

병렬 소켓 간의 트래픽 간섭 현상 분석 및 대역폭 예측

김영신*, 허의남*, 황 준*
*서울여자대학교 미디어학부
e-mail: huh@swu.ac.kr

Analysis of the Interference between Parallel Socket Connections and Predictions of the Bandwidth

Young-sin Kim*, Eui-Nam Huh*, Jun Hwang*
*Division of Media, Seoul Women's University

요 약

최근에는 대량의 데이터가 생산되고, 저장되며, 전송되고 있다. 따라서 이러한 대량의 데이터를 보다 효율적으로 관리하는 기술, 데이터를 고속으로 전송하는 기술이 연구되고 있다. 그 중, GridFTP는 고속으로 데이터를 전송하기 위해 연구되고 있는 기술로, 병렬연결들을 생성하고 이들을 통해 동시에 데이터를 전송함으로써 보다 빠른 전송을 실현하고 있다. 그러나 그리드 환경의 VO 간에 데이터를 병렬로 전송하는 응용 프로그램들이 증가하게 된다면, VO 간에 소켓 연결 수가 증가하게 되고, 따라서 각 연결들 간에 간섭이 발생하게 된다. 따라서 본 논문에서는 소켓 연결들 간의 간섭을 실험을 통하여 분석하고, 한 응용 프로그램에 의해 생성되는 병렬연결이 할당받게 될 대역폭을 예측함으로써, VO 간의 병렬 전송을 제어하는 메커니즘의 토대를 마련하고자 한다.

1. 서론

최근 들어, 자연 과학 분야 등에서 생성되는 대량의 데이터를 하나의 스토리지에 저장한다는 것은 불가능한 일 이므로, 이러한 대량의 데이터를 저장하기 위하여 지역적으로 분산된 여러 스토리지를 이용하며, 필요한 시점에 데이터에 접근하는 환경을 구성하여 활용하고 있다. 그러나 대부분의 분산된 스토리지를 관리하는 시스템은 서로 다른 관리 시스템 간에 호환될 수 없다는 문제점을 가지고 있다. 데이터 그리드는 이러한 문제점을 해결하고자 연구되고 있으며, 전 세계 통합 데이터 관리를 목적으로, 지역적으로 분산된 데이터를 통합, 관리, 분석할 수 있는 환경을 제공하고 있다[1]. 이러한 데이터 그리드는 분산된 대량의 데이터를 광대역 네트워크 환경에서 보다 안전하고 효율적으로 전송하기 위해 GridFTP를 이용하고 있다.

GridFTP는 고성능의 데이터 전송을 위해 연구되고 있는 기술(TCP tuning 등)들 중 하나로, 데이터 병렬 전송 기술을 사용하고 있다[2]. 그러나 그리드 환경에서 VO 간에 데이터를 병렬로 전송하는 응용 프로그램들이 증가하게 되면, VO 간에 소켓 연결의 수가 증가하게 되고, 따라

서 각 소켓 연결들 간에 간섭이 발생하게 된다.

따라서 본 논문에서는 소켓 연결들 간의 간섭을 실험을 통하여 분석하였으며, 한 응용 프로그램이 병렬연결을 수행함으로써 할당 받게 될 대역폭을 예측하였다. 이로 인해, VO 간의 병렬 전송 제어 메커니즘 연구의 토대를 마련하고자 한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 병렬 전송 메커니즘에 대해 설명하고, 3장에서는 병렬 전송의 간섭 현상 실험을 통하여 분석하였다. 4장에서는 분석 내용을 바탕으로 응용프로그램에 할당될 대역폭을 예측하였으며, 5장에서는 결론과 향후 연구 과제에 대해 논한다.

2. 관련연구

2.1 데이터 전송 성능 향상을 위한 기술

(1) 커널 수준의 제어 알고리즘 변경[5]

TCP 혼잡 제어(Congestion Control) 또는 흐름제어(Flow Control) 알고리즘을 수정함으로써 데이터 전송 성능을 향상시키고자 하는 방법이다. 예를 들어, slow-start

단계에서 ssthresh 값을 높게 설정한다든지, 혼잡 회피 단계에서 CWND(Congestion Window Size)의 증가량을 달리하여 전송되는 데이터의 양을 증가시키는 방법을 일컫는다.

그러나 이 방법은 커널 레벨의 소스를 수정해야 하기 때문에 다른 시스템의 커널과 호환되지 않는다는 문제점을 가지고 있으며, 복잡한 알고리즘을 수정해야 하기 때문에 수정이 어렵다는 단점을 가진다.

(2) TCP buffer tuning[5]

고성능의 원거리 네트워크에서 TCP 버퍼 크기를 적당하게 조절함으로써 데이터 전송 성능을 향상시키는 기술이다. 이 기술은 TCP를 수정하지 않는다는 장점을 가지고 있으나, 반면에 한번 조절된 TCP 버퍼의 크기는 고정되어 있으므로 네트워크 상태가 변화가 심할 경우에는 성능 향상을 기대하기 어렵다는 단점을 가진다. 또한, TCP 버퍼 크기를 조절하기 위해서는 전문지식을 갖춘 전문가를 필요로 하는 문제점을 안고 있다. 이러한 문제점을 해결하기 위해 최근에는 여러 가지 Auto TCP Buffer Tuning 기법이 제안되고 있다.

(3) P.Sockets[6]

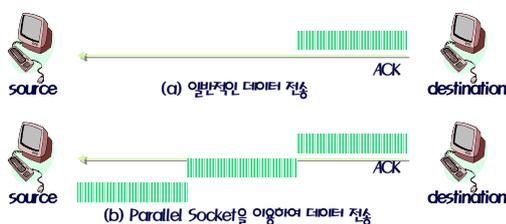
P.sockets은 C++ 라이브러리로, 송신측과 수신측 간에 병렬 소켓 연결을 설정하고, 데이터를 설정된 연결의 개수만큼 나누어 동시에 각각의 연결로 전송하는 기술이다.

이 기술은 TCP tuning의 단점을 보완함과 동시에 고성능의 원거리 네트워크에서 최대 성능을 발휘할 수 있기 때문에 데이터 그리드의 핵심 기술인 GridFTP 에서도 이용하고 있다.

그러나 여러 응용 프로그램이 P.sockets를 이용하여 병렬로 데이터를 전송할 경우 이들 간에 간섭이 발생하게 된다. 따라서 더 많은 병렬 소켓 연결들로 데이터를 전송하는 응용 프로그램이 네트워크 대역폭을 대부분 점유하는 문제가 발생하게 된다.

2.2 Parallel Sockets (P.Sockets)

그림 1은 Parallel Sockets(P.Sockets)가 전송 능력이 향상되는 원리를 나타내고 있다.



(그림 1) Parallel Sockets

그림 1의 (a)는 기본 TCP buffer size에서 단일 소켓 연결을 이용하여 두 호스트 간에 데이터가 전송되는 상태를 나타내고 있으며, (b)는 병렬 소켓 연결을 이용하여 데이터가 전송되는 상태를 보여주고 있다[4].

TCP는 전송한 데이터의 ack를 받기 전까지 다음 데이터를 전송하지 않기 때문에 그림 1의 (a)와 같이 네트워크

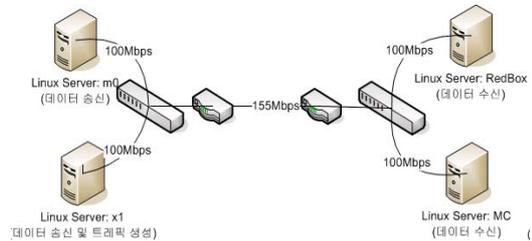
대역폭을 제대로 활용하지 못하게 된다. 이에 반해 그림 1의 (b)를 보면, 비록 TCP buffer를 tuning 하지 않더라도 세 개의 소켓 연결을 이용하여 데이터를 전송함으로써 네트워크 대역폭을 효율적으로 사용하고 있음을 보여준다.

3. 실험

3.1 실험 환경

KOREN 155M 백본 망과 100Mbps fast ethernet을 사용하는 네트워크에 4대의 컴퓨터를 연결하고, 아래 그림3과 같은 환경을 구성하였다.

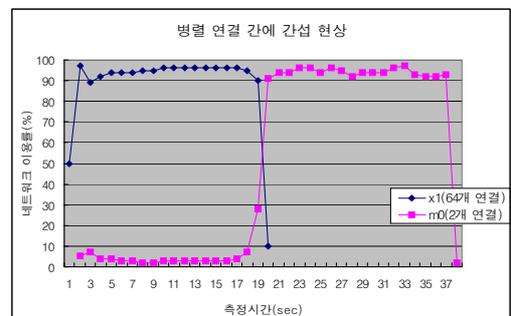
m0와 x1은 A지역에 위치해 있으며, RedBox와 MC는 B지역에 위치해 있다. A지역과 B지역 간의 거리는 약 50Km이다. B지역에 위치해 있는 호스트, RedBox와 MC의 성능은 Pentium III 800MHz 이며, A지역에 위치해 있는 두 호스트, m0와 x1의 성능은 Pentium III 733MHz 이다.



(그림 3) 실험 환경

3.2 병렬 스트림 간의 간섭 현상

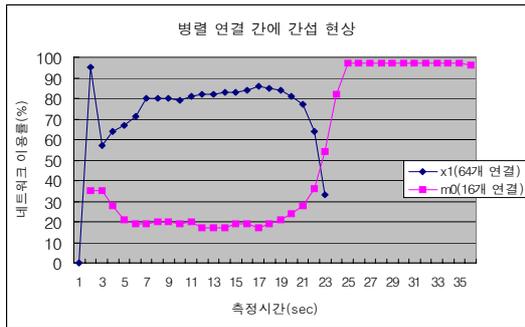
응용 프로그램에서 생성한 병렬 소켓 연결들로 데이터를 전송할 경우, 이들의 간섭현상을 분석하기 위해 x1에서 MC로 200MB의 데이터를 병렬로 전송시키고, 데이터가 전송되는 도중에 m0에서 RedBox로 병렬 소켓 연결을 설정하고 200MB 데이터를 전송하였다. 각 실험마다 x1은 SC64(64개의 소켓 연결)로 데이터를 전송시켰고, m0은 SC2(2개의 소켓 연결)에서 SC64로 연결 수를 변경시키며 데이터를 전송하였다. 이러한 과정에서 스니퍼를 이용하여 각 응용 프로그램들이 사용하는 네트워크 이용률을 측정하였다. 그림 4-7은 실험 결과들 중 일부를 나타내고 있다.



(그림 4) 두 호스트가 각각 SC64와 SC2로 데이터 전송 시 사용하는 네트워크 이용률

그림 4의 그래프는 m0에서 SC2로 데이터를 전송할 경우, x1의 병렬 소켓 연결들에 미치는 영향을 알아보기 위해 실험한 결과를 나타내고 있다. 본 실험에서 x1은 네트

워크의 약 96%를 이용하였고, m0는 약 3% 정도를 이용하는 것으로 나타났다. 두 호스트가 데이터 전송을 시작한 후 약 20초 동안 x1의 응용프로그램은 높은 이용률을 기록하는 상태를 볼 수 있다. 그러나 그에 반해 SC2를 이용하는 m0의 응용 프로그램은 네트워크를 거의 이용하고 있지 못하고 있음을 알 수 있다. 이는 x1이 m0의 소켓 연결들에게서 간섭을 거의 받고 있지 않음을 알 수 있다. 그림 4의 그래프에서 약 20초 이후부터는 m0의 네트워크 이용률이 평균 92%로 기록되고 있는데, 이는 x1이 전송을 마쳤기 때문이다.



〈그림 5〉 두 호스트가 각각 SC64와 SC16로 데이터 전송 시 사용하는 네트워크 사용률

그림 5의 그래프는 m0에서 SC16으로 데이터를 전송할 경우 x1의 데이터 전송에 미치는 영향을 알아보기 위해 실험한 결과를 나타내고 있다. 데이터 전송 시작부터 x1의 전송이 끝나는 24초 지점까지 x1에서 사용하는 네트워크 이용률은 평균 76%를 기록했으며, m0에서는 평균 28%를 기록하였다. 그림 4의 x1이 네트워크의 약 96%를 이용했던 것에 비하면, 이용률이 대폭 축소된 것을 알 수 있다. 이는 m0의 SC16에 영향을 받고 있음을 나타낸다.

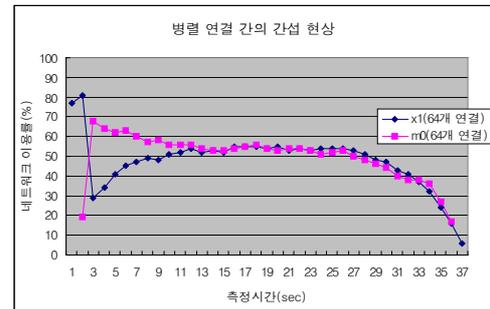
그림 5의 24초 지점에서 두 호스트의 네트워크 이용률이 서로 교차되는 현상을 볼 수 있다. 두 그래프가 곡선을 이루면서 증감하는 현상이 나타나는데, 이는 데이터 전송을 위해 생성된 쓰레드들이 시간차를 두고 종료하기 때문에 나타나는 현상이라 분석할 수 있다. 즉, x1에서 쓰레드가 순차적으로 종료되면서 확보했던 네트워크 대역폭을 반환하게 되고, m0은 반환된 네트워크 대역폭을 이용함으로써 이용률이 증가하게 된다.

그림 6의 실험에서 x1의 네트워크 이용률은 평균 48%로, m0의 네트워크 이용률은 약 50%로 측정되었다. 즉, 두 호스트에서 같은 개수의 소켓 연결을 이용함으로써 네트워크의 대역폭이 1/2씩 이용되었다.

실험 결과를 종합해 보면, x1과 m0가 SC64와 SC2를 이용할 경우 네트워크 이용률은 평균 96%와 3%로 측정되었고, SC64와 SC16의 경우에는 평균 76%와 28%, SC64와 SC32의 경우에는 58%와 40%, SC64와 SC64의 경우에는 평균 48%와 50%의 네트워크 이용률이 측정되었다.

이러한 결과는 사용 중인 소켓 연결의 총 개수 중에 현재 호스트에서 사용하는 소켓 연결의 개수의 비율로 네

트워크를 이용하고 있음을 알 수 있다. 따라서 네트워크의 대역폭은 생성된 소켓 연결 간에 공평하게 나누어 사용되는 것으로 이번 실험을 통하여 증명되었다.



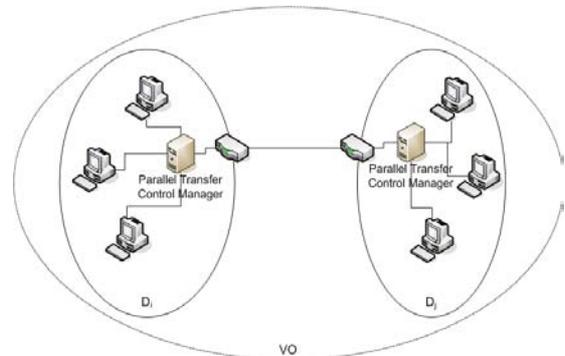
〈그림 6〉 두 호스트가 각각 SC64와 SC64로 데이터 전송 시 사용하는 네트워크 사용률

4. 대역폭 할당 값 예측 모델링

그리드에서는 기관과의 자원 공유를 위해 새로운 가상 조직이 구성되는데, 이를 VO (Virtual Organization) 라 하며, PKI 기반의 상호 보안 협약이 이루어짐으로써 안전이 보장된다.

그림 7은 VO 단위에서 병렬로 데이터를 전송할 때, 도메인 간의 연결 관리 시스템 구조를 나타내고 있다. VO에는 도메인 i와 또 다른 도메인 j가 존재하고, 두 도메인 간에는 병렬연결을 관리하는 Parallel Transfer Control Manager(PTCM)가 존재한다. 각 도메인 안에는 여러 호스트가 존재하며, 각 호스트에는 병렬 전송을 지원하는 응용 프로그램이 실행된다고 가정한다.

한 호스트가 다른 도메인의 한 호스트에 데이터를 전송하고자 할 때, 송신측 호스트는 데이터 전송에 필요한 대역폭을 PTCM에게 신청하고, PTCM는 서비스 레벨을 고려하여 몇 개의 소켓 연결을 설정해야 하는지 등에 대한 정보를 송신 호스트에 알려준다. 송신 호스트는 수신한 설정 정보를 이용하여, 수신측과 연결을 설정하고 데이터를 전송한다. 데이터 전송이 완료되면, 설정된 연결의 종료를 PTCM에게 통보하고, PTCM은 병렬 전송에 관한 정보를 갱신한다.



〈그림 7〉 데이터 병렬 전송을 위한 연결 관리 시스템 구조

호스트 간에 설정된 연결들은 네트워크 대역폭을 $1/n$ 로 공평하게 나누어 할당 받는 특성을 가지므로 새로운 연결이 생성되면, 기존의 연결들이 사용할 수 있는 대역폭이 감소하는 현상이 나타난다. 이는 3장의 실험을 통해서도 증명되었다. 따라서 새로운 연결이 설정되면, 연결간의 간섭을 예측하여 기존의 연결들이 사용할 수 있는 대역폭을 예측할 수 있다.

도메인 $i(D_i)$ 와 도메인 $j(D_j)$ 에서 현재 데이터 전송을 위해 실행중인 응용프로그램(t)의 소켓 연결 수를 $A_{t_{ij}}$ 라 하면, D_i 와 D_j 간에 전송되는 모든 소켓 연결 수의 합, V_k 는 다음 식 ①과 같이 표현할 수 있다.

$$V_k = \sum_{\forall t} A_{t_{ij}} \dots \text{①}$$

이때, D_i 와 D_j 간에 응용 프로그램(n)에서 새로 생성될 소켓 연결의 수를 $A_{n_{ij}}$ 라 하면, 새로 생성될 소켓 연결들이 할당받을 수 있는 대역폭(Bandwidth for New Allocation :BNA)은 식 ②와 같이 계산될 수 있다.

$$BNA = B \times \frac{A_{n_{ij}}}{V_k + A_{n_{ij}}} \dots \text{②}$$

(B : D_i 에서 D_j 까지의 네트워크 대역폭)

따라서 식②를 이용하여 응용프로그램에서 요구하는 대역폭을 획득하기 위해 연결의 수를 다음 식 ③과 같이 결정할 수 있다.

$$A_{n_{ij}} = \frac{V_k \times BNA}{B - BNA} \dots \text{③}$$

기존의 연결들이 새로운 연결에 의해 간섭을 받게 되므로, 기존 연결들이 이용하던 대역폭은 감소하게 된다. 이런 상황에서 기존 연결의 서비스 레벨에 따라 대역폭을 보유해야 하는 상황이 발생할 수도 있다. 따라서 소켓 연결의 서비스 레벨에 따라 대역폭을 할당하기 위해서는 새로운 연결의 생성된 이후, 기존 연결들이 사용하게 될 대역폭을 파악할 필요가 있다.

기존의 모든 응용 프로그램들 사용할 수 있는 대역폭 예측값(Interfered Bandwidth:IB)은 다음 식 ④로 계산될 수 있다.

$$IB = B \times \frac{V_k}{V_k + A_{n_{ij}}} \dots \text{④}$$

그러므로 기존에 데이터를 전송하던 한 응용 프로그램이 새로운 연결이 발생한 이후 사용할 수 있는 대역폭은 식 ⑤로 얻을 수 있다.

$$AB(A_{t_{ij}}) = IB \times \frac{A_{t_{ij}}}{V_k} \dots \text{⑤}$$

다음 그림 8은 m_0 에서 새로운 연결의 생성 후, x_1 의 기존 연결들과 m_0 의 새로운 연결들이 사용하는 대역폭을 위의 식 ②와 ④를 이용하여 구한 예측값과 비교한 그래

프이다. x_1 과 m_0 의 막대 그래프는 그림 4-6의 각 실험에서 두 호스트가 데이터를 동시에 전송하는 동안 측정된 결과를 평균한 값이다. 선형 그래프는 새로운 연결과 기존 연결의 예측된 대역폭이다.



〈그림 8〉 실제 사용 대역폭과 예측 대역폭의 비교

그림 8에서 x_1 이나 m_0 의 실제 측정된 대역폭과 예측값 BNA와 IB가 거의 일치하고 있음을 알 수 있다. 따라서 호스트 간의 연결의 수를 통하여 대역폭의 제어가 가능함을 알 수 있다.

5. 결론 및 향후 과제

본 논문에서는 TCP 버퍼를 튜닝하거나, TCP 제어 알고리즘을 수정하는 등의 커널 레벨의 작업이 이루어지지 않더라도 데이터 전송 능력을 향상시킬 수 있는 병렬 전송기술을 알아보았다.

또한, 병렬 전송 시 소켓 연결들 서로 간에 간섭 현상이 발생함을 확인하고, 이러한 현상을 분석하였다. 이를 토대로 새로운 연결이 데이터 전송 중인 연결들에 어느 정도 영향을 미치는지를 예측할 수 있도록 모델링을 하였다. 또한 원하는 대역폭을 획득하기 위해 필요한 연결의 개수를 예측할 수 있었다. 즉, 소켓 연결들의 개수를 이용하여 원하는 네트워크 대역폭을 확보할 수 있음을 알 수 있었다.

따라서 앞으로 본 연구 결과를 이용하여 VO 간의 데이터 병렬 전송을 관리할 수 있는 메커니즘 연구가 가능하다.

참고문헌

- [1] The Data Grid: Towards an Architecture for the Distributed Management. and Analysis of Large Scientific Datasets. Ann Chervenak, Ian Foster, C.Kesselman, C.Salisbury, S.Tuecke, Journal of Network and Computer Application, 23, 187-200, 2001
- [2] GridFTP v2 Protocol Description, Bill Allcock, Igor Mandrichenko, Timur Perelmutov, Germi National Accelerator Laboratory, 2004
- [3] GridFTP: Protocol Extensions to FTP for the Grid, W. Allcock, Argonne National Lab.
- [4] GridFTPexplained, Pawel Plaszczak, Joe Link, Rich Wellner, Paul Hubbard, www.icslab.agh.edu.pl/~kz-ajac/pawel/GridFTPexplained.doc
- [5] An Efficient TCP Buffer Tuning Technique Based on Packet Loss Ratio (TBT-PLR), Gi-chul Yoo, Eun-sook Sim, Dongkyun Kim, Taeyoung Byun, Kookhan Kim, Okh-wan Byun, International Conference on Internet Computing, 2004
- [6] Psockets: The Case for Application-level Network Striping for Data Intensive Applications using High Speed Wide Area Networks, H.Sivakumar, S.Bailey, R.L.Grossman, Proceedings of the IEEE/ACM SC-2000 Conference November, 2000