데이터의 효율적인 관리를 위한 데이터 그리드 시스템 설계

김상완°, 곽재혁, 황영철, 이필우
한국과학기술정보연구원 슈퍼컴퓨팅센터 그리드연구실
{sangwan°, jhkwak, hychul, pwlee}@kisti.re.kr

A Design of Data Grid System for Efficient Data Management

Sangwan Kim°, Jae-Hyuck Kwak, Youngchul Hwang, Pillwoo Lee
KISTI(Korea Institute of Science Technology Information) Supercomputing Center
Grid Technology Research Department

요 약

데이터 그리드는 방대한 양의 데이터를 효율적이면서도 편리하게 관리하는 것이 목적이다. 최근에는 멀티미디어와 정보화 기기의 발달로 개인 데이터의 양이 점차 많아 지고 있는 추세이며, 휴대전화와 PDA, PC등 다양한 환경에서 개인 데이터에 대한 접근을 필요로 하고 있다. 본 연구에서는 다양한 클라이언트 환경을 지원하기 위한 그리드 파일 시스템을 개발하고 구현하였다. 본 연구에서 개발된 시스템에서는 관계형 데이터베이스를 이용하여 웹 및 전용 클라이언트를 비롯한 다양한 클라이언트를 지원할 수 있으며, URL과 같은 고유의 파일 위치 정보를 이용하여 파일의 접근 방식을 표현한다.

1. 서 론

컴퓨터 관련 기술과 인터넷의 발달로 현대인들은 매일 상당한 양의 디지털 데이터를 접하고 생성해 내고 있다. 인터넷을 통해 다운로드 받은 음악파일을 비롯하여 업무와 관련된 문서파일, 디지털 카메라나 캠코더로 찍은 사진이나 동영상, 보이스 레코더로 녹음한 음성파일 등 디지털화 되어 있는 정보를 매일 접하며 생활하고 있다. 또한 기존의 아날로그 정보들이 점차 디지털화 되면서 정보화 시대에 우리가 접하고 있는 디지털 정보의 양은 급격히 증가하고 있다.

데이터 그리드는 컴퓨터 네트워크로 연결되어 있는 분산된 저장장치들을 효율적으로 사용하기 위한 기술이다. 데이터 그리드 분야에서 다루어지는 연구 분야들을 정리하면 다음과 같다. (참조:[1])

- 저장장치 통합: 독립된 저장장치들을 하나의 저장 장치로 통합 및 동일한 방법으로 접근
- 정보 검색: 속성 기반의 데이터 접근, 메터데이터 마이닝을 포함한 정보 검색 기술
- 데이터 네이밍: 분산된 데이터들의 위치를 표현하는 방법
- 데이터 복제 관리 및 복제된 데이터의 일관성 유지
- 데이터 공유 및 보호, 데이터 접근 관리

본 연구에서는 데이터 그리드에서 요구하고 있는 다양한 기능들을 만족시키기 위한 데이터 관리 시스템을 설계하였다. 본 논문의 구성으로 제2절에서는 기존 데이터 관리 기법들의 장단점에 대해서 알아보고 제3절에서는 본 연구에서 설계한 시스템의 기본 기능과 화면 구성을 소개한다. 제4절에서는 기본 구조를 확장한 구조를 설명한다. 제5절에서 결론을 맺도록 한다.

2. 기존의 데이터 관리 기법 및 요구 사항

컴퓨터에서 다루어지는 데이터들은 하드디스크, 테입장치, 광 저장장치, 플로피디스크, 메모리스틱 등의 다양한 저장장치에 보관된다. 컴퓨터 네트워크가 발달을 하면서 컴퓨터와 컴퓨터 간에 데이터(파일)를 주고받을 수 있는 다양한 방법 또한 다양하게 개발되었다.

표1은 일반적으로 많이 사용되는 데이터 전송 및 관리 기술들의 장단점을 비교한 것이다. 데이터의 관리 방법들이 다양한 이유는 각각 나름대로 장단점을 가지고 있으며 사용되는 환경과 용도도 조금씩 다르기 때문이다.

표1. 기존 데이터 관리 기술의 장단점

구분	장점	단점
FTP	- 단순하며, 다양한 플랫폼에서 사용가능 - 빠른 전송 속도	- 파일 전송이 목적이며 매우 단순한 기능만을 제공함 - 방화벽에 영향을 받음
Samba	- 다양한 플랫폼간 파일 시스템 공유 - 윈도우 네트웍드라이브로 연결	- LAN 환경에서만 사용가능 - 보안에 취약한 편
NFS	- 유닉스 시스템간 가장 일반적인 파일 공유 방법	- LAN 환경에서만 사용가능 - 관리자의 세심한 관리가 필요
웹하드 Web Hard	- 인터넷 브라우져만 있으면 사용가능 - 사용자간 파일 공유가 쉬움	- HTTP를 기반함으로 용량이 큰 파일을 다루기 어려움
피어투피어 (P2P)	- 불특정 다수간 파일의 공유 - 대용량 데이터의 빠른 전송	- 데이터의 개인화가 어려움 - 전송 속도가 일정치 않음
메신저 Messenger	- 개인 사용자간 간편한 파일 전송	- 전송만을 목적으로 함 - 방화벽 영향 받음

데이터 그리드는 그리드 컴퓨팅의 한 분야로 연구가 시작되었다. 주로 계산 연구 분야에서 쏟아져 나오는 방대한 양의 데이터를 관리하기 위해 연구가 시작되었으나, 대규모 데이터의 관리는 개인용 컴퓨터를 사용하는 일반인들에게도 해당된다. 특히 요즘과 같이 개인용 디지털 기기들이 보편화 되면서부터는 개인들이 가지고 있는 멀티미디어 데이터가 많아지면서 데이터에 대한 효율적인 관리가 요구되고 있다. 표2는 계산 과학 분야와 일반 분야에서 데이터 처리 관련 요구사항을 비교한 것이다. .

표2. 기존 데이터 관리 기술의 장단점

	계산과학분야	일반분야
최대파일크기	수백 GB 이하	수 GB 이하
전송속도	수백 Mbits/s 이하	수십 Mbits/s 이하
사용용도	계산 응용 데이터, 처리결과 등 수치데이터	개인 파일, 사진, 동영상, 영화, 음악 파일 등 다양함
접속환경	개인PC, 연구실내 워크스테이션, 슈퍼컴퓨터	개인PC, PDA, 휴대전화, 차량용 단말등 다양함
데이터 생성	응용 계산 프로그램, 특수 연구 장비	디지털 카메라, 캠코더, 보이스레코더, 휴대전화 등 다양함
주요 관심사항	데이터 저장능력, 전송속도, 처리 속도	편리성, 전송속도, 접근용이성, 개인정보보안, 저장용량, 비용, 공유

표2에서 나타낸 바와 같이 데이터 관리 부분에 있어서는 일반 분야의 사용용도와 접속 방법, 데이터 생성 소스 등이 계산 과학 분야보다 더 다양하다. 계산 과학 분야에서는 접속환경이나 편리성보다는 데이터의 크기가 큼에 따라 저장 용량이나 처리 속도 전송 속도가 중요시 된다. 따라서 데이터 그리드를 설계하는 입장에서는 그 용도와 접속환경이 다양한 일반 사용분야에서 접근하는 것이 더 바람직하다. 데이터 그리드의 다양한 기능적인 면들을 우선 고려를 하고, 데이터의 전송 성능이나 저장 용량에 대한 문제는 그 이후에 고려하는 것이 바람직 하다.

3. 기본 구조

본 절에서는 본 연구에서 설계한 데이터 그리드 관리 시스템을 설명한다. 그림1은 본 연구에서 설계한 시스템의 기본 구조이다. 서버-클라이언트 구조로 되어 있으며, 디렉토리의 구조와 파일에 대한 메타데이터를 저장하기 위해 관계형 데이터베이스(RDB)를 사용하였다. 실제 파일의 내용은 파일 시스템 상의 일정 영역을 할당하여 파일을 저장한다. 파일의 업로드와 다운로드는 전용 서버와 웹서버를 통하여 이루어진다. 전용 서버는 FTP 데몬과 같이 계속 실행되어 있어 데이터에 대한 접근 요청을 처리한다. 웹서버는 웹 브라우져를 이용하여 파일의 업로드/다운로드를 가능하게 한다.

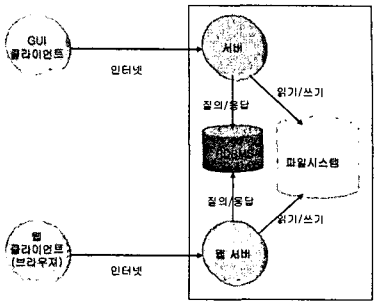

그림1. 기본 구조

그림2는 웹인터페이스를 이용하여 파일 시스템의 디렉토리를 브라우징한 모습니다.

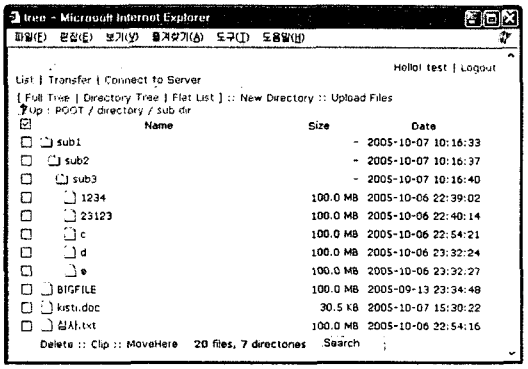

그림2. 웹 인터페이스 화면

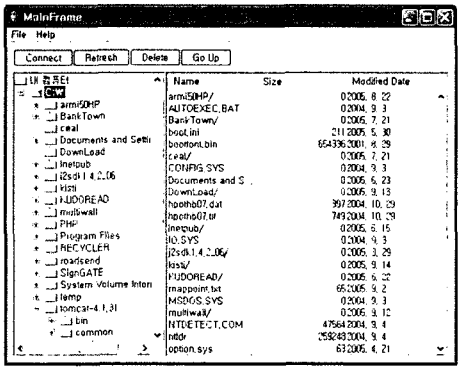

그림3. 자바 클라이언트 화면

그림3은 전용의 자바 클라이언트를 이용하여 파일 시스템을 브라우징 하고 있는 모습니다. 웹서버를 이용할 경우 웹서버의 특성상 대용량의 파일을 업로드 하는 것이 불편하고(100MB이하), 파일을 하나씩 따로따로 업로드 해야 하는 불편함이 따른다.

Field	Type	Null	Key	Default	Extra
fid	int(11)		PRI	NULL	auto_increment
parent	char(8)	YES		NULL	
ftype	char(1)	YES		NULL	
fname	char(255)	YES		NULL	
fsize	char(255)	YES		NULL	
fdate	datetime	YES		NULL	
username	char(50)	YES		NULL	
aflag	tinyint(1)	YES		NULL	
rflag	tinyint(1)	YES		NULL	
hash	char(50)	YES		NULL	

그림4. 데이터베이스의 파일 테이블 구조

그림4는 데이터베이스 테이블 중에서 가장 핵심적인 파일의 메터데이터와 디렉토리 구조를 표현하기 위한 파일 테이블의 구조를 나타낸다. 각 파일과 디렉토리는 자신의 고유한 아이디(fid)를 가지고 있으며 파일의 경우 어떤 디렉토리에 포함되어 있는지를 parent 필드를 이용하여 표현한다. username 필드를 이용하여 파일의 소유자를 구분하게 되면, hash 필드는 파일 내용의 MD5 해쉬를 계산한 값을 유지하고 있어 내용이 같은 파일을 검색하기 위해 이용될 수 있다.

4. 확장된 구조

제3절의 기본 구조에서는 서버와 클라이언트만의 파일 전송만이 가능하였다. 즉 서버에서 다른 서버로 파일을 전송하려면 서버에서 클라이언트로 받은 후 다른 서버로 전송해야만 한다. 그림5는 3절의 기본 구조를 확장한 설계이다.

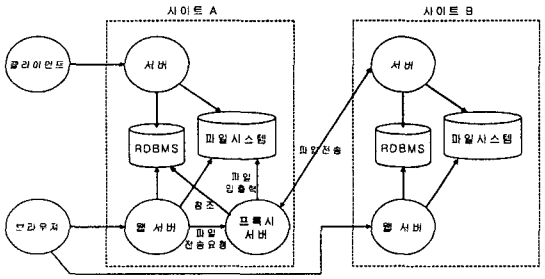

그림5. 확장된 시스템 구조

그림5가 그림1과 다른 점은 프록시 서버를 이용한다는 점이다. 프록시 서버는 웹 서버의 요청을 받아들여 다른 서버에 접속하여 파일을 전송하는 역할을 한다. 이 그림에서 웹서버는 웹 클라이언트의 요청만 처리하는 것으로 가정하였다. 따라서 간단한 요청에 대한 응답을 위해서만 웹서버가 사용되고, 파일을 전송하는 것과 같은 오랜 시간이 걸릴 수도 있는 처리는 프록시 서버를 이용하여 처리하도록 하였다. 프록시 서버는 전송할 파일의 주소를 데이터베이스로부터 읽어서 해당 서버로부터 파일을 전송받아 온다. 파일의 전송중의 진행상황은 주기적으로 데이터베이스에 반영된다.

파일 시스템에 저장된 각각의 파일들은 고유한 주소를 가지고 있으며 웹 클라이언트는 전송하고자 하는 파일의 주소를 웹 서버에게 알려줌으로써 프록시 서버를 통하여 원격지 서버의 파일을 전송 받는다.

파일의 고유 주소는 그림6와 같은 형식으로 표현된다.

gfp://서버주소:포트번호?user=사용자명&token=인증토큰&id=파일아이디{&name=파일명}

그림6. 파일의 고유 주소 표시 형식

사용자명과 인증 토큰은 사용자 인증을 위한 수단이다. 인증 토큰은 비밀번호와는 달리 사용자가 임의로 설정할 수 있으며, 대부분의 경우 유효기간이 설정되어 있어 유효기간이 지난 인증토큰은 사용을 할 수 없게 되어 있다. 파일 아이디는 해당 서버에서 파일을 식별하기 위한 고유 번호이다. 파일명은 파일의 이름으로 생략이 가능하다.

그림7은 확장된 구조에서 프록시를 이용하여 파일을 다운로드 하고 있는 화면을 보여준다. 다운로드할 파일의 URL을 입력하여 프록시 서버를 동작시키면 파일의 다운로드가 끝날 때 까지 진행상태가 웹화면으로 나타난다.

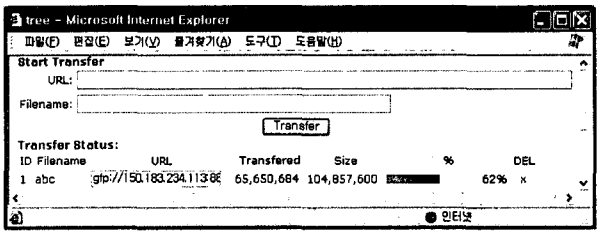

그림6. 프록시 서버를 이용한 파일 전송

5. 결론 및 향후연구

본 연구에서는 다양한 클라이언트 환경을 지원하기 위한 데이터 그리드 시스템을 설계하고 일부 구현하였다. 본 연구에서 개발된 소프트웨어를 이용하면 다양한 웹브라우져 및 자바 GUI 기반의 전용 클라이언트를 이용한 인터페이스를 통하여 파일 서버에 저장된 파일을 자유롭게 접근할 수 있다. 또한 사용자 인증 및 파일 접근시에 비밀번호가 아닌 임시 인증정보를 이용하므로 다른 사용자들과의 데이터 공유를 쉽게 한다. 또한 파일의 메터데이터를 관계형 데이터베이스에 저장하기 때문에 메터데이터의 검색과 관리가 용이하다는 장점이 있다.

향후 계획은 사용자간의 데이터의 공유와 접근 제한과 관련된 부분을 설계하고 구현할 계획이다. 파일 데이터에 대한 접근 제한 기능과 공유는 데이터 공유를 위해 꼭 필요한 기능이다.

참고자료

[1] Sheau-Yen Chen, et al. "SRB Tutorial 2002", 2002, NPACI All Hands Meeting, March 2002, http://www.sdsc.edu/srb/Tutorials/