 동기화와 스토리지와 관련된 기술 및 연구를 기존의 파일 동기화 시스템을 분석하고, 3장에서는 본 논문에서 제안하는 시스템의 체적인 구조와 지역성 메커니즘을 이용한 용량 관리 방법에 대하여 논하고, 마지막으로 4장에서는 결론 및 향후 연구 과제를 제시한다.

## II. 관련 기술 동향 및 연구

### 2.1 파일 동기화 서비스

파일 동기화는 파일을 복제하여 저장하고 항상 최신의 파일 상태를 유지하는 서비스로, 관련 기술의 대표적 상용 사례로는 Dropbox[1], Inc.가 제공하는 파일 동기화와 클라우드 컴퓨팅을 이용한 웹 기반의 파일 공유 서비스인 Dropbox와 네이버의 N드라이브[2]가 있다. Dropbox는 설치된 PC의 Dropbox 폴더를 서버와 동기화하여 저장하고 서버에 있는 모든 폴더를 자동으로 PC와 동기화하여 자동으로 다운로드하는 시스템이다. 스마트 폰에서는 자동으로 동기화하지 않지만 해당하는 파일 각각에 대해서 직접 조작함으로 사용자의 번거로움을 초래한다.

Ndrive 서비스는 Naver에서 제공하는 웹하드 서비스이다. 그 중에서 N드라이브 탐색기는 작업하는 PC에서 지정한 폴더를 서버와 동기화하는 것이 아니라 새로운 가상스토리지를 사용자에게 제공함으로 PC 지역 스토리지의 자원을 소비하지 않고도 클라우드 스토리지 서비스를 제공하지만 네트워크 속도에 의한 영향을 많이 받게 된다. N드라이브도 마찬가지로 스마트폰 환경에서는 Dropbox와 같은 번거로움이 나타난다.

### 2.2 No Disk System

김등화가 쓴 논문에서 Microsoft의 NDS(No Disk System)는 클라우드 컴퓨팅 기반의 스토리지 가상화 솔루션으로 클라이언트의 컴퓨터에서 OS 및 여러 프로그램이 동작하는 것이 아니라 중앙의 서버에서 관리하는 시스템이다 [3]. 모든 클라이언트들을 각각 관리하는 측면이 아닌 중앙의 하나의 서버만 관리함으로 필요한 스토리지의 양을 획기적으로 줄이게 된다. 하지만 NDS는 중앙서버와 클라이언트 간의 연결 속도가 빨라야 되며 일상적인 가정환경에서 혹은 모바일 환경에서는 클라이언트와 서버와의 거리가 멀어지게 되면 효율성이 떨어지고 또 많은 사람들이 같은 서비

스를 이용하기에 다양한 시스템을 사용하는 특성에 맞추기 힘든 측면이 존재하게 된다.

2.3 스마트폰을 위한 파일 관리 도구

박재민이 쓴 스마트폰 파일 관리 도구는 스마트폰에서 부족한 저장장치의 효율적인 이용을 위하여 가상화된 스토리지를 연결하고 필요한 파일들만 스마트폰에 받아서 사용할 수 있는 시스템이다. [4] 사용자의 선택을 중시하여 필요한 파일만 스마트폰에 남겨두는 방식을 이용하고 있어서 저장장치의 크기에 대한 제약을 해소했지만, 사용자가 직접 필요한 파일을 선택해야 한다는 번거로움이 있다.

2.4 클라우드 기반 센서 데이터 처리

박경욱이 제안한 클라우드 기반 센서 데이터 처리 시스템은 수 많은 센서의 데이터를 처리하기 위해 클라우드 개념의 게이트웨이를 도입하여 각 게이트웨이마다 일부 센서를 관리한다. 게이트웨이에서는 모아진 센서의 정보를 서버에 전송하고 지역 스토리지에 저장할 수 있는 데이터는 남겨두는 구조를 설계하여 최근 데이터에 대한 처리를 요구할 경우에 응답 속도를 높였다. [5]

III. 시스템 디자인

3.1 시스템 구조

본 논문에서 제안하는 지역 스토리지 용량 관리 시스템의 전체적인 구조는 그림 1과 같다. 클라이언트에 있는 데이터를 서버에 복제하는 시스템을 구성하고, 클라이언트에서는 파일의 정보를 확인하는 데몬이 백그라운드에서 돌아가고 있어서 동기화할 파일들에 대해 서버와 동기화를 하게 된다. 업로드와 다운로드를 자동적으로 이루어지지만 본 논문에서 제안하는 시스템에서는 지역성을 이용한 파일 동기화 방법을 통해 지역 스토리지를 관리한다.

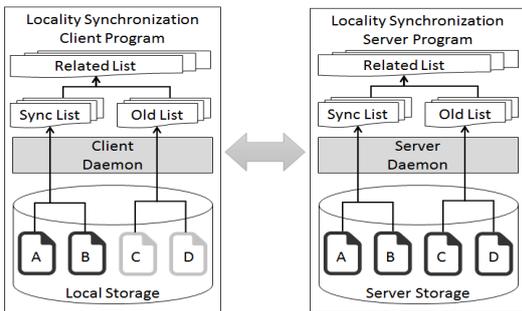


그림 1. 전체 시스템 구성도

그림 2의 A처럼 클라이언트 디바이스에서 새로운 파일이 생성이 되었거나 수정이 되었을 때 데몬은 해당하는 파일을 감지하게 된다. 해당하는

파일은 서버에서 추가 혹은 변경이 이루어지며 파일의 정보를 추가로 Sync List에 기록하게 된다. 서버는 Sync List에 들어가 있는 파일과 관련하여 지역성에 대한 방법론을 통해 인접한 파일의 정보를 확인한다. 확인된 인접한 파일들에 대해서 Sync List에 없는 파일은 Related List에 기록한다. Sync List에 들어가 있는 파일은 다른 클라이언트에서 접속했을 시에 자동으로 지역 스토리지에 다운로드하여 반영을 하게 되며 마찬가지로 Sync List에 연결된 Related List에 대해서도 클라이언트 기기에 해당하는 파일이 없을 경우 동기화하여 반영한다. 그림 2의 B처럼 오래된 파일은 Old List에 두어 동기화를 하지 않게 된다.

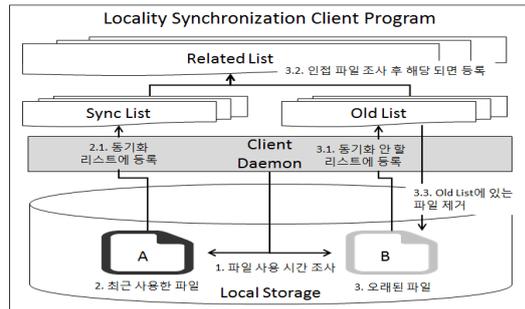


그림 2. Client Daemon 동작 순서

클라이언트 노드는 파일을 제거하기 위하여 Old List에 등록한다. 등록된 파일은 같은 디렉토리내에 연관된 파일에 대해서 관련성을 보고 Related List로 옮겨질지 Old List에 남겨둘지를 결정한다. 그림 3를 보면 C, D, G, H 파일은 사용된지 오래되어 Old List에 등록되었지만 C와 D의 경우에는 같은 디렉토리 내에 인접한 파일과 관련되어 Related List로 옮겨진다. 따라서 클라이언트 지역 스토리지에서 제거되는 것이 아니라 계속 동기화가 진행된다. 지역 스토리지에서 파일이 제거된 경우 제거된 파일에 대해 가상의 파일을 대신 제공하여 사용자가 보기에는 그대로 있는 것처럼 만들어주게 된다. 나중에 사용자가 해당 파일을 사용하게 될 경우 Old List에서 Sync List로 정보가 바뀌고 다운받아 실행하고 추후에는 동기화를 진행한다.

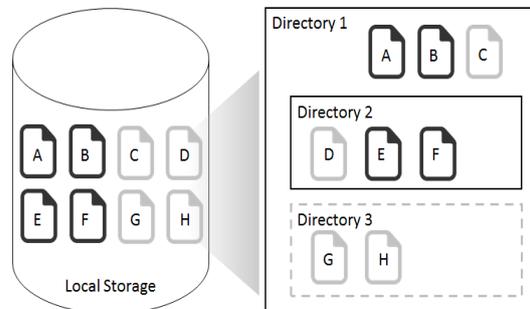


그림 3. 지역성 메커니즘 용량 관리

클라이언트 데몬과 마찬가지로 서버의 파일 동기화 목록도 세 개의 리스트로 관리하기 때문에 클라이언트 지역 스토리지에서 제거된 파일을 서버에서 역으로 동기화하여 다운하는 문제를 막는다.

### 3.2 지역성 메커니즘을 이용한 용량 관리 방법

제안하는 지역성 메커니즘을 이용한 용량 관리 방법은 LRU(Least Recently Used)에 기반을 두고 있다. 사용한 파일은 지역 스토리지에 남겨두고 사용하지 않은 파일은 서버에 이동함으로써 지역 스토리지의 공간을 확보한다.

$$(2T_r - T_l - T_d - \alpha_{day}) \geq 0 \quad (1)$$

$$T_d = \bigvee_{n=0}^n T_n \quad (2)$$

$T_r$ 은 현재 시간,  $T_l$ 은 마지막으로 사용한 시간,  $T_d$ 는 같은 디렉토리 내의 가장 최근에 사용한 파일의 시간,  $\alpha_{day}$ 는 파일을 보관하는 최장 기간을 결정하는 값이다. 위의 값들을 기반으로 파일을 제거할지 안할지를 선택하게 된다. 본 논문에서 제안한 알고리즘은 LRU에 기반 한 것으로 일정 기간 사용이 안 된 파일을 클라이언트에서 제거하고 서버로 이동시키는 값을 결정한다. 본 논문에서는 예시로  $\alpha_{day}$ 를 30일로 계산할 경우 식 (1)에서  $T_r$ 을 기준으로  $T_l$ 과  $T_d$ 가 같은 경우에는 최대 30일을 이상 사용하지 않은 파일을 클라이언트에서 제거한다.  $T_l$ 과  $T_d$ 가 다를 경우에는  $T_d$ 가  $T_r$ 과 같을 경우에 최대 60일 이후에 제거한다.

식 (2)에서는 인접한 파일로 같은 디렉토리 내의 마지막으로 사용한 시간 값을 계산한다.  $T_d$ 는 각 파일들을 확인해서 가장 최근에 사용한 날이 최대가 되는 값이다. 조사하려는 파일이 속한 디렉토리 내의 사용 날짜를 계산하여 가장 최대 값, 즉 최신 값을 계산한다. 같은 디렉토리 내에 한 파일을 참조했을 경우에 인접한 파일들도 같이 사용할 수 있다는 지역성을 기반으로 동기화의 개념이다.  $T_d$ 값을 이용해 지역성 기반의 사용 예측을 하고 클라이언트의 파일 이동 유무를 더 포괄적인 범위에서 바라보게 만든다. 따라서 제거하려는 파일이 30일이 지나더라도 시간 값의 계산 결과에 따라서 제거가 되지 않고 지역 스토리지에 보존 될 수도 있다. 본 논문에서 제안하는 방법은  $\alpha_{day}$ 를 따라 계산된 날 동안 파일이 클라이언트에 존재할 수 있게 하였다.

## IV. 결 론

본 논문은 파일 동기화 기법을 이용하여 클라이언트 지역 스토리지의 가용용량을 늘리는 방안을 제안한다. 기존의 기술들은 동기화 속도, 효율성, 안정성에 중점을 두었다. 하지만 부족한 스토리지 자원을 활용하기 위해 본 논문에서 저장 공

간의 가용방법에 대한 개념을 추구하고, 더 효과적으로 저장 공간을 이용할 수 있게 하였다. 저장 용량에 더 많은 영향을 받는 모바일 환경에서는 제안하는 시스템을 이용함으로써 오랫동안 안 쓰는 어플리케이션이나 사진, 동영상 파일을 서버에만 두어 부족한 스토리지 공간을 더 유용하게 활용할 수 있을 것으로 예상된다.

현재의 시스템은 단순히 서버와 클라이언트간의 데이터 이동에만 국한하고 있다. 향후 폴더 구조에 대한 고려와 다 계층으로 이루어진 폴더에 대해서 상위 폴더와 하위 폴더에 대해 어떤 동기화 방식을 제공할 지를 실험과 검증을 통해서 다양한 상황을 고려한 정확한 알고리즘을 구하는 연구를 수행할 것이다.

## 감사의 글

본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학IT연구센터육성 지원사업의 연구결과로 수행되었음 (NIPA-2013-(H0301-13-1012))

본 논문의 교신저자는 민덕기 교수(dkmin@konkuk.ac.kr)임.

## 참고문헌

- [1] Dropbox, Inc. Cloud storage and file synchronization software Dropbox. Available: [http://en.wikipedia.org/wiki/Dropbox\\_\(service\)](http://en.wikipedia.org/wiki/Dropbox_(service))
- [2] Naver. Cloud storage and file synchronization software Ndrive explorer. Available: <http://ndrive.naver.com/index.nhn>
- [3] Deunghwa Kim, Jaeung Namgung, Jungheum Park, and Sangjin Lee, "User behavior analysis in No Disk System Configuration", Journal of The Korea Institute of Information Security & Cryptology(JKIISC), VOL.23, NO.3, June 2013
- [4] Park Jae Min, and Ko Young Woong, "Design and Implementation of File Management Tool for Smart Phone", Korean Institute Of Information Technology, 2012
- [5] Kyoung-wook Park, Kyong-og Kim, Kyeong-jin Ban, and Eung-kon Kim, "Design and Implementation of Cloud-based Sensor Data Management System", The Korea Institute of Electronic Communication Sciences, 2010