MPTCP기반 Globus 서비스 적용을 위한 병렬 전송성능 모니터링

홍원택*

한국과학기술정보연구원

Monitoring of Parallel Transfer Performance for MPTCP-based Globus Service

Wontaek Hong*

Korea Institute of Science and Technology Information

E-mail : wthong@kisti.re.kr

요 약

대용량 데이터의 공유 및 신속한 전송을 요구하는 과학응용 분야에서는 전송 성능을 높이기 위해 Globus/GridFTP와 같은 전송 어플리케이션에서의 병행성, 병렬성, 파이프라이닝 기법 등을 통한 성능 향상을 추구해 오고 있다. 본 논문에서는 비슷한 맥락에서 전송 프로토콜 계층에서 다중 경로 지원할 수 있는 Mptcp를 도입할 경우 기대할 수 있는 인터페이스 수의 증가 및 전송 스트림들의 병렬성 확장에 따른 전송성능 향상을 에뮬레이션 환경에서 실험하고 결과를 제시한다.

ABSTRACT

For science applications that requires rapid transfer and sharing of large volume data, many efforts to improve data transfer performance have been made based on concurrency, parallelism and pipelining in data transfer applications such as Globus/GridFTP. In this paper, as a similar trial, experiments have been conducted for the expected transfer throughput enhancement by the increased number of network interface and parallelism in the Mptcp emulation environment and the result is presented.

키워드

Globus, parallelism, mptcp, emulation

Ⅰ. 서 론

기상기후, 고에너지물리, 천문학 등과 같은 과학응용 협업연구를 수행하는 연구자들에게 있어서 과학 빅데이터를 전송 및 공유하기 위한 고품질의 네트워크 확보 및 전송 환경의 구축은 연구 업무 효율을 높이기 위해 선결되어야 한다. 전 세계적으로 R&E 네트워크 커뮤니티에서 폭넓게 적용되는 Science DMZ 개념[1]도 이러한 요구 사항을 수용하기 위해 제안되었고, 대용량 데이터의 고속 전송과 관련해서 Globus와 같은 전송 플랫폼이 꾸준히 활용되고 있다. 특히, 병렬 파일시스템을 제공하는 HPC 분야에서는 대용량 데이터의 고속 전송을 위해서 전용의 데이터 전송 노드 내에서 전송 프로세스의 병행성, 전송 스트림의 병렬성 향상을 추구해 오고 있다.

이와 관련하여 본 논문에서는 Globus/GridFTP 서비스가 일반 Tcp가 아닌 다수의 네트워크 인터페이스를 지원하는 Mptcp 환경으로 변화되었을 때, 유휴 NIC을 이용하여 성능을 증가시킬 수 있다는 것을 보임과 동시에 일반 tcp 환경 대비 mptcp 환경에서 tcp 스트림에 대한 병렬성이 증가할 때 어느 정도의 성능 향상을 도모할 수 있는지 에뮬레이션 환경에서 실험하고 결과를 제시한다.

* corresponding author

II. 본 론

과학응용 협업연구 분야에서 데이터의 전송 및 공유를 위한 도구로서 가장 활발히 이용되고 있는 Globus online 서비스는 서버 급 머신에서 활용되는 Globus Connect Server (GCS)와 경량급 머신에서 활용되는 Globus Connect Personal (GCP)로 구성되어 해당 머신을 종단 포인트화 한다. 이러한 환경에서 해당 서비스의 종단 이용자는 본인의 머신으로 원하는 데이터를 가져오기 위해 GCS와 GCP간의 전송을 수행하게 된다. 캠퍼스 네트워크 또는 개별 연구소의 실험실에서의 이용자들은 많은 경우 Last 1 mile problem으로 인해 전송 성능의 저하를 경험하게 된다[1]. [2]에 따르면 실제적으로 GCS와 GCP간의 전송은 GCS와 GCS간의 전송과 함께 Globus online 서비스에서의 전체 전송 횟수 측면에서 “dominant” 서비스를 차지한다. 따라서 이 구간의 성능 향상을 추구하는 것은 국가과학기술연구망과 같은 R&E 네트워크 커뮤니티의 사용자들의 서비스 체감 지수를 높일 수 있는 중요한 시도에 해당한다.

Science DMZ 환경에서 GCS와 GCS간 전송 효율을 높이기 위해서는 GCS에 탑재된 GridFTP 프로토콜에서 제공하는 병렬성, 병행성, 파이프라이닝 등의 기법을 통해 성능 향상을 도모한다. 또한, Science DMZ에서의 데이터 전송 노드(DTN)을 Luster, GPFS와 같은 병렬 파일시스템에 대해 고속 마운트시켜 전송 성능을 향상시키고, 더 나아가 다수의 DTN 노드들을 클러스터링화 하여 병렬성과 병행성을 공격적으로 확장시키는 기법을 이용하기도 한다[3].

한편, 위의 시도들이 GridFTP에 기반한 전송 어플리케이션 계층에서의 전송 성능을 향상키는 방법이라면, 본 논문에서는 전송 프로토콜 계층에서 기존 Tcp를 대신하여 Mptcp를 이용할 경우의 GCS와 GCP간의 병렬전송 성능 향상에 초점을 둔다.

Mptcp는 일반 tcp 전송 프로토콜에 비해 전송 계층에서 “subflow”개념을 바탕으로 다수의 네트워크 인터페이스를 활용하여 성능을 개선시키는 기존 tcp의 확장 버전이다. 일반 tcp에서의 혼잡제어 메커니즘처럼 mptcp에서도 Lia (Liked Increases Algorithm), Olia (Opportunistic LIA), Balia (Balanced LIA)와 같은 혼잡제어 알고리즘을 통해 변화하는 네트워크 상황에 대해서 subflow들을 제어될 수 있게 한다. 단, 이러한 “coupled” 혼잡제어 알고리즘들은 reno, cubic, vegas 등의 기존 tcp 혼잡제어 알고리즘과 달리 자원 풀링 개념을 바탕으로 제공되는 subflow들간에 공평성을 지원하기 위해 참여하는 subflow들의 혼잡 윈도우들이 서로 연관되게 설계되었다[4].

일반적인 GCS와 GCP간 전송 서비스의 이용 패턴은 GCP 이용자가 동일한 커뮤니티의 거점 GCS 서비스에 접근하여 데이터를 전송 받는 경우가 많으므로 Mptcp 이용 시 고려가 필요한 서로 다른 네트워크 인터페이스들로 부터의 라우팅 경로가 비교적 단순할 수 있다. 또한, GCS 서버에 비해 GCP 머신의 네트워크 인터페이스 대역폭은 상대적으로 작고, 스펙도 떨어지므로 NIC 자체 튜닝 등을 통한 높은 전송 성능을 기대하기 어렵다. 반면, GCP가 설치되는 경량급 머신의 경우에도 복수개의 포트를 포함하는 NIC을 제공하는 경우가 많으므로 이러한 조건은 Mptcp를 적용하여 전송 성능의 향상을 도모할 수 있는 환경이 된다.

[3]에서도 언급되었듯이 대형 사이트들의 GCS와 GCS간 데이터 전송에서는 DTN 클러스터링을 이용하여 병행성과 병렬성을 극대화시켜 데이터 전송 성능을 높이는 방법이 적용되고 있다. 하지만 이 방법은 Backend 파일 시스템을 고속 마운트 기반으로 연계하여 구성할 때 적용될 수 있는 방법이고, 실제로 단일 서버에 구축된 GCS 엔드포인트의 경우에는 외부 파일시스템의 연동 없이 단일 DTN 기반의 전송 서버를 구축하는 경우가 많다. 이러한 경우 병행성 측면에서는 단일 노드 내에 존재하는 CPU 코어 수에 의존하여 병행성의 수준이 제한적으로 결정되고, 각각의 CPU 코어를 기준으로 생성되는 GridFTP 프로세스는 병렬화를 통해 TCP 스트림의 개수를 증가시킴으로써 전송 성능을 높이게 된다. 여기에 더하여 그림 1에서처럼 각각의 tcp 스트림들을 Mptcp 기반으로 “subflows” 개념을 적용하여 확장함으로써 전송 성능을 증가 시킬 수 있을 것이다.

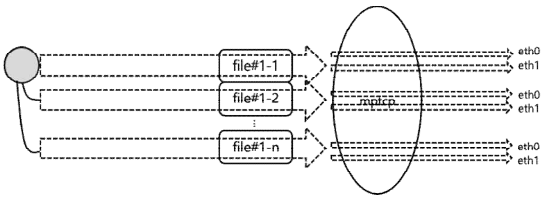

그림 1. Mptcp 기반의 병렬성 확장

본 논문에서는 이러한 “Expanded Parallelism”을 실현하기 위해 GCS와 GCP 노드에서 Mptcp 기반의 “GridFTP over Mptcp” 형상을 적용하기 전 단계로, Mptcp가 적용된 에뮬레이션 환경에서 네트워크 인터페이스 추가 및 병렬성 수준에 따라 Tcp 스트림들이 “subflows”로 더 세분화되어 나뉘어졌을 때 종단간 전송 성능의 변화 추이를 확인하고자 한다.

실험 환경은 그림 2와 같이 VM상에 Mptcp (v0.95)를 포함한 Ubuntu (v18.04.1)를 설치한다[5]. 그리고, Mininet (v2.2.2)을 이용하여 종단에 위치하는 호스트들과 이들 사이 연결하는 라우터를 생성하여 토폴로지를 구성한다. GCP의 운영을 가정하는 종단 호스트는 500Mbps 네트워크 인터페이스를 두 개 생성하여 라우터에 연결하고 GCS가 위치하는 종단 호스트는 1Gbps 단일 인터페이스를 생성하여 연결한다.

그림 2. 실험 환경 구성

이 때, net.mptcp.mptcp_enabled 속성을 이용하여 mptcp가 적용되는 경우와 일반 tcp가 적용되는 경우로 분류하여 실험 환경을 구성할 수 있다. 일반 tcp가 적용되는 경우에는 장거리 전송에 적합한 cubic 알고리즘이 혼잡제어 알고리즘으로 선택되고, mptcp가 적용되는 경우에는 Lia와 Olia의 장점을 반영하여 구현된 Balia를 적용한다. 일반 tcp 환경과 mptcp 환경에서 각각 iperf의 "-P" 옵션을 이용하여 병렬성의 정도를 조절하면서 종단간 tcp 트래픽에 대한 전송 성능을 측정한다. 또한 송신측 호스트의 CPU bandwidth을 제한하여 설정 적용함으로써 수신측 서버 대비 송신측 호스트의 성능 차이를 에뮬레이션 환경 상에서 반영하여 실험을 수행한다. 실험 결과, 그림 3에서 볼 수 있듯이, 일반 tcp 환경의 경우 최대 460Mbps의 전송 성능을 기록한 반면, mptcp 환경의 경우 최대 934Mbp의 전송 성능을 보였다. 이는 mptcp 환경에서 추가 적용된 NIC으로 인한 직관적인 전송 성능 증가에 기인한 것이다. 한편, 병렬성(1)에 대한 병렬성(10)에서의 전송 성능의 경우 일반 tcp 환경에서는 약 5% 증가되는 반면 mptcp 환경에서는 약 16% 증가되는 것으로 나타났다.

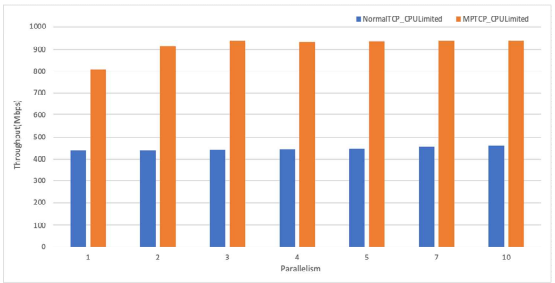

그림 3. 실험 결과

III. 결 론

본 논문에서는 Globus/GridFTP와 같은 전송 어플리케이션를 대상으로 경로의 다양성을 지원하기 위해 Mptcp를 전송 계층으로 채택하였을 때의 전송성능 변화를 사전 모니터링 하였다. 전송 성능 실험에서 Mptcp 기반의 전송 환경은 인터페이스 수의 증가에 따른 직관적인 전송 성능의 증가 외에도 Globus/GridFTP에서 주요 기능으로 활용되는 전송 스트림의 병렬성 증가와 관련하여 설정된 실험 환경에서 약 16% 성능 향상을 가져오는 것을 확인하였다.

Acknowledgement

이 논문은 2021년도 한국과학기술정보연구원(KISTI) 주요사업 과제로 수행한 것입니다.

References

[1] E. Dart et al., "The Science DMZ: A Network Design Pattern for Data-Intensive Science", SC'13 Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, Denver, USA, Nov. 2013.

[2] Z. Liu et al., "Cross-Geography Scientific Data Transferring Trends and Behavior", Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing, New York, USA, Jun. 2018.

[3] W. Hong et al., "Deployment and Performance Analysis of Data Transfer Node Cluster for HPC Environment," KIPS Transactions on Computer and Communication Systems, vol. 9, no. 9, pp. 197-206, 2020.

[4] B. Kimura and A. Loureiro, "MPTCP Linux Kernel Congestion Controls", 2018, Retrieved from https://arxiv.org/abs/1812.03210

[5] MultiPath TCP, http://www.multipath-tcp.org/