이기종 슈퍼컴퓨팅 자원들을 활용한 Globus 기반 그리드 구축에 관한 연구

강경우
천안대학교 정보통신학부
e-mail:kwkang@cheonan.ac.kr

A Study on the Development of Globus-based Grid using Heterogeneous Supercomputing Resources

Kyung-Woo Kang
Div. of Information & Communication Eng., Cheonan University

요약

지난 몇 년 동안 컴퓨터의 처리 속도와 네트워크의 속도는 아주 빠르게 향상되고 있다. 빨라진 네트워크를 이용하여 분산된 자원들을 연결하고 하나의 시스템처럼 사용하고자 하는 노력들이 진행되고 있는데 이와 같은 것을 일반적으로 메타컴퓨팅 또는 그리드라고 부른다. 본 논문에서는 국내 주요 슈퍼컴퓨팅 자원들을 활용한 그리드 개발과 전산유체역학 프로그램을 이용한 가능성 시험을 수행하였다. 시험결과로 분산자원을 이용했을 때 네트워크의 속도에 따라 성능향상을 얻을 수 있었다.

1. 서론

인터넷이 보편화되고 컴퓨터 및 네트워크의 성능이 향상됨에 따라 분야에 따라 컴퓨터를 이용하는 양상은 변하고 있다. 분산처리 분야에서 네트워크의 기술적인 변화는 통신망을 통해 서로 연결된 컴퓨터들을 하나의 문제를 해결하는데 사용할 수 있도록 해 주고 있다. 지금까지는 동일한 컴퓨팅 자원들을 통합하는 NOW(Network Of Workstations), PC클러스터링 기술에 노력하여 왔지만 이제 또 다른 하나의 시도로써 자원통합에 있어서 동일 기종 컴퓨터들만이 아니라 이기종 컴퓨팅 자원들과 대용량 저장장치, 다양한 고성능 연구 장비들이 포함되고 있는데, 이러한 통합된 환경을 "Grid"(그리드) 라는 용어를 가지고 표현하고 있다.[5]

그리드와 유사한 용어로써 메타컴퓨터 혹은 메타컴퓨팅이라는 용어는 80년대 후반에 생겨났다. 네트워크 환경에서 연결된 워크스테이션들과 슈퍼컴퓨터들을 사용자가 단일 컴퓨터를 이용하듯이 사용할 수 있는 개념에 관련지었다. 초기의 메타컴퓨터는 CM-2, Cray-2, 파일서버, 워크스테이션 정도를 기가비트 LAN으로 연결하고자 하는 것이 고작이었다. 사실 메타컴퓨팅과 유사한 용어가 각 분야마다 독립적으로 사용되어 왔는데, 예를 들면, seamless 컴퓨팅, scalable 컴퓨팅, global 컴퓨팅 등이 그것이다. 그러다가 1990년대 후반에 그리드라는 용어로 통일이 되었다.[1, 5]

2000년부터 국내 슈퍼컴퓨터를 통합하는 방안에 대해 논의해 왔다. 이들 시스템은 서로 제조한 회사가 다르고 성능도 다르지만 CPU 사용율 측면에서 차이를 보이고 있었다. 이에 따라, 본 연구에서는 Globus Toolkit을 이용하였고 그리드구축을 위하여 그림 1과 같이 SMP구조의 컴퓨팅 자원인 HPC320 시스템 1대, HPC160 시스템 1대, GS320 시스템 1대, IBM SP2 시스템 2대과 MPP구조의 컴퓨팅 자원인 Cray T3E 1대를 이용하였다.

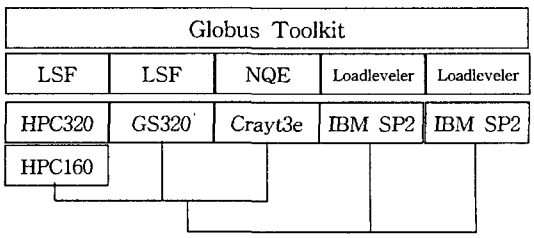

그림 1 그리드 구조도

2. 미들웨어

2.1 Globus Toolkit

Globus toolkit은 그리드 서비스를 제공하는 미들웨어로써 전세계적인 그리드 개발 과제에서 가장 많이 사용되고 있다(National Technology Grid, European DataGrid, NASA Information Power Grid, Grid Physics Network, ASCI Distributed Resource Management (DRM) Testbed, GUSTO,..등)[1, 2, 3, 4, 5]. 이렇게 Globus toolkit이 널리 사용되게 된 이유는 globus toolkit이 분리될 수 없는 단일 시스템이 아니라 그리드에서 필요로 하는 다양한 서비스들을 독립적인 요소로써 제안하고 있기 때문이다. globus의 또 다른 장점으로는 기존에 각 시스템 및 네트워크의 관리 정책이나 운영 도구들을 무시하지 않고 각 요소들과 협력하여 그리드를 이루어 나간다는 점이다. 따라서, globus toolkit의 요소들도 하드웨어 측면이나, 소프트웨어 측면에서 상이한 시스템들간에 성능 저하를 줄이면서 통합하기 위한 기능을 위한 것이 대부분이다. Globus toolkit은 크게 그리드 보안, 정보 서비스, 자원 관리, 데이터 관리 등으로 나뉘어 진다.[2, 4, 6]

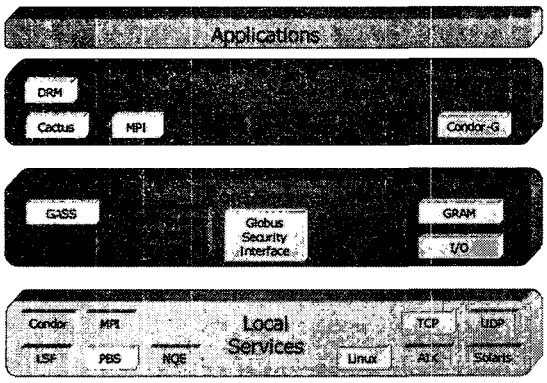

그림 2 Globus Toolkit의 구조

Globus toolkit에서는 보안을 담당하는 부분을 GSI라고 부르며, 그리드 보안은 분산 자원들을 공유함에 따라 발생하는 자연스러운 문제이다. 사용자의 입장에서는 안전하면서도 사용의 편리성을 요구할 것이고, 각 자원을 소유하고 관리하는 관리자의 입장에서는 자원이 그리드 환경에 노출되는 것이기 때문에 사용의 편리성보다는 더 안전한 보안을 원할 것이다. 이를 위해 GSI는 single-sign-on을 제공하고 globus proxy를 이용한다. 사용자는 그리드환경에 한번의 인증과정을 거침으로 사용이 허용된 자원들을 사용할 수 있고 분산된 각 자원에 대한 사용자 인증은 proxy가 대신 수행한다. 그렇지만 각 자원 내에서 자원에 대한 사용에 대한 허용범위는 각 자원이 제시하는 보안체계를 따른다.[4]

두 번째로, globus toolkit에서 정보서비스를 수행하는 요소를 MDS라고 부른다. MDS는 그리드 내에 존재하는 자원들의 상태정보를 공유하고 사용자들에게 제공하기 위한 요소로써 인터넷의 DNS와 비슷한 것이다. 정보를 저장하고 사용자들에게 제공하기 위해 MDS는 LDAP를 이용한다. 정보 서비스를 위해 Globus에서는 두 개의 서버를 제공하는데, 각 자원의 정보를 수집하는 GRIS와 수집된 정보를 통합하는 GIIS이다. 이들이 수집하여 제공하는 정보는 각 자원의 구조, 노드 수, 부하 정보, 배치작업 스케줄러, 네트워크 상태,... 등이다. 이들 정보는 어플리케이션 개발자나, resource broker,..등에 제공된다.[6]

세 번째로, globus toolkit에서 자원 관리를 담당하는 부분을 GRAM이라 부른다. GRAM은 globus toolkit의 가장 중심이 되는 요소로써 원격지 자원을 사용할 수 있게 하고 분산 자원들을 동시에 사용하게 하며 자원들의 관리의 상이함을 처리한다. 사용자는 자신의 작업을 그리드 환경에서 처리할 때 원하는 요구사항을 RSL이라는 스크립트를 이용하여 표현한다. GRAM은 resource broker를 이용하여 RSL 스크립트를 저급의 스크립트로 변환하고 각 자원에 있는 스케줄러가 처리할 수 있는 형태의 스크립트로 변환한다. 이 변환과정에서 resource broker는 현재 또는 미래에 사용 가능한 자원을 검색하기 위해 MDS를 이용하게 된다. 작업이 분산환경에 할당되면 각 작업의 협업을 위해 GRAM은 DUROC이라는 요소를 이용한다.

마지막으로 globus에서는 데이터 관리를 위해 GASS, GridFTP, Replica catalog를 제공한다. GASS는 GRAM과 밀접한 관련이 있는 요소로써 원격지에 있는 파일을 사용하여 작업을 처리하기 원하거나 원격지에서 처리한 작업의 결과를 또 다른 저장장치에 저장하고 싶을 때 사용한다. GridFTP는 그리드 내의 데이터가 대규모 대용량이란 점을 고려하여 고속으로 파일 전송과 파일의 이어받기를 가능케 하는 요소이다. Replica catalog는 데이터 그리드를 위해 개발된 것으로 데이터들을 분산 저장 및 관리함으로 필요할 때에 신속하게 데이터를 사용할 수 있게 하는 기술이다.[6]

2.2 로컬 작업 스케줄러

Globus의 자원관리 시스템인 GRAM은 각 컴퓨팅자원의 로컬 작업 스케줄러와 연결되어 수행되어야 한다. 왜냐하면 globus가 각 자원의 노드들을 마음대로 할당할 수 는 없고 각 자원의 로컬 작업 스케줄러를 통해서만 작업을 할당할 수 있기 때문이다. 그렇다고 모든 스케줄러와 연동되는 것은 아니고 다음과 같은 로컬 작업 스케줄러와 연동이 된다: Unix fork, POE, Condor, Easy-LL, NQE, Prun, LoadLeveler, LSF, PBS, GLUnix, Pexec. 본 연구에서는 각 시스템에서 사용중인 Unix fork, NQE, LoadLeveler, LSF를 사용하였다.

3. 그리드 구축 및 실험

3.1 슈퍼컴퓨팅 자원

본 절에서는 본 과제에서 사용한 슈퍼컴퓨팅자원들의 성능과 이들간의 네트워크 환경을 기술한다. 표2는 사용한 6개 시스템에 관한 정보를 보여주고 있다. 이들 중 2개 시스템인 HPC320과 HPC160은 크기만 다를 뿐 같은 시스템이라 할 수 있다.

슈퍼컴퓨팅 자원들이 연결된 네트워크 구성도는 그림 3과 같다. KISTI내에 있는 슈퍼컴퓨터들 간에는 800M bps의 HIPPI로 연결되어 있고 HPCnet으로의 연결은 Giga 스위치와 ATM스위치로 연결되어 있다. KISTI 내부에서는 800M HIPPI로 연결되어 있기 때문에 내부 자원을 활용한 메타컴퓨팅은 단일 시스템을 사용하는 것과 같이 빠르다. 전북대와 동명정보대의 슈퍼컴퓨터는 현재 HPCNET/KREONET을 이용해서 연결되며 45Mbps이다.

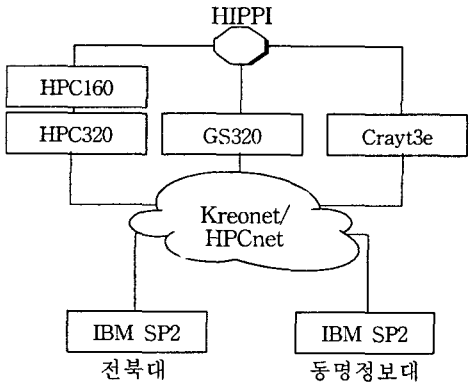

그림 3 네트워크 연결도

3.2 터보기계 유동해석 코드의 성능분석

표3는 성능분석에 적용된 컴퓨터와 CPU 갯수를 보여주고 있다.

CPU	CASE I		CASE II		CASE III			
	HPC 320	GS 320	IBM SP2	GS 320	Cluster	IBM SP2	HPC 320	GS 320
1	1		1			1		
2	1	1	1	1				
4	2	2	2	2	1	1	1	1
7	3	4	2	5	1	1	2	3
14	4	10	2	12	1	2	4	7

표 3 CPU 당 병렬 작업 수

전체 계산량을 28블록으로 고정한 상태에서 CPU 개수를 늘려가면서 계산시간을 측정하였으며, 계산시간은 iteration 수가 100에 이를 때까지의 시간으로 하였다.

보유 기관	시스템	시스템 구조	프로세서	계산성능 (FLOPS)	노드 수	노드당 프로세서 수	메모리 크기	디스크 크기	
								내장	외장
KISTI	HPC320	SMP 8 Node Cluster	Alpha EV67 (667 MHz)	42.7 Gflops (1.3G/CPU)	8	4	32G	145.6 G	360G
	HPC160	SMP 4 Node Cluster	Alpha EV67 (667 MHz)	21.35 Gflops (1.3G/CPU)	4	4	16G	72.8G	120G
	GS320	SMP	Alpha EV67 (729 MHz)	46.8 Gflops (1.46G/CPU)	1	32	34G		150G
	CRAY T3E	MPP	DEC Alpha 21264 (450 MHz)	115 Gflops (1G/CPU)	128		4G		240G
전북대	IBM SP2	SMP 32 Node Cluster	RISC Chip Power3 (200MHz)	51.2 Gflops (0.8G/CPU)	32	2	24G		726G
동명 정보대	IBM SP2	SMP 32 Node Cluster	RISC Chip Power3 (200MHz)	51.2 Gflops (0.8G/CPU)	32	2	24G		726G

표 2 슈퍼컴퓨팅 자원 사양

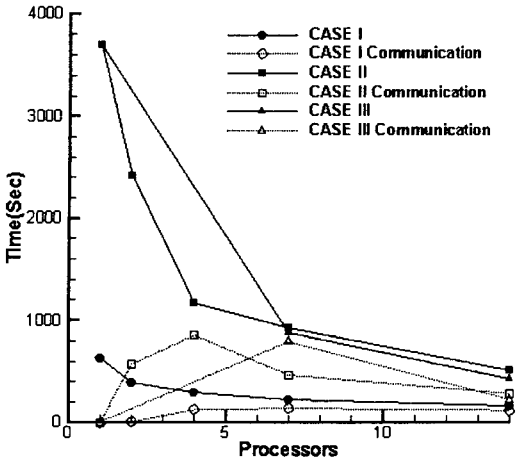

그림 4 CPU 증가에 따른 계산시간

(1) HPC320, GS320 간의 계산수행(CASE I)

CASE I 에서는 LAN환경에 있는 두 슈퍼컴퓨터를 사용해서 계산을 수행하였다. 그림4 에서 알 수 있듯이 CPU의 갯수가 늘어남에 따라 계산 시간이 감소하는 것을 알 수 있다. 그러나 각각의 컴퓨터가 비슷한 성능을 가진 상대적으로 빠른 CPU를 가지고 있기 때문에 CPU수가 증가함에 따라 통신시간이 차지하는 비중이 계산시간에 비해 커짐을 알 수 있다.

(2) IBM SP2, GS320 간의 계산수행(CASE II)

CASE II 에서는 전북대 IBM SP2와 KISTI의 GS320을 사용해서 계산하였다. IBM SP2의 속도가 상대적으로 느리기 때문에 CPU 갯수의 증가에 따라 계산시간의 감소가 현저히 나타남을 알 수 있다. 통신시간을 나타내는 그림4에서 특징적인 면은 CPU수가 4개까지는 통신시간이 증가하지만 이후 CPU개수가 증가함에 따라 통신시간이 감소함을 알 수 있다. 이는 CPU수를 늘리더라도 IBM SP2의 CPU개수를 2개로 고정했기 때문에 발생한 현상으로 CPU 속도가 빠른 GS320에 보다 많은 CPU 선정과 작업량이 분배됨으로써 대기 시간이 상대적으로 감소했기 때문이다.

(3) PC Cluster, IBM SP2, HPC320 그리고 GS320으로 이루어진 Grid 상의 계산수행(CASE III)

CASE III 에서는 MPICH-G로 구동 될 수 있는 Grid상의 컴퓨터를 총 동원하여 계산을 수행하였다. 그림4에서 보듯이 CPU수가 증가함에 따라 계산시간의 감소가 현저히 나타남을 알 수 있다. 하지만 많은 수의 컴퓨터 사용으로 인한 통신시간과 대기 시간의 증가가 발생하고 있음을 알 수 있다.

본 테스트의 경우는 PC Cluster 이외에 세대의 슈퍼컴퓨터를 사용하였으며 PC Cluster와 분산된 슈퍼컴퓨터를 하나로 묶어서 계산을 수행했다는 점에서 의미를 부여할 수 있다.

4. 결론

본 연구에서는 국내 슈퍼컴퓨팅 자원들을 활용하여 Globus기반 그리드 환경을 구축하였고 전산유체역학 프로그램을 이용하여 테스트하였다. 테스트의 결과가 말하듯이 유휴 컴퓨팅 자원들을 통합하여 사용하면 향상된 성능을 얻을 수 있다. 그리고, 더 나은 성능 향상을 위해서는 컴퓨팅 자원간의 통신속도의 향상이 필수적임을 보여주고 있다. 향후연구로 컴퓨팅 자원간 안정된 통신속도 유지를 위한 연구가 필요하고 현재는 사용자들이 유휴자원을 검색하고 사용하는 불편함이 있는데 이를 해결하는 연구가 필요하다.

참고문헌

[1] I. Foster, C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit" Intl. J. Supercomputer Applications, 11(2):115-128, 1997

[2] I. Foster, etc., "Software Infrastructure for the I-WAY High Performance Distributed Computing Experiment.", Proc. 5th IEEE Symposium on HPDC. pg. 562-571, 1997.

[3] I. Foster, etc., "The Globus Project: A Status Report" Intl. J. Supercomputer Applications, 11(2): 115-128, 1997

[4] I. Foster, etc., "A Security Architecture for Computational Grids." Proc. 5th ACM Conf. on Computer and Communications Security Conference, pg. 83-92, 1998.

[5] I. Foster and C. Kesselman (eds.) "The Grid: Blueprint for a new Computing Infrastructure" Morgan Kaufmann Publishers, 1998

[6] K. Czajkowski, etc., "Grid Information Services for Distributed Resource Sharing." Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, August 2001