

그리드 시스템 META의 구현을 위한 환경 구축

이건우*, 지정훈, 우균
부산대학교 컴퓨터공학과
e-mail:leoric99@pusan.ac.kr

Environment Construction for Implementation of Grid System META

Gun-Woo Lee*, Jung-Hoon Ji, Gyun Woo
Dept of Computer Engineering, Pusan National University

요 약

본 논문에서는 CFD(Computational Fluid Dynamics) 프로그램의 효율적인 수행을 위한 그리드 시스템 META(Metacomputing Environment using Test-run Application)의 구현을 위한 환경 구축에 대해서 기술한다. META는 그리드 시스템을 기반으로 하기 때문에 META의 구현을 위해서는 먼저 그리드 시스템 환경을 구축하여야 한다. 이를 위해 PBS(Portable Batch System), GT4(Globus Toolkit4), MPI(Message Passing Interface), PVM(Parallel Virtual Machine), PVMmaker 총 5개의 프로그램이 필요하다. PBS는 각 노드로의 작업 분할을, GT4는 그리드 시스템 미들웨어 역할을, MPI와 PVM, PVMmaker는 병렬 프로그래밍 수행 및 컴파일을 위해서 사용이 된다.

1. 서론

최근 급격한 네트워크 환경의 발달과 복잡한 계산의 보다 빠른 처리를 위해 그리드(Grid)[1] 시스템의 활용도가 매우 높아지고 있다. 그리드는 지리적으로 또 조직적으로 분산되어 있는 이 기종(Heterogeneity)의 자원, 데이터, 장비 등을 공유하게 하여 사용할 수 있게 해주는 고성능 컴퓨팅과 분산 처리를 위한 플랫폼이다.

본 논문에서 구현하고자 하는 META(Metacomputing Environment using Test-run of Application)[2]는 이러한 그리드 시스템을 기반으로 하며 네트워크를 통해 연결된 컴퓨팅 자원들 중 CFD(Computational Fluid Dynamics) 프로그램을 보다 효율적으로 수행하기 위한 가장 적절한 컴퓨팅 자원을 자동으로 선택한다. META를 위해서는 그리드 시스템을 기반으로 병렬 처리를 위한 환경을 구축해야 한다.

이를 위해 본 논문에서는 META의 구현[3]을 위한 환경 구축 방법에 대해서 기술한다. 2절에서는 환경 구축을 위해서 필요한 프로그램인 PBS, GT4, MPI PVM, PVMmaker에 대해서 기술하고, 3절에서

는 각 프로그램들을 설치하는 과정과 방법에 대해서 기술한다. 4절에서는 결론을 내리며 향후 연구 계획 및 방향에 대해서 살펴본다.

2. 환경 구축을 위한 프로그램

이 절에서는 META의 환경 구축에 필요한 프로그램들에 대해 기술한다. META는 CFD 프로그램의 수행을 위해 고성능 컴퓨팅과 분산 처리를 기반으로 한다. 환경 구축을 위해 필요한 프로그램은 PBS, GT, MPI, PVM, PVMmaker로 전체 System의 구성은 그림1과 같다.

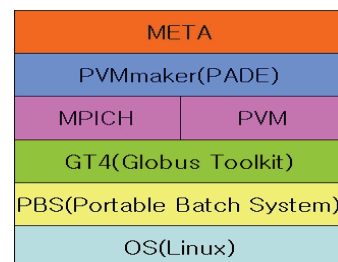


그림 1 Overview META

2.1. PBS

PBS(Portable Batch System)는 스케줄러(Scheduler) 또는 큐잉 시스템(Queueing System)이라고 하는 작업 관리 프로그램 중 널리 사용되는 것들 중 하나이다. 다수 노드로 이루어진 클러스터 또는 슈퍼컴퓨터에서 작업 관리 프로그램은 사용자가 수행시키는 작업을 각 노드로 배분시키는 역할을 한다. 일반적으로 많이 사용되는 PBS로는 OpenPBS[4]와 TORQUE[5]가 있다. 본 논문에서는 이들 중 TORQUE 2.1.6 버전을 사용한다.

2.2. GT4

GT4(Globus Toolkit4)[6]는 그리드 시스템 구축과 어플리케이션 개발에 가장 많이 쓰이는 그리드 미들웨어 도구로 그리드 환경에 필요한 대부분의 서비스를 지원한다. GT4는 크게 그리드 보안 기반(GSI), 자원 할당 관리(GRAM), 자원 정보 서비스(MDS), 데이터 관리 서비스를 지원하며 각 서비스마다 필요한 기능을 지원하는 컴포넌트들이 있다. 본 논문에서는 4.0.3 버전을 사용하였다.

2.3. MPI

MPI(Message Passing Interface)는 병렬 처리 프로그램 중 각 노드의 프로세스 사이에 정보와 명령 등 각종 메시지를 주고받는 방법을 정의한 규약이다. MPI는 1995년에 MPI-1.1이 그리고 1997년에 MPI-2가 발표되었다. MPI는 그 자체가 프로그램이 아니라 표준 규약이며 이 규약에 따라 실제 병렬 프로그램 메시지 전달을 지원하는 프로그램은 MPICH[7]와 LAM-MPI[8], MPI/Pro[9] 등이 있다. META를 위한 환경 구축을 위해 본 논문에서는 미국 Argonne 국립 연구소에서 만든 MPICH1과 MPICH2 중 MPICH1 1.2.7버전을 사용한다.

2.4. PVM

PVM(Parallel Virtual Machine)은 MPI와 더불어 많이 쓰이는, 병렬 프로그램을 위한 메시지 전달 규약 중 하나이다[10]. PVM은 여러 노드를 응용 프로그램이 하나의 가상 기계로 볼 수 있도록 구성한다. PVM의 장점 중 하나는 네트워크를 통해 외부에서 보는 pvmd 데몬이 OS에 관계없이 동일하게 보이도록 규격화 되어 있어서 다른 종류, 다른 개수의 프로세서를 갖거나 윈도우, 리눅스, 유닉스 등 다른 OS를 가진 노드들을 하나의 가상 머신으로 쉽게 구성할 수 있다. 본 논문에서는 PVM 3.4.5 버전을 사용한다.

2.5. PVMmaker

PVMmaker는 PADE(Parallel Application Development Environment)[11]에 포함되어 있는 패키지이다. PADE는 네트워크상에 연결된 이종(heterogeneous) 컴퓨터에서 병렬 프로그램의 컴파일을 위한 프로그램이다. PVMmaker는 PVM 작업을 위해 압축된 파일을 선택된 각 노드로 보내고, 호스트 노드에서는 임시 디렉토리에 받은 파일의 압축을 풀어 작업을 컴파일 한다. PVMmaker는 주로 WAMM(Wide Area Metacomputer Manager)에서 사용된다. 본 논문에서는 PADE 1.4.1 버전을 사용한다.

3. 환경 구축

2절에서는 META의 환경 구축을 위해 필요한 프로그램들에 대해서 살펴보았다. META의 환경 구축에 사용된 PC 사양과 설치한 OS는 표1과 같다.

PC Name	사양	OS
node1	P4-3.0, 512M, 80G,	Redhat Fedora Core4
node2	P4-3.0, 512M, 200G	Redhat Fedora Core4
node3	P4-2.4, 512M, 80G	Redhat Fedora Core4
node4	P4-2.4, 512M, 80G	Redhat Fedora Core4

표1 각 PC 사양과 OS

3.1. TORQUE 설치

TORQUE의 설치에는 크게 Head Node와 Compute Node 2가지로 나뉜다. 기본적으로는 Head Node는 *configure*→*make*→*make install*의 단계의 일반적인 설치가 가능하지만 보다 손쉽게 설치하기 위해서는 마지막 *make install* 대신 *make package*를 수행하여 패키지 형태로 만들어 설치를 할 수 있다. 패키지로 만들게 되면 *client*, *mom*, *doc*, *gui*, *devel*, *server* 총 6개의 파일이 생성된다. Compute Node는 만들어진 6개의 패키지 파일 중 *mom*, *client*의 순서로 설치를 하면 */var/spool/torque* 라는 디렉토리가 생성된다.

설치가 완료되면 Head Node에는 */usr/local/sbin/pbs_server -t create* 명령어를 실행하여 데몬 시작을 위해 설정 정보를 가지는 DB를 생성하여야 한다. 데몬 실행 후 설정 사항은 *qmgr* 명령어를 통하여 설정이 가능하다. 마지막으로 *\$TORQUE_HOME/pbs_server/nodes* 란 파일을

생성하여 각 Compute Node의 hostname을 기록한다. Compute Node에서는 \$STORQUE_HOME/mom_priv/config 파일에 Head Node의 hostname을 기록한다. 그림2는 기본설정을 완료한 뒤 현재 연결된 Compute node들의 상태 정보다.

```

cluster01 - Xshell
Last login: Thu Mar 1 17:54:26 2007 from 164.125.34.105
[giobus@node1 ~]$ pbsnodes -a
node2
state = free
np = 1
nptype = cluster
status = ophys=linux,uname=linux node2.cse.pusan.ac.kr 2.6.11-1.1369_FC4mp #1 SMP Thu Jun 2 23:08:39 EDT 2005 1686, sessions=2229 1662 2827, nsessions=3, users=3, idletime=1863118, totmem=1566632kb, availmem=1455028kb, physmem=514384kb, ncpus=2, loadave=0.00, netload=337782354, state=free, jobs=? 0, rectime=1174451462
node3
state = free
np = 1
nptype = cluster
status = ophys=linux,uname=linux node3.cse.pusan.ac.kr 2.6.11-1.1369_FC4 #1 Thu Jun 2 22:55:56 EDT 2005 1686, sessions=2080 1501 2722, nsessions=3, users=4, idletime=1863100, totmem=1567572kb, availmem=1444952kb, physmem=515324kb, ncpus=1, loadave=0.00, netload=33626741, state=free, jobs=? 0, rectime=1174451456
node4
state = free
np = 1
nptype = cluster
status = ophys=linux,uname=linux node4.cse.pusan.ac.kr 2.6.11-1.1369_FC4 #1 Thu Jun 2 22:55:56 EDT 2005 1686, sessions=2107 1528 2686, nsessions=3, users=3, idletime=1863087, totmem=1567572kb, availmem=1449400kb, physmem=515324kb, ncpus=1, loadave=0.00, netload=33528935, state=free, jobs=? 0, rectime=1174451453
[giobus@node1 ~]$
164.125.34.111:3939@연결됨. SSH2:aes128-cbc,hmac-sha1:zlib linux 90x30 NUM
  
```

그림 2 Compute Node State

3.2. GT4 설치

GT4를 설치하기 위해서는 먼저 다음의 소프트웨어가 설치되어 있어야 한다.

- J2SE 1.4.2 이상
- Apache Ant 1.5.1 이상
- C Compiler(gcc 3.2, 3.2.1, 2.95.x는 제외)
- GNU tar, sed, Make
- zlib 1.1.4 이상
- sudo
- JDBD database(PostgreSQL은 7.1 이상)

GT4의 설치하는 Globus 공식 홈페이지에서 제공하는 설치 매뉴얼을 참고하면 별 무리 없이 설치를 할 수 있다. 버전이 4.x대로 업데이트 되면서 이전 버전에 비해 설치 시간이 매우 줄어들었고 과정도 쉬워졌다. GT4의 설치 과정 중 GridCA Certification의 발급 및 설정 방법은 크게 2가지가 있다. 먼저 첫 번째는 KISTI나 서울대, KAIST와 같은 곳에서 운영하는 인증 서버를 이용하는 방법이고, 두 번째는 자체 인증 서버를 구축하는 것이다. 자체 서버는 SimpleCA를 이용해서 구축할 수 있다. SimpleCA를 설치하여 인증 서버를 구축한 뒤 사용자 인증을 할 때 인증키의 기간은 기본적으로 12시간이므로 필요에 따라 인증 기간을 변경하면 된다. GT4를 설치할 때 각 Node간의 작업 분배를 위해 SSH나 RSH를 통해 암호 입력 없이 각 노드간

의 접속이 가능하게 설정을 해야 한다. SSH의 경우 인증키를 이용하면 이런 설정은 쉽게 할 수 있다 [12].

GT4의 설치가 다 되면 *globus-start-container* 명령을 실행시켜 그림3과 같이 총 51개의 서비스가 정상적으로 작동하는지를 테스트 할 수 있다.

```

cluster01 - Xshell
[2]: https://164.125.34.111:8443/wsrf/services/AdminService
[3]: https://164.125.34.111:8443/wsrf/services/AuthzCalloutTestService
[4]: https://164.125.34.111:8443/wsrf/services/CASService
[5]: https://164.125.34.111:8443/wsrf/services/ContainerRegistryEntryService
[6]: https://164.125.34.111:8443/wsrf/services/ContainerRegistryService
[7]: https://164.125.34.111:8443/wsrf/services/ContainerService
[8]: https://164.125.34.111:8443/wsrf/services/DefaultIndexService
[9]: https://164.125.34.111:8443/wsrf/services/DefaultIndexEntryService
[10]: https://164.125.34.111:8443/wsrf/services/DefaultIndexServiceEntry
[11]: https://164.125.34.111:8443/wsrf/services/DelegationFactoryService
[12]: https://164.125.34.111:8443/wsrf/services/DelegationService
[13]: https://164.125.34.111:8443/wsrf/services/DelegationTestService
[14]: https://164.125.34.111:8443/wsrf/services/InMemoryServiceGroup
[15]: https://164.125.34.111:8443/wsrf/services/InMemoryServiceGroupEntry
[16]: https://164.125.34.111:8443/wsrf/services/InMemoryServiceGroupFactory
[17]: https://164.125.34.111:8443/wsrf/services/IndexService
[18]: https://164.125.34.111:8443/wsrf/services/IndexServiceEntry
[19]: https://164.125.34.111:8443/wsrf/services/ManagedExecutableJobService
[20]: https://164.125.34.111:8443/wsrf/services/ManagedJobFactoryService
[21]: https://164.125.34.111:8443/wsrf/services/ManagedJobService
[22]: https://164.125.34.111:8443/wsrf/services/ManagementService
[23]: https://164.125.34.111:8443/wsrf/services/ManagementService
[24]: https://164.125.34.111:8443/wsrf/services/NotificationConsumerFactoryService
[25]: https://164.125.34.111:8443/wsrf/services/NotificationConsumerService
[26]: https://164.125.34.111:8443/wsrf/services/NotificationTestService
[27]: https://164.125.34.111:8443/wsrf/services/PeristenceTestSubscriptionManager
[28]: https://164.125.34.111:8443/wsrf/services/ReliableFileTransferFactoryService
[29]: https://164.125.34.111:8443/wsrf/services/ReliableFileTransferService
[30]: https://164.125.34.111:8443/wsrf/services/ReliableFileTransferFactoryService
[31]: https://164.125.34.111:8443/wsrf/services/SimpleAuthzService
[32]: https://164.125.34.111:8443/wsrf/services/SecureCookieService
[33]: https://164.125.34.111:8443/wsrf/services/SecurityTestService
164.125.34.111:3939@연결됨. SSH2:aes128-cbc,hmac-sha1:zlib linux 112x33 NUM
  
```

그림 3 Globus Toolkit Service 실행

서비스가 정상적으로 실행되면 PBS 플러그인을 설치해야 한다. 설치 과정은 다음과 같다.

```

[giobus@node1]make gt4-gram-pbs postinstall
[giobus@node1]gpt-postinstall
[giobus@node1]cd $GLOBUS_LOCATION/setup/globus
[giobus@node1]./.setup-globus-job-manager-pbs --remote-shell=ssh(or rsh)
  
```

테스트를 위한 샘플 소스는 GT4 홈페이지에서 찾을 수 있다. Head Node에 설치된 GT4의 library 디렉토리만 Compute Node에 동일한 위치에 복사해 해주면 된다.

3.3. MPICH 설치

MPICH는 일반적인 *configure→make→make install* 과정대로 설치를 한다. 설치 후 각 노드에서 \$MPICH_HOME/machines.LINUX 파일에 각 노드의 hostname을 기록한다. 설치 후 \$MPICH_HOME/example 디렉토리에 있는 샘플 소스 파일들을 이용해서 테스트 할 수 있다.

3.4. PVM 설치

PVM은 사용자 환경설정 파일(.bash_profile)에 설치 경로와 library 경로, PVM이 설치된 Architecture Machine의 type을 기록해야 한다. PVM 데몬을 실행시킨 뒤 그림4와 같이 각 노드를 추가시킬 수 있다.

```

cluster01 - Xshell
파일(F) 편집(E) 보기(V) 도구(D) 윈도우(W) 도움말(H)
주소(D)
~# pvm
-rw-r--r-- 1 globus globus 47 8월 29 1997 pvm.cad
-rw-r--r-- 1 globus globus 54 2월 8 2001 pvm.cygvwin.bat
-rwxr-xr-x 1 globus globus 2388 2월 15 2000 pvmad
-rw-r--r-- 1 globus globus 36 6월 27 1997 pvmad.bat
-rw-r--r-- 1 globus globus 52 8월 29 1997 pvmad.cad
-rwxr-xr-x 1 globus globus 7098 2월 4 2004 pvmgetarch
-rw-r--r-- 1 globus globus 30 8월 29 1997 pvmgetarch.cmd
-rwxr-xr-x 1 globus globus 347 5월 20 1997 pvmgetarch
-rwxr-xr-x 1 globus globus 1055 9월 24 1996 xvpm
[globus@node1 lib]$ pvm
pvm already running.
pvm> add node2
add node2
1 successful

pvm> conf
conf
4 hosts, 1 data format
table
  HOST      DTID      ARCH      SPEED      PSIG
node1.cse.pusan.ac.kr 40000 LINUX1386 1000 0x00408841
node4      80000     LINUX     1000 0x00408841
node3      100000    LINUX     1000 0x00408841
node2      1c0000    LINUX     1000 0x00408841
pvm>
164.125.34.111:39390에 연결됨. SSH2:aes128-cbc:hmac-sha1:zlib linux 90x25 NUM

```

그림 4 PVM node 추가

3.5. PVMmaker 설치

PVMmaker를 설치하기 위해서는 pade 내의 caml_utils 디렉토리에 있는 *Makefile.aimk* 내용 중 *PVM_ROOT*와 *PVM_ARCH*에 대해서 수정해야 한다. 설치가 완료되면 *caml_utils/LINUX*라는 디렉토리가 생성되며 안에 *pvmmaker*가 생성된다. 설치된 *caml_utils* 디렉토리를 각 노드에 동일한 위치에 복사해 준다. 함께 제공되는 *examples* 디렉토리 내의 예제 소스 파일들을 이용해서 컴파일 테스트를 할 수 있다.

4. 결론

네트워크 속도가 향상되고 우리가 일반적으로 사용하는 PC의 사양이 더욱 높아짐에 따라 이를 활용하는 그리드 시스템의 필요성과 활용도는 더욱 높아지고 있다. META의 목적은 CFD 사용자가 쉽게 그리드 시스템으로 이루어진 슈퍼컴퓨터를 사용할 수 있게 하는 것이다.

본 논문은 PBS, GT4, MPICH, PVM, PVMmaker를 이용하여 그리드 시스템 META의 구현을 위한 환경 구축 방법과 과정에 대해서 살펴보았다.

향후 연구로는 META의 설계 및 구현을 바탕으로 NS-2(Network Simulation Version 2)를 이용하여 META의 시뮬레이션 모델을 만드는 것이다. 실제 CFD 프로그램 수행에 있어 META의 효율성을 시뮬레이션을 통해 미리 확인할 수 있다면 META의 활용성이 더욱 증대될 것이다.

참고문헌

- [1] Ian Foster, Carl Kesselman, "The Grid2: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, 2003
- [2] 강경우, "CFD 프로그램 개발자를 위한 메타컴퓨

팅 시스템", 산학기술성공학회논문지, Vol. 2, No. 1, pp. 43~51, 2001.

- [3] 강경우, 우균, "분산 환경에서 CFD 분석 프로그램 수행을 위한 그리드 시스템 META 설계 및 구현", 정보처리학회논문지A, 제13-A권, 제6호, 2006년 12월, pp.553~540
- [4] OpenPBS Homepage, <http://www.openpbs.org>
- [5] TORQUE Homepage, <http://www.clusterresources.com/>
- [6] Globus Homepage, <http://www.globus.org>
- [7] MPICH Homepage, <http://www.mcs.anl.gov/mpi/mpich>
- [8] LAM-MPI Homepage, <http://www.lam-mpi.org>
- [9] MPI/PRO Homepage, <http://www.mpi-softtech.com>
- [10] PVM Homepage, <http://www.csm.ornl.gov/pvm>
- [11] PADE Homepage, <http://math.nist.gov/mcsd/pade>
- [12] SSH 인증키 사용하기, <http://kldp.org/KoreanDoc/html/SSH-KLDP/x80.html>