

고성능 데이터 전송을 위한 GridFTP 성능 향상에 관한 연구

김은성^o 박형우 이상산
한국과학기술정보연구원 슈퍼컴퓨팅센터
{eskim74^o, hwpark, sslee}@kisti.re.kr

A study on the performance improvement of GridFTP for high-performance data transfer

Eunsung Kim^o Hyoungwoo Park Sangsan Lee
Supercomputing Center, Korea Institute of Science and Technology Information (KISTI)

요 약

GridFTP는 광대역의 WAN 상에 최적화된 고성능의 안전하고 신뢰할 수 있는 데이터 전송 프로토콜이다. 본 논문에서는 다음과 같은 세가지 측면에서 GridFTP의 성능 향상을 도모하였다. 1) GridFTP 서버 측에서는 성능 향상을 위해 커맨드 파이프라이닝 기법을 제시하고, 2) GridFTP 클라이언트 측에서는 데이터 채널 재사용 기법을 제안하며, 3) GridFTP 프로토콜은 third-party multitransfer 메커니즘을 지원하기 위하여 수정이 이루어졌다. 이러한 성능 향상 기법을 통하여 그리드 상에서 대규모 데이터의 효율적인 전송이 가능할 것으로 판단된다.

1. 서 론

그리드 환경에서는 테라바이트 또는 페타바이트 규모의 거대한 양의 데이터에 대한 접근을 자주 필요로 하며 또한 데이터에 대한 광범위한 접근(많은 사람에 의해서 여러 장소에서의 접근, 가상 협력 환경 등)도 필요로 한다. 이러한 사항을 만족시키기 위해서는 새로운 고성능의 안전하고 견고한 차세대 전송 프로토콜이 요구된다. GridFTP는 이를 만족시키며, 현재 가장 광범위하게 사용되는 파일 전송 프로토콜인 FTP의 기능을 확장하고 그리드 환경에 적합한 새로운 기술들을 추가하여서 구현되었다[1][2].

본 논문에서는 고성능 데이터 전송을 위한 GridFTP의 다양한 특성 및 기능에 대해서 살펴보고 GridFTP의 성능을 향상시킬 수 있는 다양한 기법들을 제시한다.

2. GridFTP

기존에 과학 기술 커뮤니티에 의해서 사용되어 온 DPSS (Distributed Parallel Storage System), HPSS (High Performance Storage System), DFS (Distributed File System)와 같은 다양한 분산 스토리지 관리 시스템들이 존재한다.

하지만 이러한 스토리지 관리 시스템들은 데이터에 접근

하기 위해서 자신들만의 특별한 프로토콜을 사용하므로 그 프로토콜을 이용하는 클라이언트만이 데이터에 접근이 가능하다. 이렇게 호환되지 않는 프로토콜의 사용은 그리드 안에서 사용 가능한 데이터들을 분할하는 결과를 초래한다. 따라서 이런 분리된 스토리지 시스템 사이의 호환성을 위한 계층으로서 공통되고 확장 가능한 데이터 전송 프로토콜이 필요하게 되었다[4].

GridFTP는 이러한 목적을 위해서 제안되었으며 다음과 같은 특징을 가진다.

- 대규모 데이터 전송
- 산업 표준 기반
- 안전한 전송
- 빠르고 효율적인 전송
- 장애에 강한 견고한 전송
- 서버 대 서버 간 데이터 전송을 위한 third-party transfer

3. GridFTP 성능 향상 기법

3.1 커맨드 파이프라이닝(Command pipelining)

현재 GridFTP는 여러 개의 파일 전송(MGET이나 MPUT 명령어 사용) 시 어떤 파일 하나에 대한 전송 관련 명령어

(RETR/ERET/STOR/ESTO)를 보낸 후 이 명령어에 대한 처리가 완료되었다는 서버의 응답을 받은 후에 다음 파일을 전송하기 위해서 다시 전송 관련 명령어를 보낸다. 이러한 방식은 기존의 FTP 프로토콜 때부터 사용되던 것으로 여러 개의 파일 전송 시 파일 하나 당 RTT (Round-Trip Time)가 소비되는 단점이 있다. 커맨드 파이프라이닝은 다수의 전송 관련 명령어를 보낼 때 최대의 효율을 얻기 위해서 이러한 명령어들을 서버 측에서 파이프라이닝(큐잉)하는 것을 말한다. 그림 1은 커맨드 파이프라이닝의 메커니즘을 보여주고 있다.

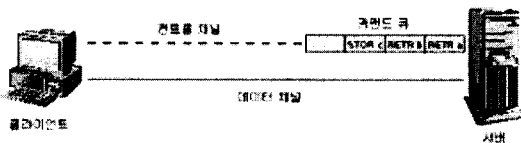


그림 1 커맨드 파이프라이닝

커맨드 파이프라이닝 기법을 사용하면 다수 파일 전송시 클라이언트는 첫 번째 파일 전송을 위해서 첫 번째 전송 관련 명령어를 보내고 이에 대한 서버의 응답을 기다린 후 다음 전송 관련 명령어를 보내는 것이 아니라, 여러 개의 전송 관련 명령어들을 연속하여 보내고 서버는 이 명령어들을 파이프라이닝하여 첫 번째 전송 관련 명령어에 대한 응답이 보내짐과 동시에 다음 전송 관련 명령어에 대한 처리가 이루어질 수 있도록 한다.

커맨드 파이프라이닝 기법은 클라이언트와 서버 사이의 다수 파일 전송 시 다음과 같은 양의 시간을 줄일 수 있다.

$$\text{단축시간} = (\text{파일 개수} - 1) * \text{RTT}$$

즉, 파일 개수가 많아질수록 더 많은 시간을 줄일 수 있다. 이것은 다수 파일 전송을 많이 요구하는 그리드 환경에서 많은 이점을 줄 것으로 판단된다.

3.2 데이터 채널 재사용(Reuse of Data channels)

GridFTP는 하나의 파일을 전송할 때마다 데이터 채널을 생성하고 삭제하는 과정을 반복한다. 이것은 클라이언트와 서버에 많은 오버헤드를 주며 데이터 채널을 생성하기 위해 주고 받는 명령어마다 RTT를 소비하게 되어서 데이터 전송 준비를 위한 지연 시간이 요구되게 된다. 데이터 채널의 재사용을 통해서 이러한 문제들을 해결할 수 있다. 그림 2는 데이터 채널의 재사용 과정을 보여주고 있다.

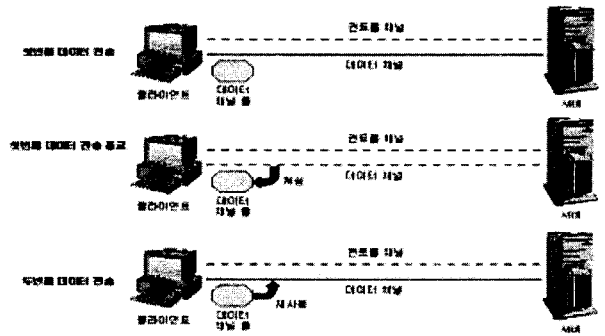


그림 2 데이터 채널의 재사용

데이터 채널을 재사용하기 위해서는 데이터 채널을 캐싱해야 한다. 데이터 채널의 캐싱은 데이터 전송 시 사용했던 데이터 채널을 데이터 채널 풀(pool)에 저장하는 방식으로 이루어진다. 캐싱된 데이터 채널은 일정 시간이 경과되면 연결이 해제되는 메커니즘과 컨트롤 채널 해제 시 같이 해제되는 메커니즘이 적용되어야 한다. 데이터 채널 재사용 기법을 통해서 다음과 같은 효과가 기대된다.

- 클라이언트와 서버 모두 데이터 채널을 생성하고 삭제하는 과정은 많은 비용을 요구하는 작업이다. 이를 제거함으로써 클라이언트와 서버 자원의 효율성을 높일 수 있다.
- 데이터 채널을 생성하기 위해서는 관련 명령어들을 주고 받아야 하고 이 때마다 RTT가 소요된다. 데이터 채널 재사용을 통해서 이러한 지연 효과 없이 데이터 전송이 가능하다.

3.3 Third-party multitransfer

Third-party transfer는 중개자 역할을 하는 클라이언트를 통해서 서버대 서버로 데이터 전송을 하는 것을 말한다. 그리드 환경에서 이 기법은 대규모 데이터를 여러 데이터 서버에 리플리케이션(replication)하기 위해서 사용되거나 그리드 포탈을 통해서 계산 작업을 수행하려는 사용자가 계산에 필요한 데이터를 각 계산 노드에 저장하기 위해서 사용되는데 현재 third-party transfer는 한 쌍의 서버대 서버간 전송만을 지원한다.

따라서 전체 노드로 데이터를 전송하기 위해서는 그림 3에서처럼 각 노드마다 중개자 역할을 하는 클라이언트와 데이터 서버 사이에 third-party transfer 세션을 연결하고

데이터를 전송하고 그 세션을 해제하는 과정을 반복해야 한다.

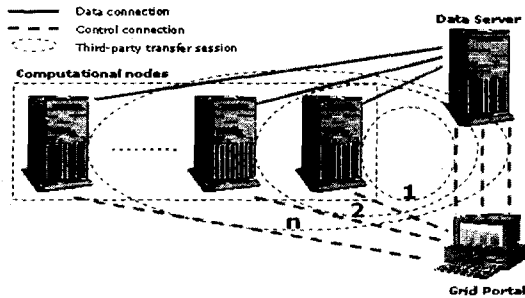


그림 3 다수의 third-party transfer 세션

결국 전체 노드로 데이터를 전송하는데 걸리는 시간은 노드 수에 비례해서 증가하게 된다. 이러한 상황은 리플리케이션과 계산을 위해서 많은 노드들을 확보할 수 있는 그리드에서는 바람직하지 못한 방법이 된다. 이러한 문제를 해결하기 위해서 Third-party multitransfer 기법을 제안한다. 이것은 데이터를 다수의 노드로 순차적으로 전송하는 것이 아니라 그림 4에서 보는 것과 같이 데이터를 각 노드로 동시에 전송하므로 하나의 세션 안에 모든 노드로 데이터 전송을 완료할 수 있다.

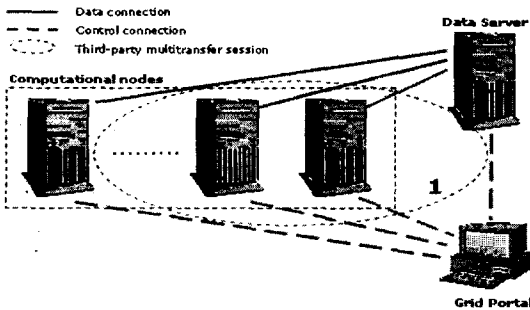


그림 4 third-party multitransfer 세션

Third-party multitransfer는 하나의 컨트롤 채널만을 생성하고 이 컨트롤 채널을 사용해서 모든 데이터 채널을 관리한다. 이를 위해서 GridFTP 프로토콜에 대한 수정이 요구되며 다음과 같은 GridFTP 커맨드의 추가가 필요하다.

* Multiple Data Port (MPOR)

이 커맨드는 중개자 역할을 하는 클라이언트를 통해서 어떤 서버로부터 다수의 서버로 데이터 전송을 하기 위해서 사용된다.

또한 다음과 같은 GridFTP feature가 추가되어야 한다.

* TPM (Third-Party Multitransfer)

이 서버는 MPAS와 MPOR 커맨드를 지원한다.

4. 결론

GridFTP는 그리드 환경에서 대규모 데이터의 효율적인 전송을 위해서 개발된 고성능의 안전하고 견고한 차세대 전송 프로토콜이다. 현재 GridFTP는 GGF(Global Grid Forum)를 통해서 표준화가 진행 중에 있으며 초안이 완성된 상태다. 2003년 상반기 안에 표준화가 될 것이라고 판단된다.

본 논문에서는 이러한 GridFTP의 특징과 다양한 기능에 대해서 자세히 살펴보았다. 또한 GridFTP의 성능을 향상시키기 위해서 데이터 채널의 재사용, 데이터 전송 관련 명령어의 파이프라이닝, third-party multitransfer 등의 기법을 통해서 GridFTP의 성능 향상을 도모했다. 추후 이러한 기능들의 구현을 통해서 성능 향상의 정도를 판단할 수 있을 것이다.

5. 참고문헌

[1] I. Foster, C. Kesselman, S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International J. Supercomputer Applications, 15(3), 2001.

[2] W. Allcock, A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke, "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets", Journal of Network and Computer Applications, 23:187-200, 2001.

[3] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, S. Tuecke, "Data Management and Transfer in High Performance Computational Grid Environments", Parallel Computing, 2002.

[4] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, S. Meder, S. Tuecke, " GridFTP Protocol Specification ", GGF GridFTP Working Group Document, September 2002.