

## 데이터 그리드를 위한 HTTP-FUSE 원격 파일 시스템 구현

김상원<sup>○</sup>, 함재균, 명훈주, 곽재혁, 황영철

한국과학기술정보연구원 고성능연구망사업단 그리드컴퓨팅연구팀  
{ sangwan<sup>○</sup>, jaehahm, hjmyung, jhkwak, hychul }@kisti.re.kr

An Implementation of HTTP-FUSE File System for Data Grid

Sangwan Kim<sup>○</sup>, Jaegyoon Hahm, Hoonjoo Myung, Jae-Hyuck Kwak, Youngchul Hwang  
KISTI(Korea Institute of Science Technology Information) HPCn Department  
Grid Computing Research Team

### 1. 서론

본 연구에서는 리눅스의 사용자 공간에서 파일시스템을 구현할 수 있는 라이브러리인 FUSE(Filesystem in Userspace)를 이용하여 웹서버의 파일시스템에 있는 파일과 디렉터리에 접근 할 수 있는 시험적인 소프트웨어를 구현하였다. 본 연구의 구현 결과를 이용하면 원격 웹서버상의 저장 공간을 로컬에 있는 파일 시스템으로 마운트 하여 사용할 수 있게 된다. 본 연구의 결과는 데이터 그리드 구현에 사용될 수 있을 것으로 기대된다. 데이터 그리드는 스토리지 서버를 상호 연동하여 사용하는 것으로, 데이터 공유와 저장장치의 논리적인 통합이 요구된다. 본 연구의 결과를 활용하면, 가장 일반적으로 사용되는 HTTP 프로토콜을 사용하여 원격서버의 파일시스템을 로컬시스템과 연동할 수 있게 되므로, 데이터 그리드에 유용하게 사용될 수 있을 것으로 기대한다.

### 2. FUSE 파일 시스템

FUSE(Filesystem in Userspace)[1]는 파일시스템의 기능을 커널 공간(kernel space)이 아닌 사용자 공간(userspace)에서 구현한 라이브러리로서, 리눅스 운영체제에서 동작한다. FUSE는 라이브러리와 API로 구성되어 있으며, 커널의 패치나 재컴파일은 필요 없이 커널모듈로 간단하게 설치할 수 있다. FUSE 파일 시스템의 구현은 사용자공간에서 동작하며 커널 인터페이스는 효율적으로 설계되어 있으며, 안정적이라는 평가를 받고 있다.

FUSE 시스템은 리눅스의 VFS(Virtual Filesystem Switch) 기반으로 한다. 리눅스 운영체제에서 파일 시스템 호출은 glibc를 통해 VFS로, 다시 파일시스템의 마운트 정보에 따라 커널 내에서 해당 파일 시스템의 구현 부분으로 전달된다. FUSE에서는 파일시스템에 대한 요청을 커널에서 처리하지 않고, 다시 사용자공간으로 돌려보내어 처리하게 된다. 이때 FUSE 라이브러리(libfuse)가 사용되며 libfuse와 FUSE 커널 모듈간의 통신은 /dev/fuse 를 통하여 이루어진다.

### 3. HTTPFS

HTTPFS[2]는 FUSE를 사용하여 구현되었으며 HTTP 프로토콜을 이용하여 원격지에 있는 파일을 로컬 시스템에 마운트 하여 사용할 수 있다. HTTPFS 는 원격 웹서버를 통하여 접근이 가능한 파일을 로컬 디렉터리에 마운트하여 읽을 수 있다.

HTTPFS의 동작 원리는 HTTP 프로토콜의 부분 전송(partial transfer) 기능을 이용하는 것이다. 이 기능은 웹서버에 대한 HTTP GET 요청시 Range헤더를 이용하여 해당 파일에서 임의의 부분만 요청하는 것이다. HTTPFS는 HTTP의 부분 전송 기능을 이용하여 파일 시스템에서 특정한 파일의 열기/읽기 시스템 호출을 HTTP GET 요청으로 바꾸어 처리한다.

그러나 HTTPFS의 단점은 단순히 웹서버로 접근 가능한 하나의 파일에 대해서만 동작을 한다는 것이

다. 하나의 파일에 대한 읽기 기능만 가능하므로 디렉터리를 포함한 파일시스템의 내용전체를 로컬 시스템에서 접근할 수는 없다는 기능적인 제약이 있으며, 또 읽기에 대한 것만 가능하며 쓰기는 가능하지 않다.

#### 4. HTTP-FUSE 파일시스템

본 연구에서는 HTTPFS와 유사하지만, 웹서버의 파일시스템의 디렉터리와 파일의 구조까지 접근할 수 있는 기능을 구현하였다. 본 연구에서 구현한 결과를 HTTP-FUSE라고 명명하였으며, FUSE에서 제공하는 파일시스템 기능 중에서 디렉터리 목록 읽기(readdir), 파일 상태 읽기(getattr), 심볼릭 링크 정보 읽기(readlink), 파일 열기(open), 파일 읽기(read)의 기능을 구현하였다.

HTTP-FUSE는 HTTPFS와는 달리, 웹서버에서 http-fuse의 요청을 처리할 수 있는 간단한 PHP 스크립트가 설치되어 있어야 한다. 이 PHP 스크립트는 HTTP-FUSE 구현 프로그램으로부터 웹서버로 전송되는 GET 요청을 처리하여 돌려주며, 웹서버상의 특정 디렉터리를 기준으로 삼아 디렉터리 구조를 클라이언트로 전달한다.

#### 5. 성능비교

본 연구에서 구현한 HTTP-FUSE의 성능을 테스트해 보았다. 비교를 위해 HTTPFS와 웹 사이트 다운로드 프로그램인 wget (<http://www.gnu.org/software/wget/>)을 이용하였다.

http-fuse가 실행되는 클라이언트와 웹서버가 LAN상에 있고, 웹서버에 1GB 크기의 파일을 만들고 HTTP-FUSE로 웹서버의 파일 시스템을 마운트 시킨 후, 클라이언트에서 cp 명령을 이용하여 웹서버에 있는 파일을 로컬로 복사하는데 걸리는 시간을 측정하였다. 같은 환경에서 HTTPFS를 이용하여 파일을 마운트하고 복사하는데 걸리는 시간과, 동일한 파일을 웹서버에서 클라이언트로 wget을 이용하여 다운로드 하는데 걸리는 시간을 측정하였다.

측정결과 HTTP-FUSE를 이용했을 때 6.70MB/s, HTTPFS의 경우가 7.13MB/s wget을 이용했을 때 8.61MB/s 속도가 측정되었다. 즉, HTTP-FUSE가 wget에 비교했을 때 약 80% 정도의 성능을 보이는 것으로 측정되었다. 성능저하의 원인을 살펴보면, 일반적인 파일시스템에서는 open -> read -> read -> ... -> read -> close 순서로 시스템 호출이 되지만, FUSE를 이용하여 원격의 파일에 접근을 할 때 로컬에서 한번의 read 시스템 호출을 처리하기 위해 원격 서버에서는 파일을 open-read-close로 처리해야 하는 때문이다. 이것은 HTTP-FUSE와 HTTPFS의 경우 모두 해당하며, HTTP-FUSE에서는 클라이언트의 요청을 Apache가 자체적으로 처리하지 않고 PHP 스크립트에 의해서 처리하기 때문에 약간의 오버헤드가 추가로 발생하였다.

#### 6. 결론

본 연구에서 구현한 프로그램은 파일시스템에 대한 읽기 기능만을 지원하지만, 클라이언트에서 웹서버 쪽으로 쓰기 기능도 마찬가지로 방법으로 손쉽게 확장이 가능하다.

데이터 그리드에서는 원격 저장 공간에 대한 손쉬운 접근과 검색이 중요하다. HTTP-FUSE는 현재 가장 널리 사용되고 있는 HTTP 프로토콜을 이용하여 파일 시스템을 구현함으로써, 원격 데이터 저장소에 접근성이 뛰어나며, 성능과 보안적인 측면 조금 더 고려한다면, 데이터 그리드 구현을 위해 유용한 도구가 될 것으로 기대한다.

#### 참고자료

- [1] M. Szeredi. Filesystem in Userspace - <http://fuse.sourceforge.net/>, February 2005.
- [2] HTTPFS - <http://httpfs.sourceforge.net/>