WAN 상에서의 GridFTP 최적화 구축

Best practices for GridFTP service in WAN

최윤근, 김성준, 성진우, 이상동
한국과학기술정보연구원 연구원

Youn-Keun Choi, Sung Jun Kim, Jin Woo Sung, Sang Dong Lee
Korea Institute of Science and Technology Information

요약

초고속 파일 전송을 위한 GridFTP 서버를 구축 하고 WAN상에서 최적의 성능을 내기위한 파라미터를 테스트 한다. production 레벨에서의 서비스 시 발생할 수 있는 여러가지 문제점들을 테스트를 통해 찾아내고 실제로 상용 서비스가 가능한 구성을 제안한다.

Abstract

We build the striped and parallel gridftp server on WAN and test to find the best parameters for high performance. Also we check the various problems about gridftp service as production level and propose a best configuration for real service environment.

I. 서 론

1. 개요

GridFTP[1]는 고성능의 안전하고 견고한 데이터 전송 메커니즘을 가지며, 특히 데이터 전송의 써드파티 제어, 병렬 데이터 전송, 스트라이핑 데이터 전송, 부분 파일 전송자동, TCP 버퍼/윈도우 크기 협상, 신뢰적이고 재시작 가능한 데이터 전송 등 대용량 데이터 전송에 적합한 기능을 포함하고 있다.[2]

1.1 데이터 전송의 써드파티 제어

써드파티 제어는 한 사이트에 있는 사용자나 애플리케이션이 다른 두 사이트 사이의 데이터 전송을 시작시키고 전송 상황을 모니터링하고 제어할 수 있게 한다.

1.2 병렬 데이터 전송

WAN에서 병렬로 다수의 TCP 스트림을 사용하는 것은 하나의 TCP 스트림을 사용하는 것보다 전체 대역폭을 향상시킬 수 있다. GridFTP는 FTP 커맨드 확장과 데이터 채널 확장을 통해 병렬 데이터 전송을 지원한다.

1.3 스트라이핑 데이터 전송

GridFTP는 같은 데이터가 여러 서버에 분리돼 저장되어 있는 경우 이 데이터를 전송하기 위해 각 서버로 다수의 TCP 스트림을 열어 사용하는 스트라이핑 전송을 지원한다. 스트라이핑 전송은 병렬 전송보다 더욱 전체 대역폭을 향상시킬 수 있다.

1.3 부분 파일 전송

어떤 애플리케이션은 전체 파일보다는 파일의 일부분만을 필요로 할 수 있다. GridFTP에서는 이러한 기능을 더욱 확장해 파일의 임의의 영역을 전송하도록 하는 커맨드를 제공한다.

1.4 자동 TCP 버퍼/윈도우 크기 협상

TCP 버퍼/윈도우 크기를 최적으로 설정한다면 데이터 전송 성능을 크게 향상시킬 수 있다. GridFTP는 커다란 파일과 대규모의 작은 파일들에 대한 수동 및 자동 TCP 버퍼 크기 협상을 지원한다.

1.5 신뢰적이고 재시작 가능한 데이터 전송 지원

FTP 표준은 실패한 전송을 재시작하는 기능을 지원하지만 많이 이용되고 있지는 않다. GridFTP는 FTP의 이러한 기능을 이용하며 새로운 데이터 채널 프로토콜을 지원하기 위해 이를 확장했다.

즉 GridFTP는 WAN 에서의 고속, 대용량 파일 전송에 적합하게 FTP 프로포콜을 확방하여 개발되었으며 CERN과 같은 Data-intensive한 연구를 가는 센터등에서 광범위하게 활용되고 있다.

2. 연구 방향

초고속 파일 전송을 위한 GridFTP 서버를 구축 하고 WAN상에서 최적의 성능을 내기위한 파라미터를 테스트 한다. production 레벨에서의 서비스 시 발생할 수 있는 여러가지

문제점들을 테스트를 통해 찾아내고 실제로 상용 서비스가 가능한 구성을 제안한다.

II. 배 경

1. TCP 튜닝

TCP 성능은 가용한 대역폭 뿐만 아니라 OS에서 설정된 TCP 윈도우 크기에 따라 달라진다[3]. BDP는 네트워크 링크를 채울 수 있는 데이터의 양을 결정한다.

TCP에 대한 기본 적인 튜닝은 TCP 윈도우 크기를 조절하는 것이다. 만일 값이 너무 작을 경우 sender는 매번 idle 상태가 되고 낮은 성능을 나타낼 것이다. TCP 윈도우 크기에 사용되는 이론적인 값은 BDP(bandwidth delay product)이다.

고성능 BDP 네트워크상에서 TCP connection의 전송률을 최대화하기 위해서는 TCP 버퍼가 적어도 BDP 크기만큼은 확보되어야 한다.

BDP는 다음과 같이 계산된다.

BDP = 대역폭(Bandwidth)(MB/s) × RTT (seconds)

예를 들어 1Gigabit ethernet에서 10ms의 RTT였다면
= (1,024 / 8) × (10 / 1000)
= 1.28MB (1,242,177.28B)

이 값은 최상의 윈도의 크기를 결정하는 시작점이 된다. 이 값을 기준으로 높거나 낮게 값을 셋팅하면서 최적의 값은 구하면 된다. 예를 들어 BPD 값이 1.23MB임에도 1MB 이상에서 더 이상의 성능 증가가 없을 수도 있다.

대부분의 OS에서는 TCP send 와 receive의 버퍼 크기를 기본 값으로 설정해 놓고 있다, 이것은 아마도 100Mbps 성능의 기본적인 로컬 네트워크를 지원하기 위해서 설정해 놓은 것으로 보인다. 그러나 이것은 높은 대역폭과 낮은 지연율을 가진 네트워크에서는 상당히 비효율적이다. 그래서 버퍼 크기를 조절하는 파라미터의 튜닝은 최적의 throughput을 위해 필수적인 작업이다. 최적의 window 크기는 링크된 대역폭과 지연시간을 고려하여 계산되고 커널에 있는 TCP stack의 파라미터가 조정되어야 한다.

2. 병렬 스트림(parallel stream)

보통 1개의 TCP connection이 sender와 receiver 간에 만들어 지지만 다중 소켓에 다중 connection이 open 될 수 있다. 즉 보내질 데이터가 n 개로 나누어지고 n 개의 소켓을 통해 n 개의 connection이 만들어 진다, receiver는 이런 데이터 파티션들을 재결합 한다. 병렬 소켓은 커널 파라미터 변경 없이 효과적으로 버퍼 크기를 늘릴 수 있다.

III. 테스트

본 테스트는 GridFTP를 이용하여 WAN 상에서 두 사이트 간의 3rd-party, parallel transfer를 수행한다. 최적화된 성능을 위해 이론적인 값은 기준으로 TCP 윈도의 크기 및 병렬화 등의 값을 조절하며 각각의 성능을 비교한다.

1. 구성 환경

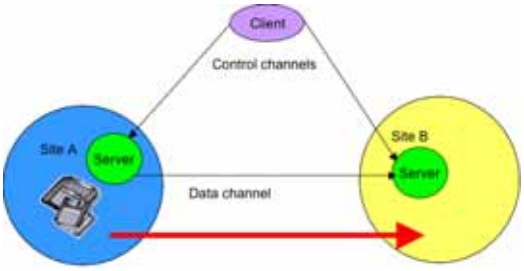
PNU(부산대)와 KISTI 간 네트워크는 dedicated 1Gbps로 연결되어 있다. 양 사이트에 각각 GridFTP 서버 1대씩이 설치되어 있고 세부 구성은 다음과 같다.

이름	사양	MTU	Bonding
kgfs	Intel xeon 1.6G 2코어 3G memory	9,000	no
pgfs	Intel xeon 2.3G, 4코어 2G memory	9,000	2
x4500	AMD opteron 2.6G 4 코어 16G memory	9,000	4

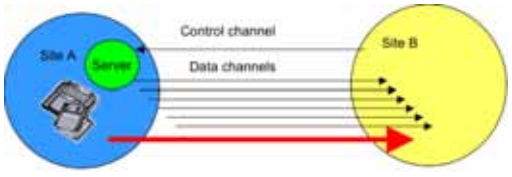
kgfs와 x4500은 KISTI내에 있는 서버이고 pgfs는 부산대에 설치되어 WAN을 통해 kgfs와 x4500 서버와 연결되어 있다. 물리적으로 세 서버는 각각 다른 곳에 위치하나 전용 네트워크의 구성으로 논리적으로는 같은 도메인 내에 존재한다.

각 서버의 이더넷 디바이스는 1개, 2개 4개가 있으며 2개 이상의 이더넷 디바이스는 bonding(여러 개의 디바이스를 논리적으로 1개로 구성)되어 있고 모두 MTU 9000인 Jumbo Frame이 설정되어 있다. TCP 윈도의 크기는 최대 125MB까지 설정해 놓았다.

본 환경의 특징은 3rd-party와 striped transfer가 가능하도록 구성되어 있다.



▶▶ 그림 1. 3rd party transfer



▶▶ 그림 2. 병렬 데이터 전송

GridFTP 클라이언트로 globus-url-copy를 사용하였다. 이 명령어의 중요 옵션은 다음과 같다.

-p (병렬 또는 스트림 수) : 보통 4-8사이의 값이며 4부터 시작

-tcp-bs (TCP 버퍼 크기) : 이 값은 두 호스트간의 BW와 RTT값으로 결정된다. 두 호스트 간의 RTT를 결정하기 위해서 ping이나 traceroute를 사용한다.

buffer size = BW(Mbs) * RTT(ms)

-vb : 성능에 대한 피드백을 보고 싶을 때 사용

-dbg : 디버깅 옵션

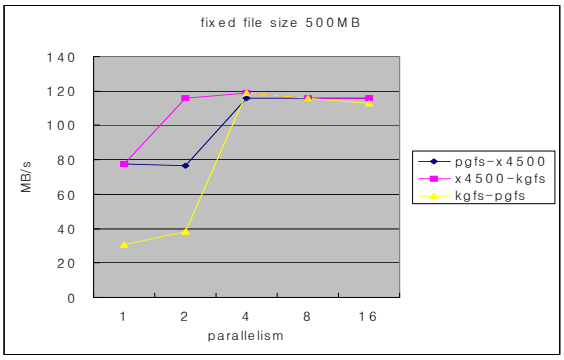
GridFTP server는 2.5버젼이면 GT4.0.3에 포함되어 있다.

2. 테스트 방법 및 결과

2.1 싱글 스트림과 병렬 스트림의 비교

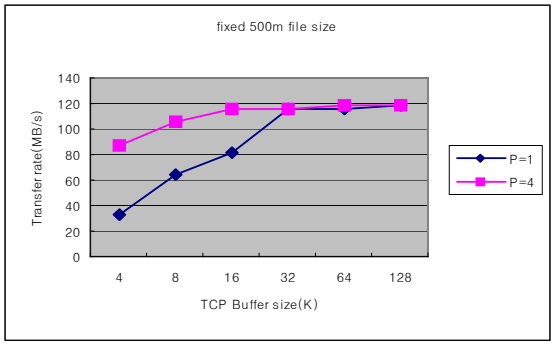
고정된 파일 크기(500M)에 파라미터 -p 값(1. 2. 4. 8, 16)을 변경하면서 transfer rate 성능을 비교한다.

성능 측정은 kgfs↔pgfs, pgfs↔x4500, x4500↔kgfs간의 3rd-party transfer 방식으로 파일을 전송한 다음 이에 대한 성능을 비교했다.



2.2 TCP 버터 크기별 성능 비교

고정된 -p 값에 파라미터 -tcp-bs 값(4, 8, 16. 32, 128k)을 변경하면서 transfer rate 성능을 비교한다.



IV. 결 론

본 테스트를 통해서 나온 결론은 다음과 같다.

- 적절한 TCP 튜닝이 최대 성능에 도달하는 중요한 요소이며 병렬 스트림 또한 튜닝된 단일 스트림보다 25% 이상 성능 향상을 가져온다
- 튜닝되지 않은 TCP의 병렬 스트림은 튜닝된 버터의 TCP 성능과 거의 같아 질 수 있다.
- 데이터 전송동안 동적으로 버퍼 크기가 변하는 메카니즘 필요

위의 결론을 바탕으로 데이터의 고속 전송을 위한 GridFTP 성능 최적화 방법은 다음과 같다.

- 두 호스트 간의 RTT 측정
- 이를 바탕으로 BDP 계산
- TCP 버퍼 크기 및 병렬 스트림 수를 조정하며 전송률 비교 측정
- 최적화된 옵션값 제시

▌참 고 문 헌▐

[1] GSIFTP http://www.globus.org/gsiftp-alpha
[2] Globus Porject. http://www.globus.org
[3] TCP Tuning Guide for Distributed Application on Wide Area Networks, http://www-didc.lbl.gov/tcp-wan.html