ISBN 978-89-6211-030-2 93560

글로벌공유파일시스템 I/O 성능 분석 보고서

우 준

한국과학기술정보연구원
Korea Institute of Science and Technology Information
www.kisti.re.kr

<제목 차례>

<표 차례>

<그림 차례>

제1장 개요

미국의 TeraGrid, NERSC, NCAR, 유럽의 DEISA 등과 같은 해외의 그리드 및 멀티 클러스터 서비스 환경에서는 글로벌공유파일시스템을 각 서비스 모델의 핵심 구성 요소로 활용하고 있으며, LAN에 연동된 KISTI 내부 슈퍼컴퓨터 간 또는 WAN에 연동된 KISTI와 타 기관의 슈퍼컴퓨터 간 연구 데이터의 공유를 통한 효율적이고 편리한 슈퍼컴퓨팅 자원 연동을 위해서는 기존 GridFTP, SRB, NFS 등 대신 성능, 안정성, 이기종 간 호환성 등이 보장될 수 있는 병렬파일시스템 기반의 글로벌공유파일시스템을 구축할 필요가 있다.

본 연구는 WAN을 통해 서로 공유되는 글로벌파일시스템을 구축하고, WAN 영역의 글로벌공유파일시스템 I/O 성능에 영향을 미치는 네트워크 요소들을 분석하고 최적화하였다.

제2장 글로벌공유파일시스템 I/O 성능 분석

글로벌공유파일시스템은 각 사이트마다 구축되어 있는 병렬파일시스템 클러스터들 간 네트워크를 통하여 연결되어, 서로의 파일시스템을 마운트 하여 자신의 파일시스템인 것처럼 투명하게 사용된다. 이러한 글로벌공유파일시스템들은 네트워크를 통해서 1:1 파일시스템 공유를 넘어 1:N, N:N으로 파일시스템을 공유하여, 국내·국외 간 연구데이터 공동 활용을 극대화하고, 데이터 전송속도로 인한 제한을 해결하여 효율적으로 고비용 자원들을 공유할 수 있도록 한다.

그렇지만 이러한 글로벌공유파일시스템들 간에는 네트워크를 통해서 서로 연결되므로, 서로 공유된 병렬파일시스템 간의 성능은 네트워크에 의해 크게 영향을 받게 된다. 따라서 이 장에서는 글로벌공유파일시스템에 영향을 주는 네트워크의 요소들을 다루고, 최적의 성능을 위한 최적화방법을 제시한다.

또한 최적화된 네트워크 위에 병렬파일 시스템 중의 하나인 GPFS를 통해 글로벌 공유파일시스템을 구축하고, 이후 구축될 글로벌 공유파일시스템에서 사용될 워크로드에 대한 성능을 보인다.

1. 네트워크 영향 분석 및 최적화

가. WAN 특징

일반적으로 WAN과 같이 대역폭이 높고 응답시간이 긴 네트워크(Long Fat Network : LFN)에서는 TCP 프로토콜이 병목구간이 되어서, 기본적인 TCP 설정으로는 최고 대역폭 만큼의 성능을 낼 수 없게 된다.

앞으로 살펴볼 KISTI-부산대 전용망(WAN) 구간도 1Gbps의 최대 대역폭과

4.8ms의 비교적 긴 응답시간을 갖는 네트워크로 LFN의 특징을 갖는 네트워크로 볼 수 있다.

나. WAN에서 TCP 윈도우 크기 최적화

LFN에서 사용가능 대역폭을 높이기 위해서는 TCP의 윈도우 크기를 적절하게 설정해야 한다. TCP 프로토콜은 윈도우라는 개념을 통해서 송·수신지 간 흐름제어를 하게 되는데, 송신지에서 수신지로 전송한 세그먼트에 대해서 Ack를 받아야 전송 실패 시 재전송을 위한 송신측 버퍼를 비우게 된다. 여기서 송신측 윈도우의 크기는 수신측에서 Ack를 받지 않고 한 번에 전송할 수 있는 세그먼트의 크기를 정의하는데, 송신지와 수신지 사이의 물리적 매체를 논리적으로 하나의 파이프로 생각했을 때, LFN에서는 기본적인 윈도우 크기로는 파이프의 일부분만을 채우고, 이 세그먼트가 수신지까지 갔다 올 때까지, 즉 Ack를 받을 때까지 대기하고, 이것은 치명적인 성능 저하를 야기한다.

이 파이프에 담길 수 있는 최대의 크기를 Bandwidth Delay Product(BDP)라 부르며, 이론적으로 BDP 크기 만큼의 윈도우 크기가 설정 되어야하지만, BDP 전후의 값들을 시험하면서 최적의 값을 계산할 필요가 있다.

(1) TCP 윈도우 크기 결정 방법

BDP 크기는 Full Bandwidth × RTT(Round Trip Time)로 계산할 수 있다. 실험적으로 적절한 윈도우 크기는 BDP × 2 로 알려져 있다. KISTI 전용망의 Full Bandwidth는 1 Gbps 이고 RTT는 4.8ms이므로

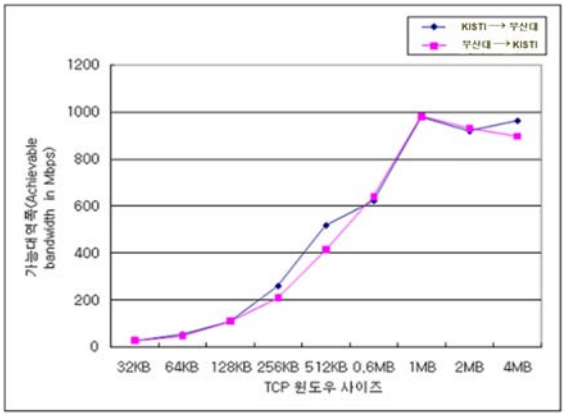
BDP = 1Gbps × 4.8ms = 4.8Mbits = 0.6MBytes

즉 윈도우 크기는 BDP × 2 = 0.6MBytes × 2 = 1.2MBytes 가 적당하다.

<그림 2-1>은 TCP 윈도우 크기에 따른 KISTI-부산대 간 사용가능 대역폭

성능을 나타낸 그래프이다. TCP 윈도우 크기를 조정함으로 인해 사용가능 대역폭이 거의 최고 대역폭까지 높아지는 것을 볼 수 있었다. 이것은 큰 대역폭과 높은 응답시간을 가지고 있는 LFN에서 나타나는 현상인데, 송신 윈도우의 크기는 수신측에서의 Ack 없이 계속해서 보낼 수 있는 데이터의 크기이고, 따라서 BDP 크기의 데이터를 송신 윈도우가 담을 수 있어야 한다. 또한 수신 윈도우의 크기는 연속해서 데이터를 받을 수 있는 수신측 버퍼의 크기인데, 데이터를 수신한 후 남은 윈도우 크기를 송신측에 알려줘서 버퍼가 넘치지 않도록 흐름제어를 하게 된다. 따라서 계속해서 최적의 성능으로 데이터를 수신하기 위해서 데이터를 수신한 후, 남은 윈도우 크기가 BDP가 되어야 한다. 이에 따라 송수신 윈도우의 크기는 앞에서 기술했던 것처럼 BDP × 2 정도의 크기가 적당할 것으로 예상된다.

(2) TCP 윈도우 크기의 영향 분석



<그림 2-1> TCP 윈도우 크기에 따른 사용가능 대역폭

<그림 2-1>은 윈도우 크기가 증가함에 따라 사용가능 대역폭이 최대 대역폭에 거의 도달하는 것을 보여준다. TCP 윈도우 크기가 1 MB에 도달할 때부터 사용가능 대역폭이 최대 대역폭에 근접하는 것을 볼 수 있다.

사용가능 대역폭의 변화는 NFS 시험 결과에도 영향을 미쳐, TCP 윈도우

크기 변경 전에는 160.74 Mbps의 사용가능 대역폭이 병목구간이 되어, 사용가능 대역폭에 수렴했지만, 변경 후에는 디스크의 성능이 병목구간이 되어, NFS의 최대성능인 Ext3의 60~70% 성능을 나타내었다.

이에 따라 네트워크 대역폭을 디스크 성능과 NFS Overhead로 인해 충분히 사용하지 못하므로, 1Gbps 네트워크에서 병렬 파일시스템인 GPFS 또는 Lustre를 사용했을 경우에, 성능 향상이 더 있을 것으로 기대할 수 있다.

우리가 목표로 하는 글로벌 공유 파일시스템의 경우 대역폭이 높고, 응답시간이 큰 네트워크(LFN)를 기반으로 하게 되며, 이러한 LFN의 경우 파일시스템의 성능에 영향을 미치는 네트워크 요소를 정리하면, 최대 대역폭, 응답시간이 가장 큰 영향을 미치게 되며, 최대 대역폭을 활용하기 위해서 TCP 윈도우 크기를 조정해 줄 필요가 있다.

(3) TCP 윈도우 크기 설정 방법.

다음은 리눅스에서 윈도우 크기를 조정하기 위해 수정해야 하는 커널변수이다.

rmem_max : receive 버퍼 크기
wmem_max : send 버퍼 크기
tcp_rmem : TCP receive 윈도우 크기
tcp_wmem : TCP send 윈도우 크기
tcp_window_scaling : 윈도우 창 배율

모두 /proc/sys/net/ 디렉토리에 위치하며, 직접 또는 sysctl 명령어로 수정할 수 있다. “rmem_max”와 “wmem_max"는 네트워크 연결 당 할당되는 버퍼의 크기이고, ”tcp_rmem"과 “tcp_wmem"은 TCP 흐름제어를 하는 윈도우

크기의 범위를 지정한다. 윈도우의 크기는 버퍼의 크기를 넘을 수 없으므로, 버퍼크기보다 크게 설정한 윈도우 크기는 의미가 없다. tcp_window_scaling은 TCP헤더의 윈도우 크기에 64 KB 이상 저장할 수 없는 제약을 해결하기 위한 커널변수로 윈도우 크기에 따라 자동으로 설정된다.

다음은 윈도우 크기를 1 MB로 변경하기 위한 예를 보인다. “sysctl -a |grep mem" 명령을 실행하여 현재 TCP 윈도우 크기 설정을 확인한다. 확인할 커널 변수는 다음과 같다.

```
# sysctl -a |grep mem
net.ipv4.tcp_rmem = 302400        87380    302400
net.ipv4.tcp_wmem = 4096          87380    302400
net.core.rmem_max = 302400
net.core.wmem_max = 302400
```

TCP 윈도우 크기가 적절하게 설정되어 있지 않다면, 앞에서 기술한 방법으로 BDP를 계산하여 TCP 윈도우 크기를 결정한다. KISTI-부산대 구간에는 1MB로 설정하였다. TCP 윈도우크기를 변경하기 위해서 “sysctl -w" 명령을 사용하여 각 커널변수에 값을 할당한다. 다음은 변경하는 예를 보여준다. 변경한 값은 그 이후의 TCP 연결에 대해 바로 적용된다. ”net.ipv4.tcp_rmem" 과 “net.ipv4.tcp_wmem"에는 윈도우의 최소값, 기본값, 최대값을 지정하고, ”net.core.rmem_max"와 “net.core.wmem_max"에는 버퍼의 크기값을 지정한다.

```
#sysctl -w net.ipv4.tcp_rmem =“4096 87380 1048576”
#sysctl -w net.ipv4.tcp_wmem =“4096 87380 1048576”
#sysctl -w net.core.rmem_max = 1048576
#sysctl -w net.core.wmem_max = 1048576
```

(4) 기타 최적화 요소

기타 최적화할 요소들로 MTU(Maximum Transmission Unit)과 Selective Acknowledgement가 있다. LFN은 대부분 고성능 네트워크이기 때문에, 기존의 1500바이트보다 큰 9000바이트의 점보프레임을 지원한다. 점보프레임을 사용함으로써 패킷 처리량이 감소하여 10% 정도의 성능 향상을 기대할 수 있다. 커널옵션에서 연결된 네트워크 간 사용할 수 있는 가장 큰 MTU를 자동으로 결정하는 Ip_no_pmtu_disc 변수가 있으며, 최신의 커널에는 자동으로 가능하게 설정되어 있다. 다음은 MTU 크기를 변경하는 예와 Ip_no_pmtu_disc를 설정하는 예를 보인다.

```
# ifconfig eth0 mtu 9000
```

```
# sysctl -w net.ipv4.Ip_no_pmtu_disc = 0
```

Selective Acknowledgement 커널변수는 혼잡한 네트워크에서 일부 TCP 세그먼트가 손실되어 모든 세그먼트를 받지 못하더라도 받은 것에 대해 선택적으로 응답을 해줌으로써 전체 재전송을 방지하여, 성능을 개선시킨다. 다음은 Selective Acknowledgement를 설정하는 예를 보인다.

```
# sysctl -w net.ipv4.tcp_sack = 1
```

2. KISTI-부산대 간 네트워크 영향 측정

가. 테스트베드

네트워크의 성능을 최적화 한 후 앞으로 사용될 GPFS 클러스터 파일시스

템을 KISTI와 부산대에 구축하여 파일시스템의 I/O 성능을 시험하였다. KISTI에 있는 X4500이 GPFS 서버로, 부산대에 있는 PGFS01, PGFS02등 4대와 KISTI에 있는 KGFS01, KGFS02등 4대를 GPFS 클라이언트로 구성하였다.

(1) 테스트베드 사양

시험에 사용된 테스트베드의 사양은 다음과 같다.

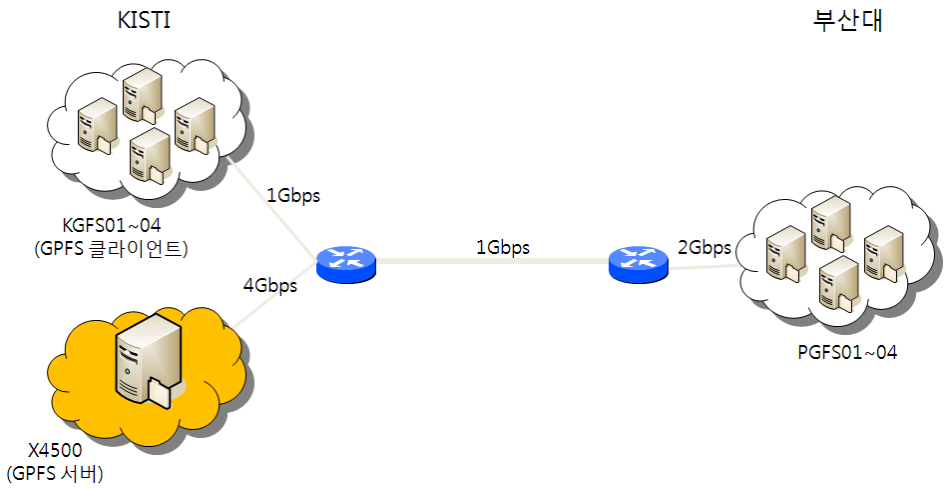
<표 2-1> 테스트베드 사양

테스트베드 이름	CPU	RAM	OS	위치
X4500	Opteron Quad Core 1.7Ghz*4	16G	Redhat Enterprise Linux 4	KISTI
KGFS	Xeon 2.80Ghz Dual 32bit cache 512K	3G	Red Hat Enterprise Linux AS (2.6.9-54.ELsmp)	KISTI
PGFS	Xeon 2.33Ghz Quad 64bit cache 4096K	2G	Red Hat Enterprise Linux ES (2.6.9-42.ELsmp)	부산대

(2) 네트워크 구성

<그림 2-2>는 네트워크 구성을 보인다. X4500-KGFS의 네트워크 대역폭은 X4500-Switch(KISTI) 간이 4 Gbps이고 KGFS-Switch(KISTI) 간 1 Gbps로 KGFS 클라이언트 한 대당 최대 1 Gbps 의 성능을 기대할 수 있고, KGFS 클라이언트 4대이상이 동시에 X4500에 I/O를 일으킬 경우 최대 4 Gbps의 네트워크 성능이 지원된다.

X4500-PGFS의 네트워크는 X4500-스위치(KISTI) 구간 4 Gbps이고 각 PGFS 클라이언트-스위치(부산대) 구간이 2 Gbps 임에도 불구하고 KISTI와 부산대 두 스위치간의 대역폭이 1 Gbps로 병목구간이 되어 최고 1 Gbps의 성능을 기대할 수 있다.



<그림 2-2> 네트워크 구성

(3) GPFS 클러스터 구성

이 시험에서 GPFS의 테스트베드 구성을 보면 총 세 개의 GPFS 클러스터로 되어 있다. KISTI에 2개의 클러스터(GPFS_X4500, GPFS_KGFS), 부산대에 1개의 클러스터(GPFS_PGFS)이고, GPFS_X4500 클러스터가 서버의 역할을 하였고 GPFS_KGFS, GPFS_PGFS 2개의 클러스터가 시험을 위한 클라이언트로서 사용되었다.

<표 2-2> 클러스터 구성

GPFS 클러스터	구성노드	GPFS 노드수	위치
GPFS_X4500	X4500	1	KISTI
GPFS_KGFS	KGFS01~04	4	KISTI
GPFS_PGFS	PGFS01~05	5	부산대

시험은 크게 2가지로 실시되었는데, KISTI 내에 있는 GPFS_KGFS에서 GPFS_X4500으로의 I/O성능과 KISTI-부산대간의 GPFS_PGFS에서 GPFS_X4500으로의 성능을 측정하여, 근거리의 LAN 파일시스템 성능과 원거리의 WAN 파일시스템에서 성능을 비교하고 분석함을 목적으로 하였다.

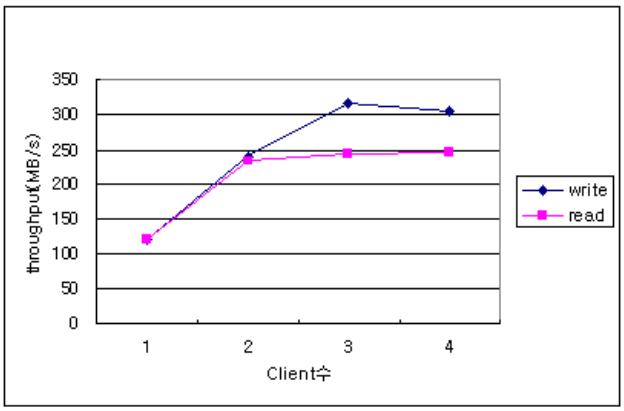
X4500에 구성된 GPFS 서버는 8개의 SATA 디스크로 구성이 되어서 스펙상의 데이터로는 최대 800 MB/s의 결과가 예상되지만, EXT3로 8개 디스크의 동시 I/O 성능을 IOZone을 통해서 측정해본결과 500 MB/s 의 성능을 확인할 수 있었다.

나. 성능측정 결과

(1) 클라이언트 수에 따른 성능

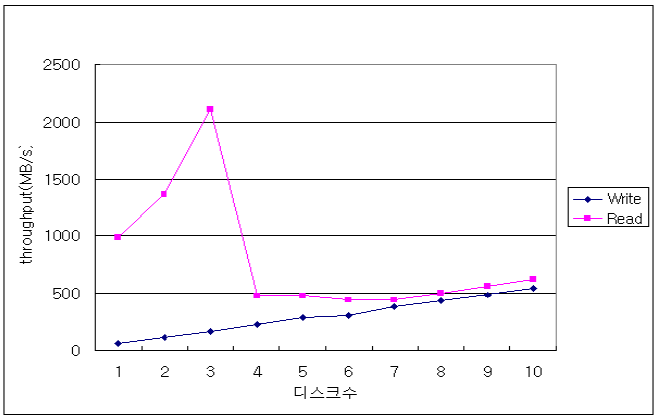
I/O의 성능은 Workload가 되는 데이터의 크기, 블록크기, 액세스 패턴 등 여러 가지 요소에 의해 영향을 받게 된다. 여기서는 앞으로 우리가 서비스를 제공하게 될 과학응용 의 워크로드로 가정하여 대용량의 파일크기(4 GByte)에 큰 블록크기(1 MB), 순차적인 액세스 패턴으로 I/O를 시험하여, 네트워크 대역폭을 충분히 활용하기 위해 클라이언트의 수를 변수로 하고 시험을 진행하였다.

<그림2-3>은 KISTI내에 있는 KGFS와 X4500간의 GPFS I/O 성능을 측정한 결과이다. GPFS_KGFS 클러스터를 클라이언트로 하여 GPFS_X4500 GPFS 서버에 I/O를 요청하였다. 결과를 보면 클라이언트 수가 증가함에 따라 write의 경우는 3에서 read의 경우는 2에서 최대 성능에 수렴되고 있다.



<그림 2-3> KGFS에서 X4500로의 GPFS 전송성능 측정

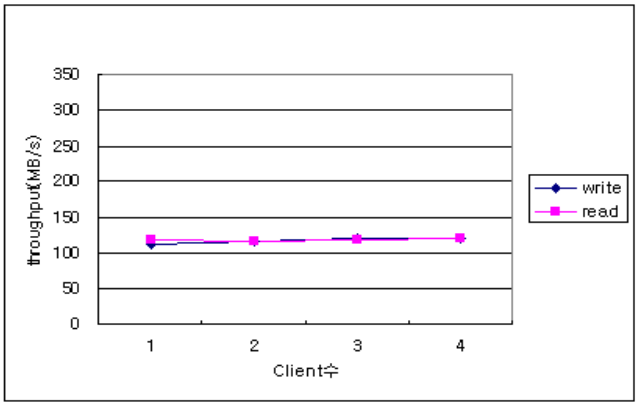
<그림2-4>의 EXT3 8개의 디스크에 동시에 I/O를 일으켰을 때 성능과 비교해서 GPFS 자체의 오버헤드로 인해 성능이 40%가량 떨어지는 것을 볼 수 있다. X4500과 KGFS간의 네트워크의 성능이 4 Gbps(500 MB/s)이기 때문에 GPFS가 네트워크를 충분히 사용하지 못하고 있는 것을 알 수 있다. GPFS_X4500 서버에 구성된 디스크 수를 늘리면 성능이 더 개선될 것으로 예상되지만, X4500-PGFS간의 대역폭이 1 Gbps 이기 때문에, 이 이상의 성능은 의미가 없다고 판단되어 그 이상의 성능개선을 위한 실험은 진행하지 않았다.



<그림 2-4> X4500 EXT3 성능

<그림2-5>의 결과는 부산대에 있는 PGFS와 KISTI에 있는 X4500간의 GPFS I/O 성능을 측정한 결과이다. 즉 KISTI와 부산대간의 WAN GPFS 파일 시스템의 I/O 성능으로 GPFS_PGFS 클러스터를 클라이언트로 하여 GPFS_X4500 GPFS 서버에 I/O를 요청하였다. <그림 2-4>에서 보인것 처럼 GPFS_X4500 서버의 GPFS I/O 성능이 KISTI내 구간에서 1 Gbps를 넘기 때문에, KISTI-부산대간 사용가능 대역폭이 1 Gbps로 병목구간이 되어 네트워크가 제공하는 성능을 완전히 사용한다면, 최대 115 MB/s의 성능을 예상할 수 있다.

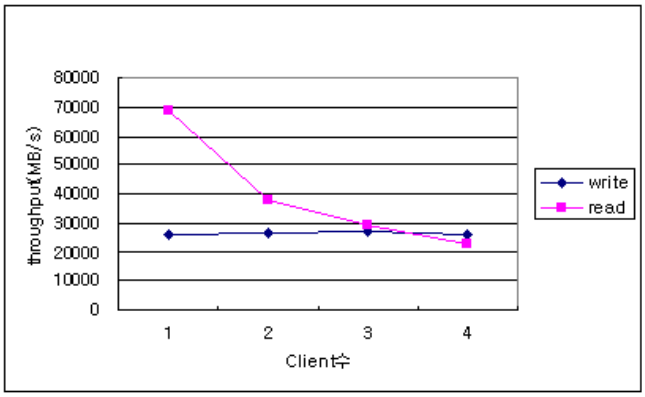
시험결과에서 보면 이론적으로 예상할 수 있는 최적의 성능을 나타내고 있어, 네트워크가 제공하는 성능을 모두 이용하는 것으로 판단된다.



<그림 2-5> PGFS에서 X4500로의 GPFS 전송성능 측정

<그림2-6>은 KGFS에서 PGFS로 NFS I/O 성능시험을 진행한 결과이다. 여기서 사용된 디스크는 RAID1 Mirroring 디스크로 write 30 MB/s, read 110 MB/s의 성능을 보여주는 디스크로 NFS를 통해 시험 했을 때 최대 read 70 MB/s, write 28 MB/s 정도를 보여주고 있다. NFS의 경우에 클라이언트 수가 증가함에 따라 성능이 계속해서 감소하며, GPFS와 비교해 확장성이 크게 떨어지므로, 다수의 노드에 동시에 서비스를 제공하기 위해서는 GPFS와 같

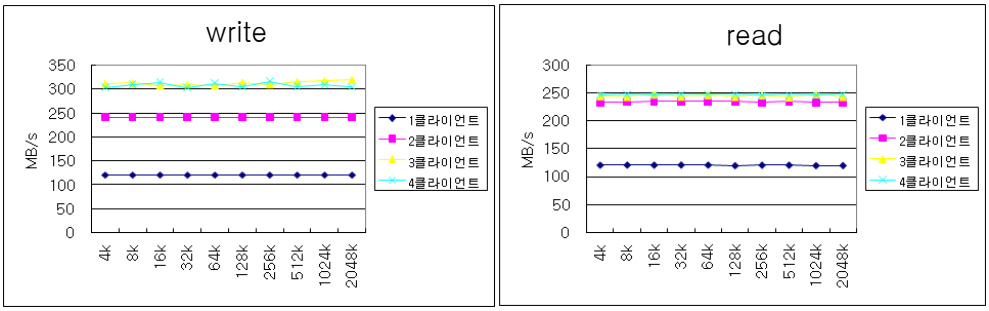
은 클러스터 파일시스템이 필요하다.



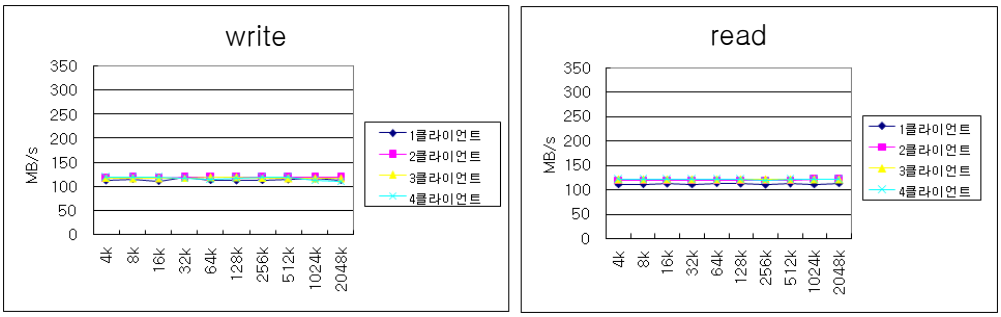
<그림 2-6> KGFS에서 PGFS로의 NFS 전송성능 측정

(2) 블록크기에 따른 성능

<그림 2-7>과 <그림 2-8>은 위와 동일한 환경에서 블록크기에 따른 KISTI내의 LAN 구간과 KISTI-부산대 WAN 구간에서의 GPFS의 I/O 성능이다. 이후 사용될 워크로드와 유사한 4G 파일에 Sequential I/O 패턴으로 진행을 하였다. LAN 구간에서 블록크기에 따라 영향을 받지 않고 동일한 성능을 보이며 WAN 구간에서도 역시 블록크기에 따른 영향 없이 네트워크의 성능을 충분히 사용하여 115 MB/s의 성능을 나타내고 있다. 따라서 앞으로 KISTI-부산대 간 서비스 될 워크로드가 600 MB이상 대용량 파일의 Sequential I/O 패턴임을 고려할 때, 현재의 구성으로 네트워크가 제공하는 최대 대역폭을 충분히 활용할 수 있다고 판단된다.



<그림 2-7> KGFS에서 X4500로의 블록크기에 따른 GPFS 성능



<그림 2-8> PGFS에서 X4500로의 블록크기에 따른 GPFS 성능

3. KISTI-부산대 간 글로벌공유파일시스템 성능 분석

병렬파일시스템의 특징 중 하나는 다양한 병렬파일시스템 마운트가 일어나게 되고, 따라서 다양한 하드웨어 조합이 되는 경우가 많다. 각 클러스터의 하드웨어 조합에 따른 GPFS 파일시스템의 영향을 시험하기 위해서 KISTI-부산대 간에서 여러 가지 테스트베드 조합의 GPFS 병렬파일시스템 마운트를 하여 GPFS I/O 성능시험을 하였다.

가. 테스트베드

(1) 테스트베드 사양

시험에 사용된 테스트베드의 사양은 <표 2-3>과 같다.

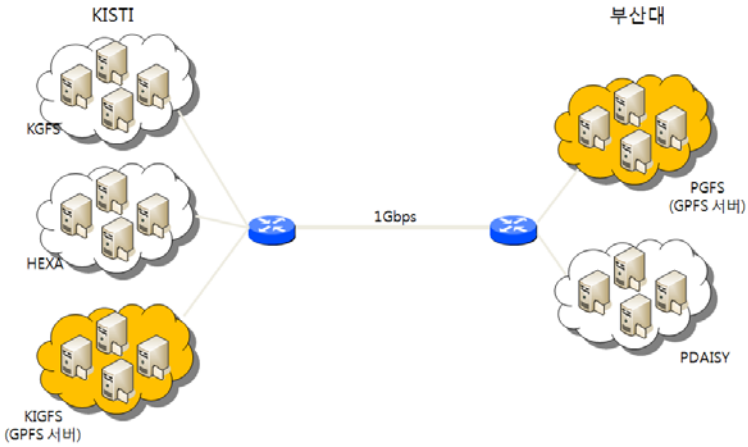
<표 2-3> 테스트베드 사양

테스트베드 이름	CPU	RAM	OS	위치
KGFS	Xeon 2.80Ghz Dual 32bit cache 512K	3G	Red Hat Enterprise Linux AS (2.6.9-54.ELsmp)	KISTI
KIGFS	IBM POWER5+ 2.3GHz × 16	30G	AIX 5.3	KISTI
HEXA	Xeon 3.06Ghz Dual 32bit cache 1024K	3G	Fedora Core (2.6.18.8)	KISTI
PGFS	Xeon 2.33Ghz Quad 64bit cache 4096K	2G	Red Hat Enterprise Linux ES (2.6.9-42.ELsmp)	부산대
PDAISY	IA64 1.5Ghz × 16	16G	2.4.21-sgi306rp1	부산대

(2) 네트워크 구성

KISTI에는 세가지 종류의 테스트베드가 있으며 KGFS, HEXA, KIGFS 노드들은 각각 1 Gbps로 KISTI 라우터에 연결되어 있다. 부산대에는 2가지 종류의 테스트베드가 있으며 PGFS, PDAISY 노드들도 역시 각각 1 Gbps로 부산대 라우터에 연결되어 있으며, KISTI 라우터와 부산대 라우터간에는 1 Gbps

로 연결되어 있다.



<그림 2-9> 테스트베드 네트워크 구성

(3) GPFS 클러스터 구성

GPFS_KGFS, GPFS_KIGFS, GPFS_PDAISY 총 세 개의 GPFS 클러스터로 구성되어 있다.

<표 2-4> 클러스터 구성

클러스터 이름	구성노드	GPFS 노드수	위치
GPFS_KGFS	KGFS01~04	4	KISTI
GPFS_KIGFS	KIGFS01~04, HEXA01	5	KISTI
GPFS_PDAISY	PGFS01~05, PDAISY00~04	9	부산대

시험에서 GPFS 파일 서버로 사용된 클러스터는 GPFS_KIGFS와 GPFS_PDAISY이다. KISTI에 위치해있는 GPFS_KIGFS 서버의 성능을 시험하

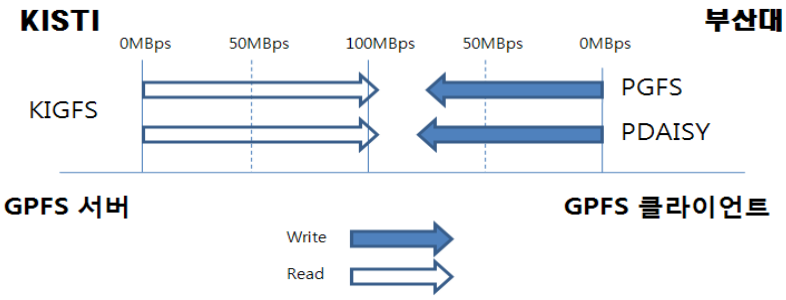
기 위해서 부산대에 있는 GPFS_PDAISY클러스터를 사용하였으며, 부산대에 위치해 있는 GPFS_PDAISY 서버의 성능을 시험하기 위해서는 KISTI에 있는 GPFS_KGFS와 GPFS_KIGFS 클러스터들을 클라이언트로 사용하였다.

나. 장거리 전용 네트워크 전송속도 측정

<그림 2-10>은 KISTI 안에 있는 GPFS 클러스터인 GPFS_KIGFS를 파일시스템 서버로 두고 부산대에 있는 GPFS_PDAISY클러스터(PGFS01~05, PDAISY00~04)를 클라이언트로 시험한 결과이다. 테스트베드의 조합에 따른 성능을 측정하기 위해서 PGFS와 PDAISY를 각각 클라이언트로 선정해서 I/O 성능시험을 진행하였다.

파일크기는 GPFS_PDAISY 클러스터의 GPFS Pagepool 크기보다 충분히 큰 4 GB를 사용해서 캐쉬효과를 최소화하여 네트워크에 의한 전송속도를 측정하였고, 블록크기는 GPFS 파일시스템의 블록크기와 동일한 256 KB를 사용하여 Sequential 액세스 패턴으로 시험하였다.

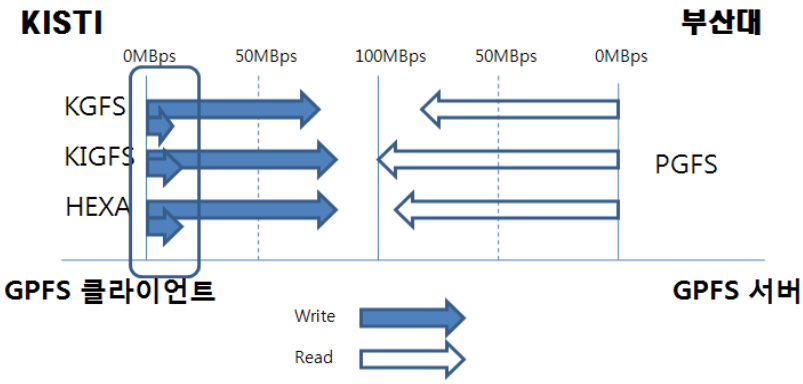
I/O 성능을 보면 PGFS와 PDAISY 모두 write 80 MBps, read 105 MBps로 비슷한 성능을 나타낸다.



<그림 2-10> GPFS 서버에 대한 전송성능 측정(1)

<그림 2-11>은 부산대에 있는 GPFS 클러스터인 GPFS_PDAISY를 파일시스템 서버로 두고 KISTI에 있는 GPFS_KGFS클러스터(KGFS01~KGFS05)와 GPFS_KIGFS클러스터(KIGFS01~KIGFS04, HEXA01)를 클라이언트로 두고 I/O 성능을 시험한 결과이다.

부산대의 GPFS_PDAISY 클러스터를 GPFS 서버로 구성하여 KISTI의 클러스터로 I/O를 발생시켰을 때 이상 현상을 발견할 수 있는데 read 연산의 경우 모두 80 MBps 이상의 성능을 일관성있게 나타나는 반면 write 연산에서 I/O 성능이 13 MBps와 80 MBps 로 양극화 되는 현상을 발견하였다.



<그림 2-11> GPFS 서버에 대한 전송성능 측정(2)

다. 문제점 분석

(1) 네트워크 성능

KISTI와 부산대 간에는 1 Gbps 네트워크로 연결되어 있기 때문에 GPFS I/O 성능 문제에 가장 큰 영향을 미칠 수 있는 가능성은 네트워크 성능이다. 클라이언트와 서버 모두 MTU 9000으로, TCP 윈도우 크기는 1 MB로 설정되

어있다. PGFS 클러스터의 GPFS 서버 노드는 5대로 I/O가 발생할 때 5개의 네트워크 플로우가 발생한다. 이에 따라 각 클라이언트 서버 조합에 대해서 5개의 네트워크 플로우로 네트워크 성능 시험을 진행하였다.

<표 2-5>는 GPFS I/O가 일어날 때와 동일한 네트워크 성능이다. KGFS, KIGFS, HEXA 의 각 클라이언트 1로부터 PGFS01~05까지의 5개의 네트워크 플로우 성능을 측정하여 모든 플로우의 성능을 합한 값이다. 모든 테스트베드 조합에 대해서 양방향으로 최고 대역폭인 1 Gbps에 근접하게 나타나며, 따라서 네트워크의 영향으로 인한 성능의 저하는 아닌 것으로 보인다.

<표 2-5> KISTI 클라이언트-부산대 서버 간 네트워크 성능

클라이언트	송신	수신
KGFS	980Mbps	978Mbps
KIGFS	986Mbps	967Mbps
HEXA	985Mbps	950Mbps

(2) 디스크 I/O 성능

디스크는 GPFS 클러스터 파일시스템을 이루는 가장 기본적인 부분으로서, 디스크의 I/O 성능은 전체 파일시스템에 영향을 미친다. 이에 따라 PGFS 클러스터에 연결되어 있는 SAN 디스크의 I/O 성능을 측정하였다.

디스크의 성능은 write시 264 MBps, read시 354 MBps 로 네트워크의 대역폭이상으로 충분한 성능을 나타내었다.

(3) NFS 성능

GPFS의 성능을 다른 분산 파일시스템의 성능과 비교하기 위해서 기존의 커널을 수정할 필요가 없고 설치가 간단한 NFS 파일시스템을 통해서 GPFS와의 성능을 비교하였다. GPFS 파일시스템 서버를 구성하는 PGFS01~04중에서 PGFS01을 NFS 서버로 설정하고 각 클라이언트에서 NFS 파일시스템 성능을 측정하였다. 성능시험은 iozone을 통해서 GPFS 파일시스템을 측정할 때와 동일한 조건으로 수행하였다. 시험에 사용된 디스크는 RAID 1 디스크로 write시 30 MB/s, read시 120 MB/s의 성능을 갖는다.

<표 2-6>의 NFS I/O 성능측정 결과를 보면 write시 26 MB/s로 디스크의 성능을 모두 발휘한다.

따라서 위의 문제점 분석들을 통해서 보면, 네트워크와 디스크 성능은 13 MBps 이상으로 충분한 성능을 나타내고 있음에도 불구하고, NFS와 비교해서 성능이 더 떨어지므로 문제점은 GPFS 내부에 있는 것으로 판단할 수 있다.

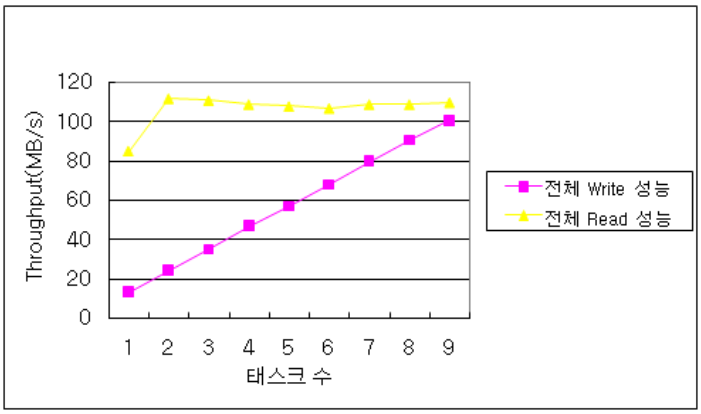
<표 2-6> KISTI 클라이언트-부산대 서버 간 NFS 성능

클라이언트	Write	Read
KGFS	26MBps	71MBps
Hexa	36MBps	55MBps

라. 적합성 평가

GPFS의 성능을 최대로 활용하기 위한 방안은 클라이언트의 수가 증가하면서, 이에 따라 I/O 작업에 참여하는 태스크 수를 증가시키는 것이다. <그림 2-12>는 KISTI 클라이언트 태스크 수를 증가시키며 부산대 쪽의 PGFS 서버에 대한 전체 I/O 성능을 측정한 것이다. I/O 작업에 참여하는 태스크의 수가 증가함에 따라 전체 write 성능이 이에 비례하여 증가하는 것을 볼 수 있으며

태스크 수가 9에 이를 때에는 101 MBps 까지 증가하여 네트워크를 거의 최대로 사용하는 것을 볼 수 있다.



<그림 2-12> KGFS 클라이언트-PGFS서버 간 GPFS I/O 성능

이러한 경우 클러스터 시스템에서 GPFS를 사용하는 응용들에 따라서 write I/O 성능이 다르게 나타날 수 있다. I/O 연산에 참가하는 노드 수가 적은 기존의 응용들에 대해서는 좋은 성능을 기대하기 힘들지만, I/O 연산에 참가하는 노드가 많은 과학응용에 대해서는 더 좋은 write I/O 성능을 발휘한다. write I/O 성능을 충분히 활용하기 위해서 각각의 노드들이 I/O 작업에 참여해야한다.

클러스터 환경에서 대부분의 과학응용은 다수의 노드가 유기적으로 연결되어 작업을 수행하며, 따라서 다수의 노드가 I/O 연산에 참가하게 된다. I/O에 참여하는 노드가 증가함에 따라 I/O 성능이 함께 증가하며, 이는 활용가능한 네트워크의 대역폭을 최대로 사용할 때까지 증가한다. 현재 구축된 WAN GPFS 클러스터 파일시스템은 병목구간인 KISTI와 부산대간 1 Gbps 네트워크의 최적화를 통해서, 프로토콜 오버헤드를 포함하여 이론상 가능한 115 MB/s까지의 최대의 대역폭을 제공하며, 다수의 노드가 데이터를 공유하는 클러스터 환경에 적합하다.

제3장 결론

본 연구는 WAN을 통해 KISTI와 원거리에 있는 타기관의 슈퍼컴퓨팅 자원 간 데이터의 공유를 통한 효율적이고 편리한 슈퍼컴퓨팅 자원 연동을 위해서 기존의 분산파일시스템이나 전송프로그램 대신 성능, 안정성, 호환성이 보장될 수 있는 글로벌공유파일시스템에 대한 구축기술을 확보하기 위한 것이다. WAN을 통해서 글로벌공유파일시스템들이 서로 연결되므로, 서로 공유된 병렬파일시스템 간의 성능에 영향을 미치는 네트워크에 대해 살펴보았다. LFN 네트워크에 대해서 성능 최적화를 수행하였고, 최대 대역폭까지 사용가능 대역폭이 향상되어 네트워크가 글로벌공유파일시스템에 충분한 성능을 제공하는 것을 보였다. 여러 테스트베드에 글로벌공유파일시스템을 구성하여 GPFS가 글로벌공유파일시스템으로서 네트워크를 충분히 활용하여 적합한 성능을 제공하는 것을 보였다.

참고문헌

1. GPFS V 3.1 : Concept, Planning, and Installation Guide, IBM

2. GPFS V 3.1 : Advanced Administration Guide, IBM

3. Raymond L. Paden, GPFS Programming, Configuration and Performance Perspectives, GPFS Tutorial

4. Rob Latham, Rob Ross, et. Al, Parallel I/O in Practice, Tutorial material

5. IOZone User Guide

6. How To : Network / TCP / UDP Tuning, http://wwwx.cs.unc.edu/~sparkst/howto/network_tuning.php