xCAT 1.2.0을 이용한 테라플롭스급 클러스터 설치가이드

Installing Teraflops-scale Linux Cluster System using xCAT ver. 1.2.0

홍태영, 홍정우, 차광호 편저

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

xCAT 1.2.0을 이용한
테라플롭스급 클러스터 설치가이드

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

이 책의 무단 복제 및 무단 전제를 금합니다.

Published by Supercomputing Center, KISTI
Printed in Korea

ISBN 89-5884-070-6 93560

이 책에 대한 의견이나 조언을 주시고자 할 때 그리고 오탈자나 버그 등을 발견했을 경우, 언제든지 다음의 저자의 메일주소로 연락하시기 바랍니다.

tyhong@kisti.re.kr
jwhong@kisti.re.kr
khocha@kisti.re.kr

머리말

고성능 마이크로프로세서와 고성능 네트워크의 급속한 발전과 더불어, 클러스터 기반의 대형 병렬컴퓨터의 안정성 및 확장성이 입증되면서, 클러스터 시스템에 대한 관심이 증가하고 있다. 특히 기존의 슈퍼컴퓨터에 비해 뛰어난 가격대 성능과 시스템 구성의 유연성과 확장성 때문에 최근 슈퍼컴퓨팅 분야에서 차지하는 비중이 날로 증가하는 추세이며(2004년 6월 현재 top500.org 리스트에서 약 58 % 정도를 차지함) 기존의 MPP, SMP 기반의 슈퍼컴퓨터 시장을 급속도로 대체해 나가고 있다. 하지만 클러스터의 대형화가 일반화 되면서 초기의 Beowulf 클러스터 제작과 달리 설계, 설치, 및 관리에 있어 전문적인 지식과 이해가 요구되고 있으며, 클러스터 시스템 설계, 클러스터 미들웨어, 시스템 영역 네트워크, 그리고 병렬 파일시스템 등 관련 분야의 연구도 활발히 진행되고 있다. 이 책에서 소개한 xCAT은 중대형급의 고성능 컴퓨팅 플랫폼을 제작 및 운영하는 데 필요한 도구를 제공하며, 다양한 클러스터의 구성 하드웨어 및 소프트웨어를 효율적으로 셋업하는 데 필요한 거의 모든 기능을 가지고 있다. 하지만 이와 같은 기능을 최대한 활용하기 위해서는 많은 클러스터 구성 하드웨어 및 소프트웨어 문서 및 매뉴얼을 참조하여야 하며, 또한 xCAT의 설정 파일의 내용을 이해하고 작성하기 위해서 별도의 시간을 투자해야 한다. 이에 이 책에서는 이와 같은 복잡한 설치과정을 xCAT의 mini-howto 문서를 기반으로 일관되게 풀어서 작성하고자 했으며, xCAT의 정교하고 세련된 클러스터 설치과정을 보다 쉽게 풀어서 설명하고자 노력했다. 아무쪼록 이 책이 xCAT의 이러한 장점을 익히고자 하는 독자에게 조금이나마 도움이 되었으면 한다.

홍태영, 홍정우, 차광호
슈퍼컴퓨팅센터,
한국과학기술정보연구원

목차

1. 소개 1

2. 시스템 개요 3

3. 네트웍 7
   3-1. 계산네트웍 9
   3-2. 파일서비스 네트웍 11
   3-4 외부 네트워크 장비 13

4. 서버 및 노드 19
   4-1. 계산노드(256 대) 19
   4-2. 예비노드 (8대) 21
   4-3. 로그인 노드 (4대) 21
   4-4. 디버깅 노드 (4대) 21
   4-5. 파일서비스 노드 (8대) 22
   4-6. 관리노드 22
   4-7. 스케줄링 노드 22

5. SAN(storage area network) 23

6 Master 노드 리눅스 및 xCAT 설치 27
   6-1 redhat 7.3 설치 27
   6-2 xCAT 소스파일 /opt에 설치 27
   6-3 xCAT setup 27
   6-4 로그아웃/재로그인 27
   6-5 NTP 서버 설정 27
   6-6 xCAT 설정 28

7 계산노드의 펌웨어/바이오스 업데이트 및 설정 44
7-1 계산노드 시스템의 firmware 업데이트 44
7-2 BIOS 설정 업데이트 44

8 /etc/hosts 파일 설정 45
9 DNS 서버 설정 46
10 Non-collectable MACs를 /opt/xcat/etc/mac.tab에 추가 47
11 DHCP 서버 설정 48

12 클러스터의 Ethernet switch 설정 48
12-1 telnet을 이용한 스위치 접속 48
12-2 Privileged EXEC 모드로 전환 49
12-3 기존의 startup-config 내용을 tftp 서버에 백업하기 50
12-4 Global configuration 모드로 전환 50
12-5 Config-vlan 모드로 전환 50
12-6 VLAN 추가 51
12-7 (참고사항) 다른 방법 52
12-8 세팅 확인 53
12-9 설정 저장하기 54
12-10 원래대로 되돌리기 55

13 터미널 서버 셋업 57
13-1 케이블 연결 및 /etc/hosts 파일 확인 57
13-2 nodelist.tab 설정 57
13-3 DNS DB 업데이트(makedns) 58
13-4 터미널 서버 셋업 58

14 conserver 대몬 재 시작 59
15 stage boot image 생성 59

16 각 계산노드 수동으로 리부팅하기 60
17 /etc/dhcpd.con 업데이트 61
18 Management processor의 펌웨어 업데이트 61
19 각 계산노드 수동으로 리부팅하기 61
20 Management processor 설정 테스트 62
21 원격 전원 관리 테스트 63
22 리눅스 배포판 복사 63
23 xCAT의 post-installation 파일 복사 64
24 root의 SSH keys 생성 65
25 /etc/exports 파일업데이트 65

26 미리넷 GM RPM build 65
26-1 kernel 준비26-3 GM active mapper 지정 65
26-2 GM 소스를 RPM으로 빌드 66
26-3 GM active mapper 지정 67
26-4 IP over Myrinet 세팅 67

27 Kickstart Template 파일 수정 67
28 계산노드 설치 68
29 계산노드의 SSH hostkey 수집 69
30 계산노드의 system date 체크 70
31 (GM 2.0 이하인 경우) GM routes 생성 70
32 (선택사항) PBS와 Maui 설치 70
33 사용자 계정 추가 70
34 (선택사항) PBS/Maui 테스트 73

35 컴파일러와 라이브러리 설치 74
35-1 GNU compiler 설치 75

35-2 absoft 컴파일러 설치 75
35-2 Intel compiler 설치 75
35-3 PGI compiler 설치 76

36 로그인노드 MPICH-GM 설치 76
37 HPL benchmark 테스트 후 결과를 top500.org에 제출 78
38 Enjoy your cluster. 78

Appendix 78
A1 계산노드 전체 재설치 78
A2 시스템의 교체 후 계산노드의 재설치 78

1. 소개

xCAT은 IBM의 Egan Ford의 의해 2001년도부터 독자적으로 개발되기 시작하였으며, 핵심 설치 패키지를 공개 소프트웨어 형태로 제공하여 비 IBM 하드웨어에도 쉽게 사용할 수 있다. 하지만 xCAT의 모든 기능이 비 IBM 하드웨어에서 동작하는 것을 보장하지는 않으며, 이것은 클러스터 시스템의 구성 하드웨어의 종류와 규격이 너무도 다양하기 때문이다. xCAT을 여타 클러스터 설치도구와 비교할 때 장점으로는 클러스터 시스템에 관한 거의 모든 세부적인 설정을 설치 이전에 할 수 있도록 지원하며, 관리 프로세서 네트워크, KVM 및 직렬 네트워크 하드웨어를 비롯한 대부분의 클러스터 관리용 하드웨어의 설정 및 시스템의 소프트웨어 설치에 관한 거의 완벽한 자동화를 지원한다는 것 등을 들 수 있다. 하지만 이러한 자동화를 위해서 20개 이상의 설정파일을 통해 클러스터 시스템과 그 구성하드웨어의 특성을 정확히 기술하는 설정 작업을 해야 하며, 매우 많은 단계의 설치과정을 순서대로 진행해야 한다. 때문에 일반적으로 OSCAR나 Rocks와 달리 클러스터 설치과정을 익히는 것이 매우 어려우며, 64노드 이하의 중소형 범용 리눅스 클러스터의 경우 보다 간단한 설치 도구를 사용할 것을 권장한다. 하지만 xCAT의 폭넓은 하드웨어 및 소프트웨어 지원과 설치 자동화 및 중대형 클러스터 설치에 쉽게 사용될 수 있는 확장성 등은 공개소프트웨어 기반의 다른 클러스터 설치도구가 근접하기 어려운 장점이며, 이것은 xCAT이 IBM 하드웨어에 최적화 되어 있음에도 불구하고 많은 Beowulf 클러스터 사용자들에게 각광받는 이유가 되고 있다.

2. 시스템 개요

KISTI 슈퍼컴퓨팅센터에서 2004년에 새로이 도입된 256 노드 512 cpu 규모의 테라플롭스급 리눅스 클러스터 시스템은 2.867 Tflops 이론 성능과 768 GB의 총 메모리, 그리고 약 10 TB의 저장자원을 구비하고 있다.

이 테라 클러스터 시스템은 IBM의 클러스터 제품인 e1350을 기반으로 구축되었다. Pentium-IV (2.8GHz) 512 CPU 규모의 이론 성능 2.867TFlops의 테라플롭스급 클러스터로서, 계산노드(256개) 및 예비 노드(8개), 디버깅 노드(4개), 로그인노드(4개)는 IBM x335 서버 기반으로, 파일서비스 노드(8개) 및 관리 노드(1개) 및 스케쥴링 노드(1개)는 IBM x345 서버로 구성되어 있다.

계산 네트웍은 6개의 M3-E128 미리넷 스위치를 이용하여 Fat-tree 형태로 구성되었으며, 파일 서비스 네트웍은 두개의 Cisco 4006 백본 기가비트 스위치를 star topology 형태의 기기비트 이더넷으로 구성되어 있다.

관리 네트웍으로는 e1350에서 제공하는 serial 포트와 iTouch Terminal sever 9개를 이용한 Serial 네트웍, IBM의 KVM daisy chaining 기술인 C2T를 이용한 KVM 네트웍과 Management Processor 네트웍으로 구성되며 네트웍 스위치로는 Cisco 3550(1개), Cisco 3508G(1개)를 사용하고 있다.

스토리지는 두개의 FAStT900 서버를 사용하고 있으며, 각각 디스크박스로 Exp700을 3개씩 사용하고 있다. 이 스토리지는 두 개의 16 port IBM SAN 스위치를 거쳐 파일서비스노드에 연결되어 있다.
전체적인 랙의 배치도는 다음그림을 참조하기 바란다.

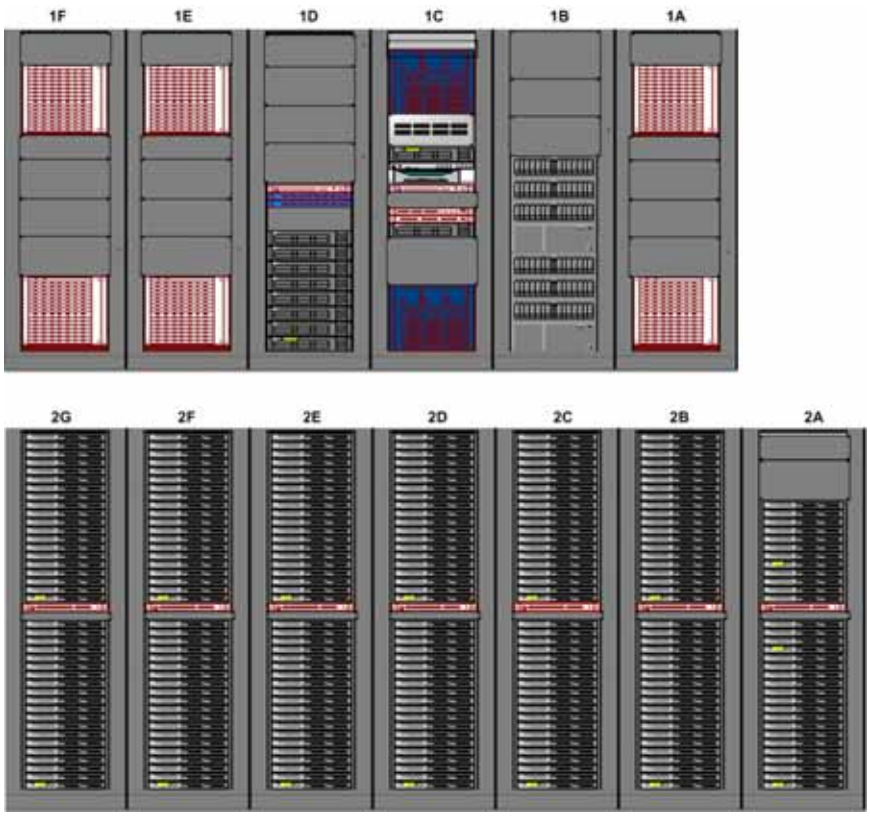

그림 5

2B ~ 2G에 있는 랙에는 각각 40개의 계산노드와 가운데에 iTouch 8040 terminal server가 위치한다. 따라서 총 40 X 6 =240개의 노드가 위치한다. 2A 랙은 이 랙에는 16개의 계산노드와 로그인 노드 4개 그리고 4개의 디버깅노드와 8개의 예비노드가 위치하고 있다(아래 그림 참조).

뒷열의 구성을 살펴보면 1A, 1E, 1F 랙에는 각각 2개의 M3-E128 미리넷 스위치가 들어 있으며, 1B 랙에는 FAStT900 스토리지 서버와 디스크 박스 EXP 700 3개로 구성된 스토리지 장비 2대가 위치한다. 1D 랙에는 8개의 스토리지(파일서비스) 노드와 2개의 Fibre channel

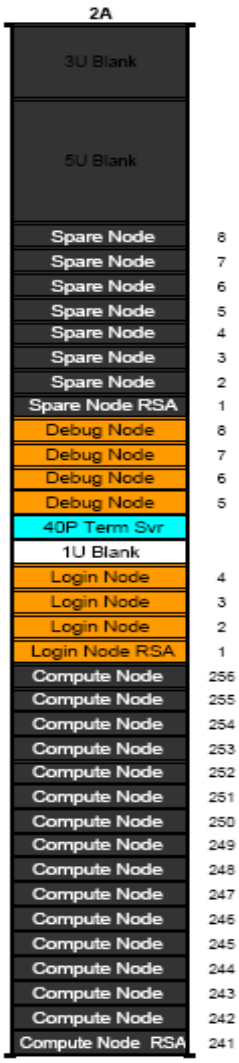

그림 6

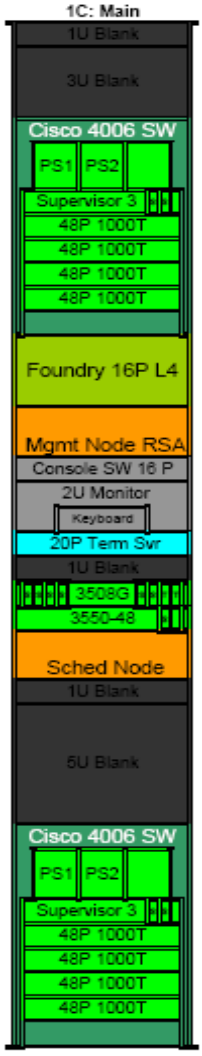

그림 7

스위치가 위치한다.

앞의 랙들과 달리 1C랙(아래 그림 참조)은 좀 더 복잡한 구성을 가지는 데, 파일서비스 I/O를 담당할 Cisco 4006 스위치 2개와 스케쥴링노드, 관리노드, 그리고 관리네트워크를 위한 장비들인 콘솔 스위치(KVM 스위치), 모니터/키보드, Cisco 3508G 그리고 Cisco 3550 스위치가 위치한다.

그리고 각 x345, x335노드들이 하나라도 있는 랙(2A ~ 2F, 1C, 1D)에는 iTouch terminal server가 배치되어 있다(각 랙의 가운데(21U)에 위치함).

3. 네트웍

전체네트웍의 구성은 그림 4와 같다. 전체네트웍은 서비스 종류에 따라 미리넷 기반의 계산전용 네트웍, 기가비트 이더넷 기반의 파일 I/O 네트웍 및 관리 네트웍으로 구성된다.각 네트웍을 물리적으로 혹은 VLAN(Virtual LAN) 구성을 통해 분리하여 불필요한 bandwidth의 남용 및 네트웍간의 간섭을 최소화 했다.

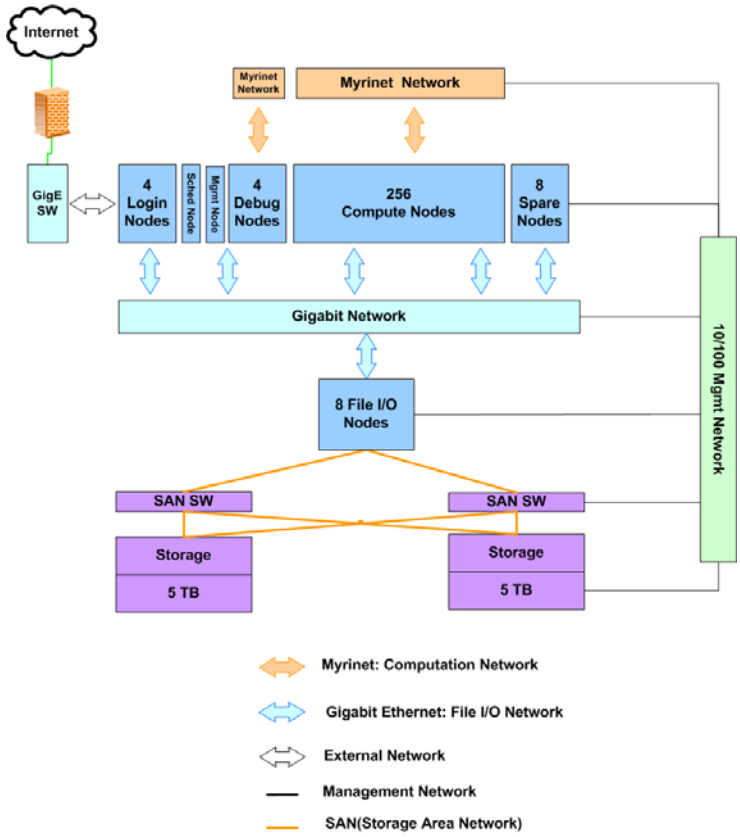

그림 8

이 그림을 통해 미리넷과 기가비트 네트웍 그리고 관리네트웍을 쉽게 구별할 수 있다. 관리 네트워크용 케이블과 스위치를 포함하

여 좀 더 자세히 나타내면 시스템 구성도는 그림 5와 같은 형태로 표현할 수 있다. VLAN이 미리넷, 기가비트 네트웍, 그리고 관리 네트웍에 따라 서로 다름을 확인할 수 있다.

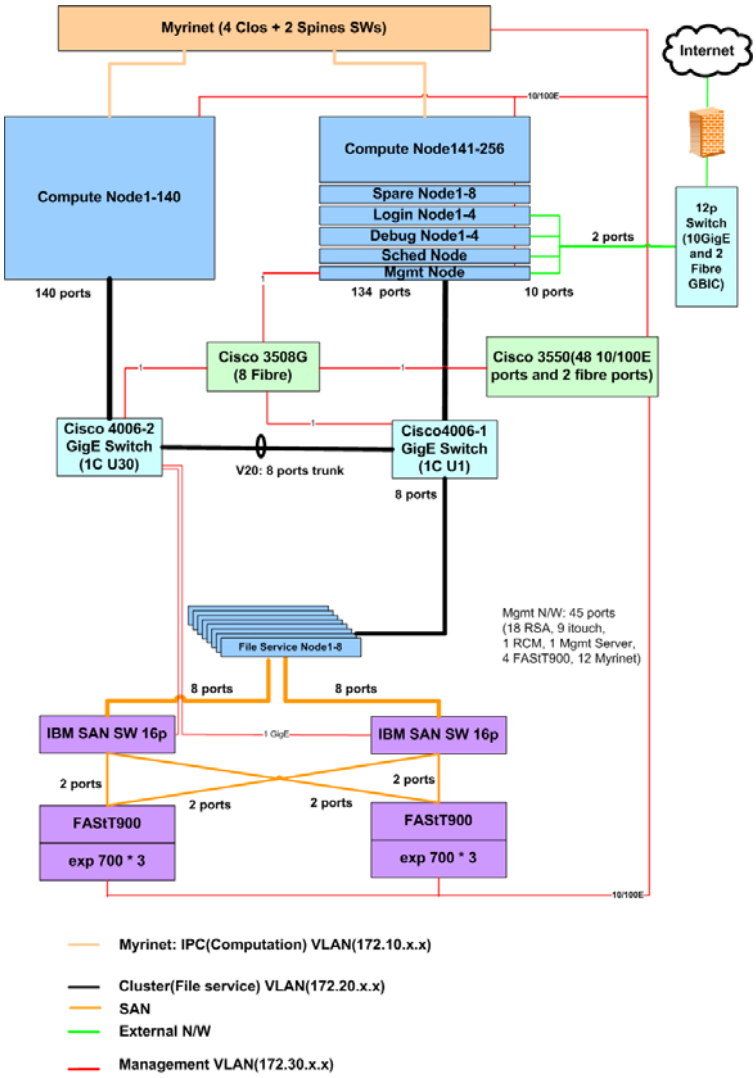

그림 9

위의 그림 5에서 관리네트웍을 제외하고 미리넷 계산네트웍, 파일 I/O 네트웍, SAN, 그리고 외부 연결네트웍을 다시 다음과 같이 보기 쉬운 형태로 나타낼 수 있다.

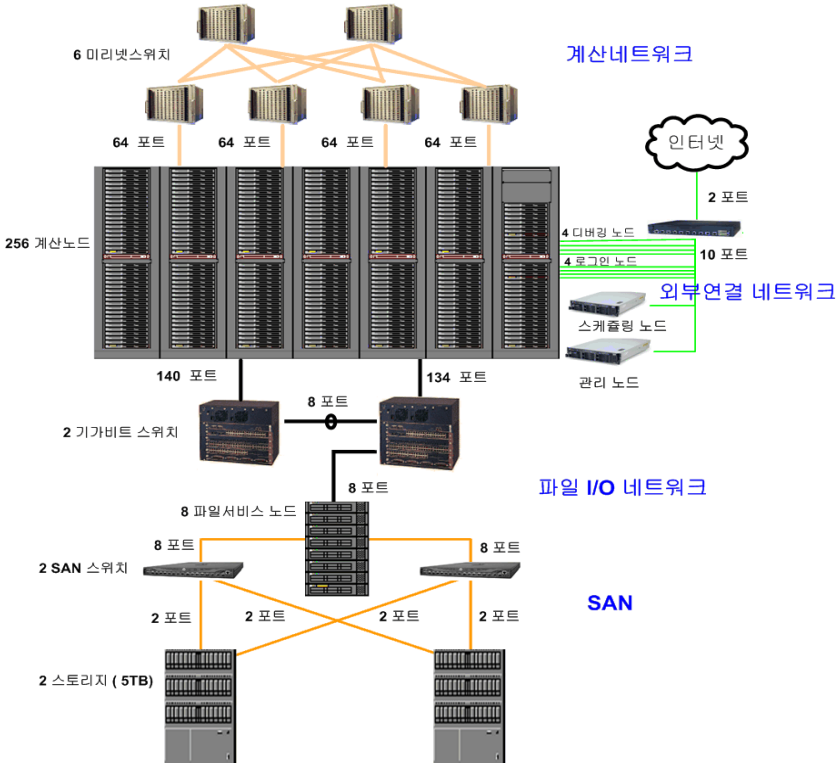

그림 10

3-1. 계산네트웍

계산네트웍은 초고속 SAN(System Area Network) 네트워크 중 하나로서, 초기 모델에 비해 성능과 안정성이 매우 향상된 myrinet 스위치 기반으로 구성하였다. 일반적으로 256노드 클러스터가 Full Bisection Bandwidth를 지원하는 fat-tree 구조를 이루

도록 구성하기 위해 필요한 128포트 myrinet 스위치의 최소한의 수는 6개이며, 이를 위해 128포트 M3-E128모델의 Myrinet 스위치 6개를 사용했다. 이중 4개는 Clos 스위치 타입으로, 2개는 Spine 스위치 타입으로 조립되어야 fat-tree 형태로 구성할 수 있다.

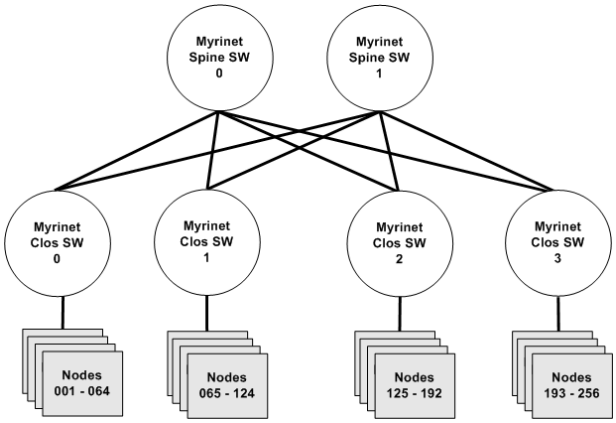

그림 11

Clos 스위치는 8포트 Switch 라인카드 8개와 8포트 Spine 라인카드 8개로 구성되며 Spine 스위치는 8포트 Spine 라인카드 16개로 구성된다. 각 Clos 스위치의 Switch 라인카드의 포트들(8 * 8)은 64개의 계산노드의 Myrinet Host Interface에 연결된다. 이를 위해 256개의 계산노드는 Lanai PCI-XD type의 Myrinet PCI 카드를 장착하고 있다. Netpipe를 이용한 성능 측정에서 64bytes message의 경우, 약 8.5usec 의 latency를 보였으며, 1MB message에 대해 약 1716Mbps 의 대역폭이 측정되었다.

계산 네트웍의 개략적인 구성도는 아래 그림을 참조하기 바란다.

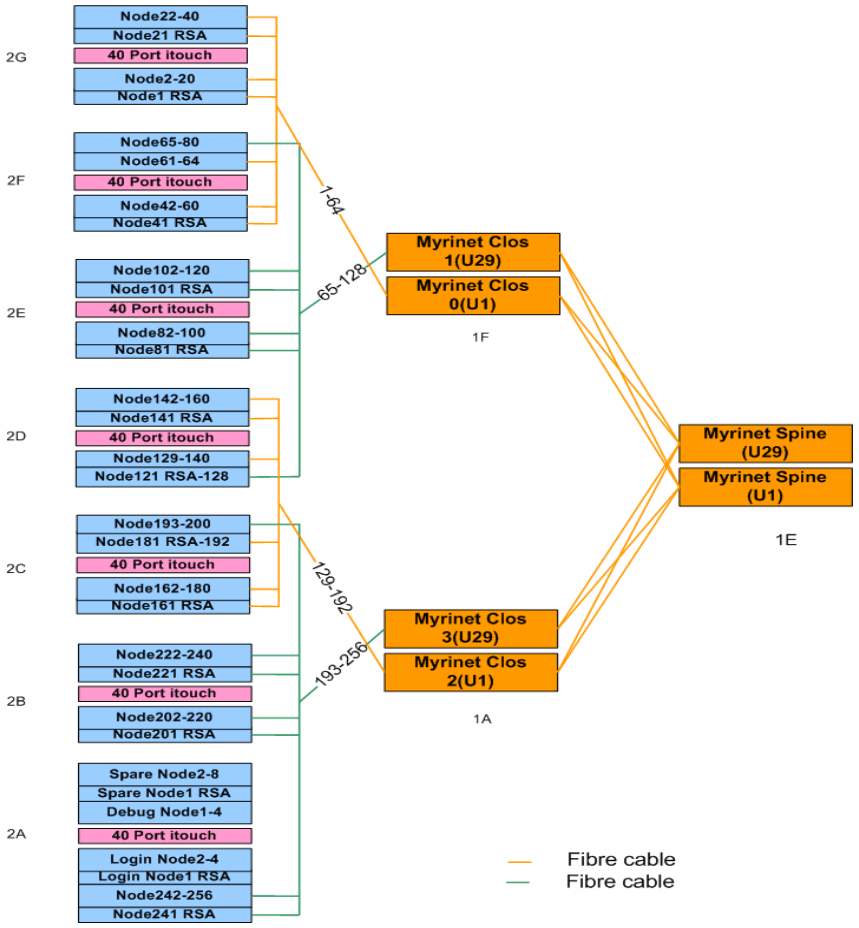

그림 12

다음은 미리넷 구성 하드웨어의 사양이다. 디버깅 노드를 위한 별도의 미리넷 네트웍에 대한 사양은 표시되어 있지 않다.

Myrinet Host Interface: Myrinet Fiber/PCI-XD type Interface(256개)

Myrinet 2000 Switch 라인카드: M3-SW16-8F(32개)

Myrinet 2000 Spine 라인카드: M3-SPINE-8F(64개)

PCI to Myrinet Cable : Fibre Channel 2GB Cable, LC to LC 10m(256개)

Myrinet Spine to Spine Cable: Fibre Channel 2GB Cable, LC to LC 10m(256개)

3-2. 파일서비스 네트웍

Cisco 4006 기가비트 백본 스위치(192 GigE ports, 32 Gbps switching fabric) 두 대로 구성된 파일 I/O 네트웍은 모든 계산 노드 및 서버와 star topology 구조로 연결되어 있다(그림 6 참조). 또한 두 스위치를 8-port trunking으로 연결하여 두 스위치를 경유하는 트래픽의 대역폭을 최대한 확장하고자 했다.
Netpipe를 이용한 성능 측정에서 64bytes message에 대해 약 64 usec의 latency를 보였으며, 1MB의 message에 대해 약 780Mbps의 대역폭이 측정되었다. 위의 그림에서 확인할 수 있는 바와 같이 기가비트 네트웍에 연결된 파일서비스 노드 8대는 두 개의 FC HBA를 통해 2개의 FC fabric 스위치와 스토리지 시스템으로 연결된 이중의 SAN 구성에 참여하고 있으며, 이를 통하여 확장된 대역폭과 가용성을 동시에 얻을 수 있도록 구성했다.
이 네트웍은 이름에서 알 수 있듯이 파일 I/O 및 xCAT을 이용한 리눅스 설치 시 사용된다.

백본스위치: Cisco4006
backplane bandwidth :32 Gbps nonblocking, full duplex
WS-X4448-BG-RJ45(48 GigE ports) 4개,
supervisor engine III 1 개 탑재

그림 13 WS-X4448-BG-RJ45 모듈

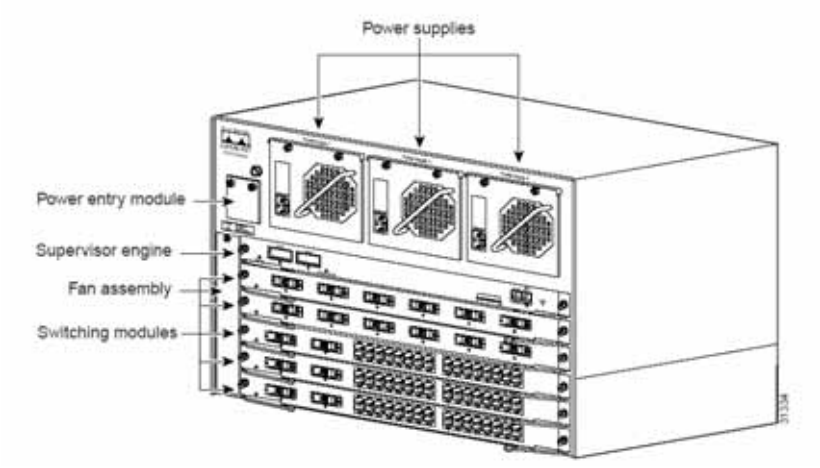

그림 14 Cisco4006 스위치

3-3 관리 네트웍

클러스터 하드웨어의 설치과정(deployment)에서 가장 어려운 부분 중 하나는 여러 종류의 네트워크 케이블을 효율적으로 정리하는 것이다. 이것은 미관상의 이유뿐만 아니라 시스템의 방열에도 심각한 영향을 줄 수 있기 때문에 클러스터의 성능 및 안정성에 영향을 끼치는 중요 요소이다. 본 클러스터에서 사용한 Serial 네트웍의 터미널 서버와 MPN(Management Processor Network)과 KVM 네트웍에서 사용한 C2T 기술은 관리 네트워크의 케이블링 구조를 효율적으로 단순화시켰다. C2T는 IBM에서 개발한 KVM (Keyboard Video Mouse) Daisy Chaining 기술로서, KVM 신호 및 IPMI규격을 따르는 관리 프로세서의 신호를 전송한다. 랙 타입 클러스터 시스템의 네트웍 케이블링의 단순화를 실현한 C2T는 원격 KVM 신호뿐만 아니라 관리 프로세서의 신호 전송을 통하여 원격 시스템 제어, 원격전원관리, 바이오스 이벤트 모니터링 등 시스템의 전체 혹은 일부를 순차적으로 혹은 동시에 관리할 수 있도록 한다.

Serial 네트웍은 일반적인 타입의 직렬 케이블과 터미널 서버를 통해 구성되며, Serial over LAN을 통해 하나의 관리노드에서 클

러스터 전체의 노드를 관리할 수 있도록 해준다. 직렬 네트웍은 빠른 응답시간과 복수 개의 직렬 콘솔을 통해 동시에 여러 노드를 모니터링 및 관리 할 수 있다는 장점을 가지나 바이오스 및 운영체제의 지원을 통해 동작하기 때문에 노드의 운영체제에 심각한 문제가 발생할 경우 모니터링 기능을 상실한다는 단점을 갖고 있다.

3-3-1 KVM 네트웍

C2T daisy chain과 KVM 스위치, 2U 모니터/키보드로 구성된다. 한 랙의 40개의 계산노드는 하나의 C2T daisy chain으로 연결되

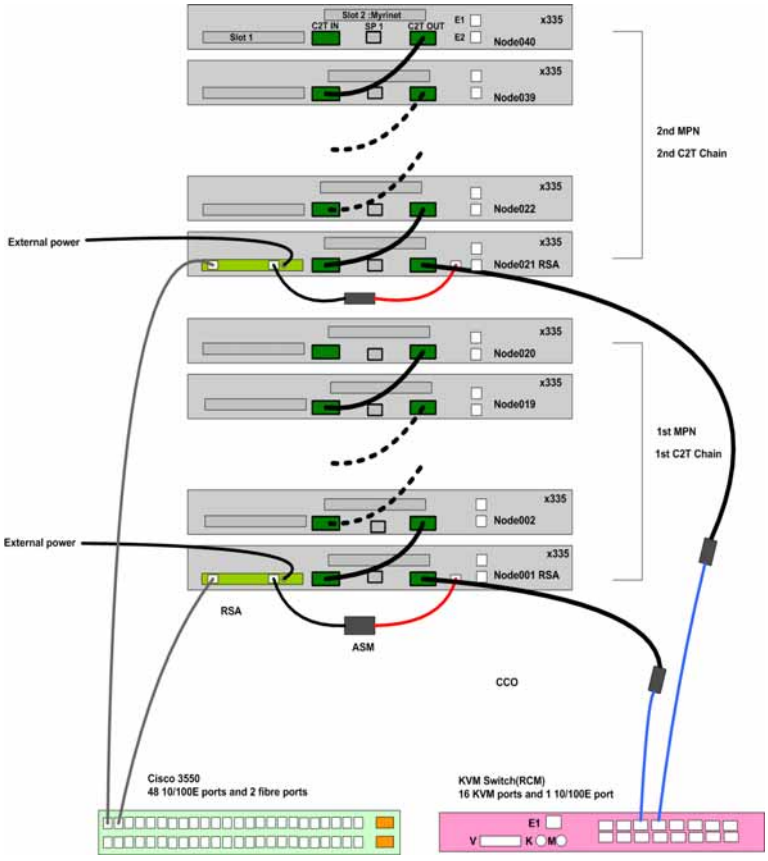

그림 15

며, 이 체인의 제일 하단의 계산노드는 CCO(C2T Conversion Option)을 거쳐 KVM 스위치(a.k.a. RCM(remote console manager), console sever)의 한 포트에 연결된다. 위의 그림은 RSA(Remote Supervisor Adaptor)기반의 MPN도 더불어서 도시되어 있다.

이와 유사하게 8개의 파일서비스노드는 KCO(KVM Conversion Option)를 이용한 daisy chain으로 연결되어 있으며, 관리노드와 스케줄링 노드도 역시 KCO chain으로 연결되어 있다. KCO chain과 C2T chain의 유일한 차이는 C2T chain은 KVM 신호와 더불어 MPN(Manamament Processor Network)의 신호도 전송할 수 있는데 반해 KCO chain은 KVM 신호만을 전송할 수 있다는 데 있다.

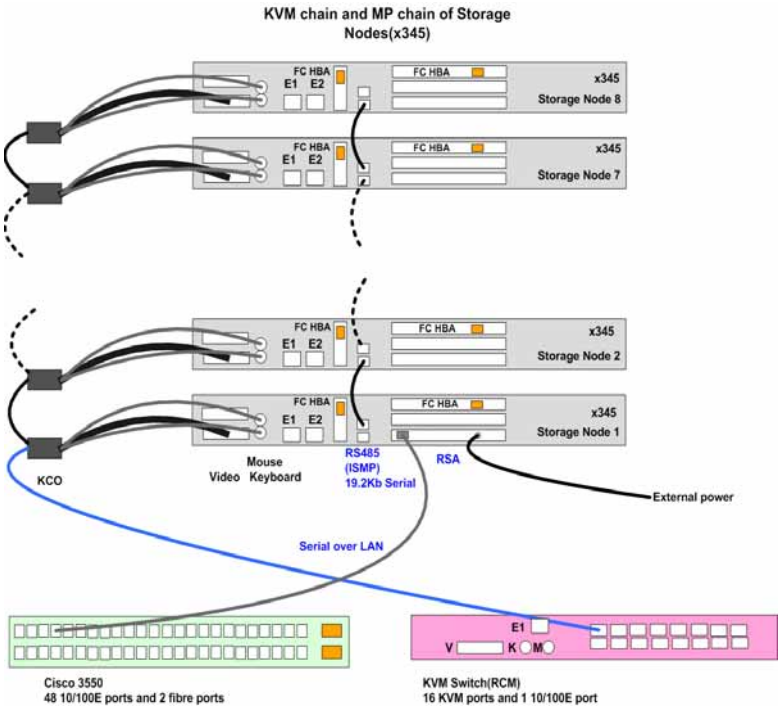

그림 16

KVM 스위치: e1350 NetBAY Remote Console Manager(16-port KVM switch w/ RJ45 connectors)
모니터: e1350 2U Flat Panerl console without keyboard
키보드: e1350 Black TrackPoint Spacesaver Keyboard

3-3-2 serial 네트웍

serial console 네트웍은 40port iTouch terminal server 7대와 20 port itouch terminal sever 2대로 구성된다. 각 랙별로 하나씩

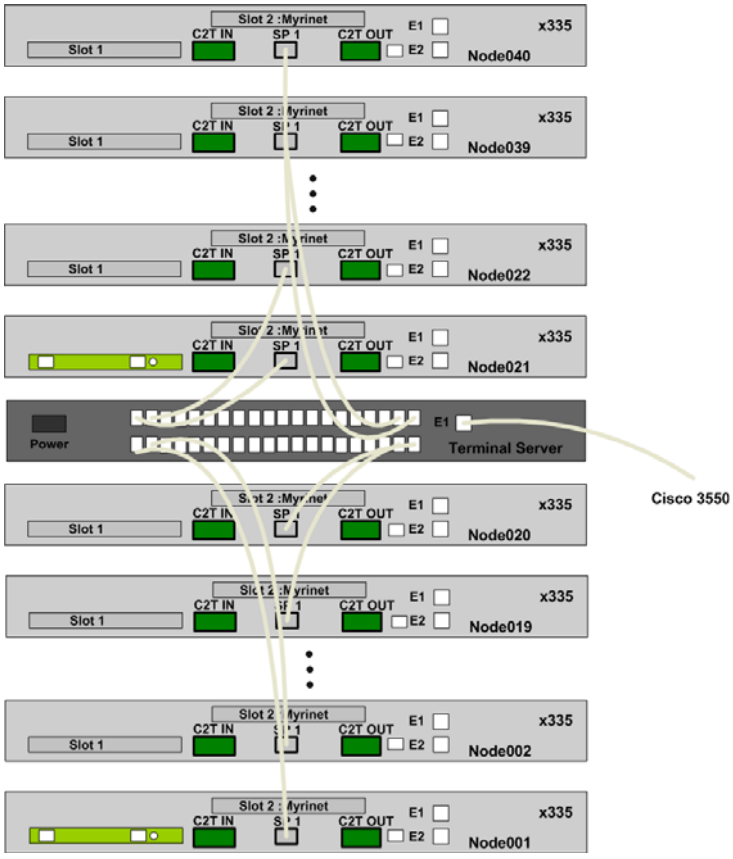

그림 17

위치한 이 terminal server의 각 포트에 해당 랙의 모든 노드들의 serial console이 연결된다(그림 9,10). 또한 terminal server의 10/100Ethernet port는 cisco3550 스위치에 연결되어 있어 네트웍(Serial over Lan)을 통해 serial console이 제공하는 모든 정보를 제공받을 수 있으며 직접 interactive하게 명령을 내릴 수 있다.

구성 장비:
Terminal server: iTouch 8040-101: InReach 40 port w/ RJ45 connectors(7대)
iTouch 8040-101: InReach 20 port w/ RJ45 connectors(2 대)
serial 케이블: e1350 Harness, Serial RJ45 to DB9
ethernet 케이블: e1350 CAT5E Cable Inter Rack -10M

3-3-3 MPN(Management Processor Network)

클러스터의 각 노드의 management processor(a.k.a service processor)와 RSA(remote supervisor adaptor)로 구성되며 RSA가 들어있는 노드는 node001, node021, node041, ..., node221, node241, login001, spare001, mgmt, sched, storage001 이다. 위의 node001의 RSA는 node002부터 node020까지 19 노드의 MPN(management processor network)을 관리하며, x335의 RSA가 관리할 수 있는 최대 노드수는 24개이다. MPN을 관리하는 RSA는 하나의 ethernet port가 있으며, 이것은 cisco3550 스위치에 연결되어 있다. 따라서 네트웍을 통해 RSA에 access한 후 RSA가 관리하는 모든 노드에 접근할 수 있다. event log(fan failures, temperature alarms, hardware failure등)의 정보는 http를 통해 확인할 수 있으며 그리고 xcat을 통해 MPN의 설정후에는 remote power on/off등의 remote control(power, BIOS console 등)명령 을 shell command를 통해 원격에서 할 수 있다.

구성 장비:
MP: H8-based management processor (Hawk)
RSA: e1350 Remote Supervisor Adaptor - 32-bit, 1/2 length (x335 slot 1 only)

3-4 외부 네트워크 장비

넷기어 12 포트 기가비트 스위치를 사용하고 있으며, 로그인 노드 4대, 디버깅 노드 4대, mgmt 및 sched 노드가 연결되어 있으며, 두 개의 포트가 방화벽을 통해 외부로 나가도록 연결되어 있다.

4. 서버 및 노드

4-1. 계산노드(256 대)

계산노드로 IBM x335 서버를 사용하고 있으며 제품의 사양은 다음과 같다.

CPU: Xeon DP 2.8GHz/ 533MHz FSB/ 512K L2(2개)
RAM: 1GB PC2100 CL2.5 ECC DDR SDRAM RDIMM(2개)
RAM: 512MB PC2100 CL2.5 ECC DDR SDRAM RDIMM(2개)
하드드라이브: e1350 36.4GB, 10K RPM, Ultra320 SCSI Hot-swap HDD(1개)
미리넷카드: e1350 Myrinet Fiber/PCI-XD type Interface

2-3-3 에서 보았듯이 13개의 계산노드에는 RSA가 설치되어 있다.
RSA: e1350 Remote Supervisor Adaptor - 32-bit, 1/2 length (x335 slot 1 only)

사용하고 있는 RAM은 Micron사의 MT36VDDT12872G-265B2 (1GB PC2100R DDR 266Mhz CL2.5 ECC REG)과 Hynix사의 HYMD212G726AS4M-HAA(1GB PC 2100R DDR 266Mhz CL2.5 ECC) 그리고 Mosel Vitelic사의 V827464N24SATL-B0(512MB, 2100R DDR 266Mhz CL2.5 ECC) 등이다. HDD로는 Hitachi사의 1C35L036UCDY10-0 모델을 사용하고 있다.

계산노드로 사용하고 있는 x335는 ServerWorks의 메인보드를 사용하고 있으며 내부구조는 다음과 같다. CPU는 코드네임이 Prestonia인 2.8 GHz Xeon dual processor(DP)을 사용하고 있으며, 20KB의 L1 Cache와 512 KB의 L2 Cache on chip이 들어 있다. FSB는 533MHz 클럭으로 작동하며 north bridge 칩셋과 south bridge 칩셋으로 각각 broadcom사의 Grand Champion Low

End chipset(GC-LE)과 CIOB-X2를 사용하고 있다. 메모리는 1 GB PC2100 DDR SDRAM 두개와 512 MB PC2100 DDR SDRAM 두개로 구성되며, 2-way interleaved 방식으로 작동하여 최대 4.2 GB/s(2 * 266 GHz * 8B) 의 bandwidth를 가진다.
계산노드의 후면은 다음과 같다.

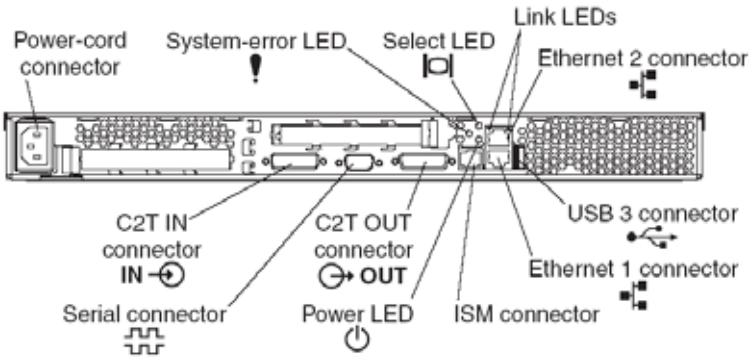

그림 18

시스템의 내부구성은 다음과 같다.

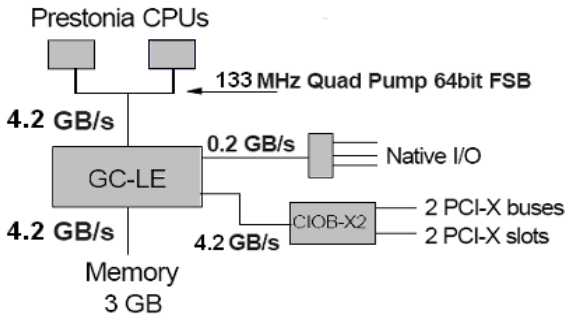

그림 19

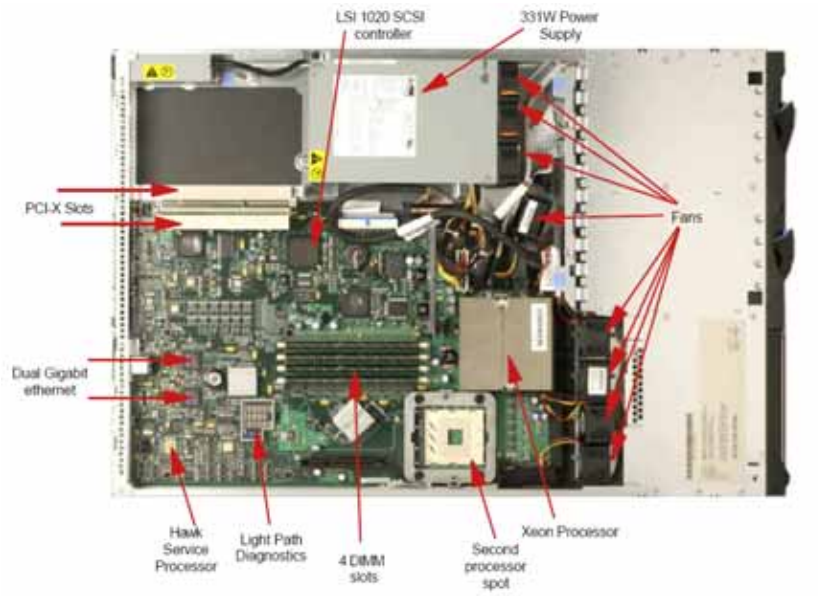

그림 20

4-2. 예비노드 (8대)

예비노드의 사양은 위의 계산노드의 사양과 동일하며 8 개의 예비노드중 1 개의 노드에는 RSA가 설치되어 있다.

4-3. 로그인 노드 (4대)

로그인 노드는 1GB (PC2100 ECC DDR 266 SDRAM) 4개(총 4 GB)의 메모리를 사용하며, 하드 드라이브도 계산노드와 달리 73.4GB(10K RPM< Ultra320 SCSI Hot-swap) HDD 1개를 사용한다.

4-4. 디버깅 노드 (4대)

디버깅노드에는 Myrinet Fiber/PXI-XD type Interface가 설치되어 있는점을 제외하면 로그인 노드와 동일한 사양을 갖는다. 또한 로그인 노드 1개에는 RSA가 설치되어 있으며, 이 RSA는 로그인

노드와 디버깅 노드의 MPN을 관리한다.

4-5. 파일서비스 노드 (8대)

파일서비스노드로 IBM x345서버를 사용하고 있다. x345 서버와 x335서버는 지원하는 디스크 베이의 수(x345: 6, x335: 2)와 확장 슬롯의 수(x345: 5, x335: 2) 그리고 hot-swap 지원 장치의 종류(x345: fan, power supply, hard drive, x335: hard drive) 그리고 form factor (x345: 2U, x335:1U)를 제외하면 동일한 사양을 갖는다. 파일 서비스 노드에는 총 4GB의 메모리가 설치되어 있으며, 두개의 73.4 GB SCSI 하드드라이브가 RAID 1(x335 및 x345 모두 default로 RAID1을 지원함)을 구성하도록 했다. 또 redundant power supply 350W가 추가 되어 있으며, FC Host Bust Adaptor low profile 2개가 설치했다.
8대 파일서비스 노드 중 한 대는 RSA가 들어 있다. 주요 사양은 다음과 같다.

HDD: IBM 73.4 GB 10K RPM SCSI
redundant power supply
FC Host Bus Adaptor: Qlogic FC2-133PCI 2GB/s FC HBA

4-6. 관리노드

파일 서비스 노드와 마찬가지로 x345 서버로 구성되었다. 단 FC2-133PCI 2GB/s FC HBA는 없으며, NetXtreme 1000 SX Fiber Ethernet Adaptor가 설치되어 있다.

4-7. 스케줄링 노드

관리 노드와 동일한 사양으로 이루어져 있다. 단 관리노드나 파일 서비스 노드와 달리 2개의 36.4GB HDD로 RAID 1을 구성하고 있다.

5. SAN(Storage Area Network)

본 클러스터는 파일시스템을 위한 환경으로 광채널 기반의 SAN(Storage Area Network)을 채택하였다. Storage의 FC RAID Controller로 FAStT900 Fibre Channel Storage Server를 사용하고 있으며, Storage Expansion Unit으로 FAStT EXP700 3개를 사용하고 있으며, 각각 14, 14, 7개의 146GB FC HDD가 장착되어 있다. 각 Storage의 디스크의 물리적인 총 크기는 5.11TB이다. FAStT900 FC Storage Sever는 2개의 Drive Mini Hub(2개를 더 추가 할 수 있으며, 이 경우 총 4개의 Drive Loop를 만들 수 있다)와 4개의 Host Mini Hub가 설치되어 있으며, 각각 EXP 700과 IBM 16 port SAN 스위치로 연결된다.

주요 스펙은 다음과 같다.

FC RAID controller	IBM FAStT Strorage Server 900 (2대) 주요 특징: dual redundant 2 Gb RAID controllers (각 controller 당 1 GB Rambus cache 메모리) RAID 0, 1 10, 3, 5 지원 Drive mini-hub 2 개: 각 mini-hub에 2개의 2Gb FC adaptor를 사용할 수 있으며, mini-hub의 Aggregate Bandwidth는 2Gb/s(256 MB/s)이나 실제의 서비스환경에서 sustaned performance 150 MB/s 정도이다. FC-AL로 구성되므로 하나의 mini-hub에 두 개의 FC adaptor를 사용할 필요는 없다. Host mini-hub 4 개: 각 RAID controller에 2 개의 mini-hub가 연결되어 있으며, min-hub당 2개의 2Gb

	FC adaptor를 사용할 수 있다. mini-hub의 Aggregate Bandwidth는 2Gb/s(256 MB/s)이나 실제의 서비스환경에서 sustaned performance 150 MB/s 정도이다. Drive Loop 최대 4개 지원 Host Loop 최대 4개 지원 Global hot spare 지원 최대 16 EXP 700 지원 최대 772 MB/sec disk throughput
FC 디스크박스	IBM FAStT EXP 700 최대 디스크 수: 14 개
FC 디스크	146.8GB 10K RPM, FIBRE, ST 3146807FC (70 개)
FC 스위치	IBM TotalStorage SAN FC 스위치 F16 Brodcade 3800과 동일 모델이며, IBM으로 relabelled 된 제품임.

SAN의 실제 구성은 다음과 같다. SAN 구성에서 redundancy를 보장하기 위해서는 케이블링시 다음 사항에 대해 주의를 해야 한다. 각각의 파일서비스 노드가 두 개의 FC HBA를 가질 때 하나의 FC HBA는 각 Storage의 하나의 RAID controller에만 연결이 되어야 한다. 즉 storage01 노드의 첫 번째 FC HBA는 FAStT900-1의 RAID controller 1에만 물리적으로 연결이 되어야 하며, 두 번째 FC HBA는 FAStT900-1의 RAID controller 2에만 연결이 되어야 한다. 또한 파일서비스 노드에서 두 개의 FAStT900 중 먼저 인식되어야 할 FAStT900을 SAN switch의 낮은 쪽 포트에 연결해야 한다.

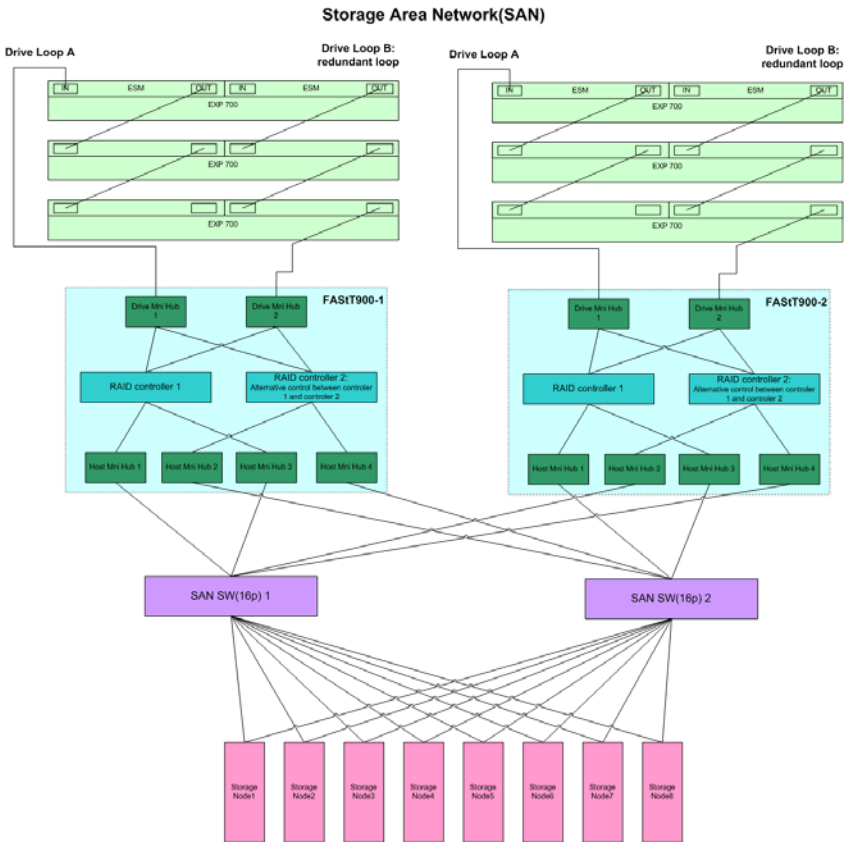

그림 21

EXP 700의 내부 구조는 다음과 같다. Fibre 타입의 SCSI 디스크를 최대 14개까지 장착할 수 있다.

아래는 RAID Controller의 내부모습이다. 하단의 1 GB Rambus DRAM cache는 RAID controller 간의 write cache mirroring을 지원하여 Redundancy를 보장한다. 하지만 이 기능은 cache hit rate가 낮은 경우 불필요한 데이터 caching으로 인해 data access time을 증가시킬 뿐만 아니라 미러링을 위해 drive쪽의 FC loop를 사용하기 때

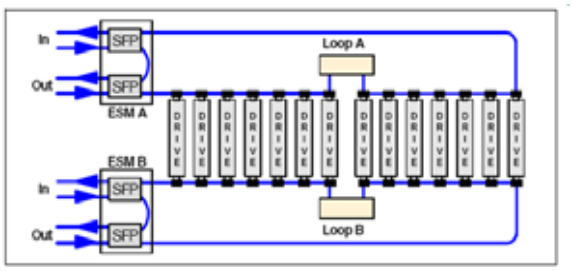

그림 22

문에, 일반 data 전송의 bandwidth을 감소시키는 역할도 하게 된다. 따라서 I/O 패턴에 따라서 다르게 구성될 수 있지만, 고성능 클러스터 환경에서는 read, write cache 기능을 중지시키고, redundancy 대신 dual active RAID controllers 로 설정하여 data throughput을 향상시키도록 사용하는 것이 보다 일반적이다.

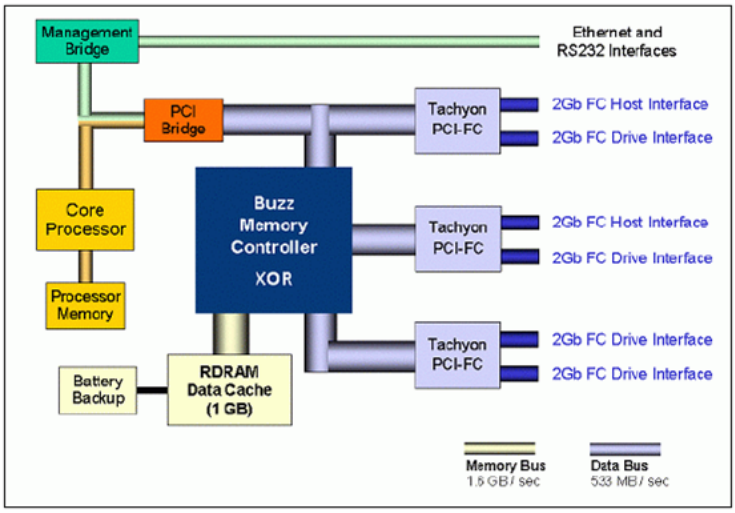

그림 23

6 Master 노드 리눅스 및 xCAT 설치

이후 문서는 xCAT 미니 하우투를 바탕으로 작성하였다. 자세한 내용은 xCAT Redbook 및 영문 xCAT 하우투 문서를 참조하기 바란다.

6-1 redhat 7.3 설치

모든 패키지를 포함하도록 설치를 진행한다. /opt를 위한 별도의 2GB 이상의 파티션을 만들어 둘 것을 권장한다. 모든 redhat package 및 xCAT 툴은 여기에 위치하게 되며, 또한 nfs를 통해 각 계산노드에 마운트될 것이다.

6-2 xCAT의 소스파일 /opt에 설치

```
# cd /opt
# tar zxvpf /tmp/xcat-dist-core-1.2.0.tgz
# tar zxvpf /tmp/xcat-dist-oss-1.2.0.tgz
# tar zxvpf /tmp/xcat-dist-ibm-1.2.0.tgz
# tar zxvpf xcat-dist-ibm-1.2.0-pre4.tgz
```

6-3 xCAT setup

```
# export XCATROOT=/opt/xcat
# cd $XCATROOT/sbin
# ./setupxcat
```

6-4 로그아웃/재로그인

xCAT의 환경변수가 적절히 작동하도록 하기 위해 로그아웃 후 재 로그인하여 추후 작업을 진행할 것을 권장한다.

6-5 NTP 서버 설정

클러스터 환경에서 모든 계산 노드간의 time sync는 필수 조건이

다. 이를 위해 관리 노드(Master 노드)를 NTP 서버로 설정해야 하며, 다음의 작업을 진행한다.

```
# mv -f /etc/ntp.conf /etc/ntp.conf.ORIG
```

다음의 내용을 가지는 /etc/ntp.conf 파일을 만든다.

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
driftfile /etc/ntp/drift
```

다음의 명령어를 적절히 사용하여 time, date, time zone을 설정하고 NTP 서비스를 시작시킨다.

```
# setup --> timezone configuration
# setclock OR hwclock -w
# chkconfig --add ntpd
# service ntpd start
```

특정시간으로 시스템의 시간을 맞추고자 할 경우 아래의 명령어를 사용할 수 있다.

```
# hwclock --set --date="9/22/96 16:45:05"
# hwclock --hctosys
```

다음의 명령어를 사용하여 ntpd 가 정상적으로 작동하는지 확인한다.

```
# ntpdate -q localhost

server 127.0.0.1, stratum 2, offset -0.000002, delay 0.02570
22 Jan 08:04:24 ntpdate[14540]: adjust time server 127.0.0.1
offset -0.000002 sec
```

6-6 xCAT 설정

xCAT은 클러스터 시스템의 설치 및 관리를 위해 별도의 DB를 사용하지 않으며, 아래의 텍
스트 설정 파일등을 통해 시스템을 정의하며, 이것을 시스템의 설치 및 관리운영에 매우 효율적으로 사용한다. 이 설정 파일들은 /opt/xcat/etc/에 위치한다. 이 설정 파일들은 추후 계산노드 설치 및 시스템 운영시 핵심이 되는 파일이다. 따라서 정확히 설정이 되어야 하며 이를 위해 xCAT 1.1.0 Redbook 및 howto 문서를 참조할 것을 권장한다.

site.tab : 이 파일에서 클러스터의 전반적인 구성에 관하여 정의를 한다.

```
# site.tab
rsh                 /usr/bin/ssh
rcp                 /usr/bin/scp
gkhfile             /opt/xcat/etc/gkh
tftpdir             /tftpboot
tftpxcatroot        xcat
domain              kisti.re.kr
dnssearch           NA
nameservers         172.20.100.1
forwarders          NA
dnsdir              /var/named
domainaliasip       172.20.100.1
mxhosts             kisti.re.kr,master.kisti.re.kr
mailhosts           master
master              master
homefs              master:/home
localfs             master:/usr/local
pbshome             /var/spool/pbs
pbsprefix           /usr/local/pbs
pbsserver           NA
scheduler           NA
xcatprefix          /opt/xcat
keyboard            us
```

```
timezone          Asia/Seoul
offutc            9
mapperhost         node200,node201,db01
serialmac          0
serialbps         9600
snmpc                public
snmpd                172.30.100.1
poweralerts          Y
timeservers          master
logdays              7
installdir           /install
clustername          WOPR
dhcpver              2
dhcpconf              /etc/dhcpd.conf
dynamicr     eth0,ia32,172.35.0.1,255.255.0.0,172.35.1.1,172.35.254.254
usernodes             master
usermaster           master
nisdomain            NA
nismaster            NA
nisslaves            NA
homelinks            NA
chagemin              0
chagemax             6000
chagewarn            10
chageinactive         0
mpcliroot            /opt/xcat/lib/mpcli
mpcli2root           /opt/xcat/lib/mpcli2
dpccliroot           /opt/xcat/i686
ipmitool             /opt/xcat/i686/sbin/ipmitool
sshkeyver            2
myrimask             255.255.0.0
ipmimaxp             16
ipmitimeout          30
ipmiretries          2
xcatdport            3001
```

.

nodehm.tab: 각 노드(계산노드 뿐만 아니라 모든 console server,

switch, rsa등을 포함)의 하드웨어 관리 및 사용 방법을 결정한다. 가령 아래의 getmacs는 각 노드의 MAC address를 얻기 위한 방법을 정의하는 것이며, 가능한 옵션으로 rcons, cisco3500, cisco2950, cisco3550, cisco3548, cisco3524등이 있는데 여기에서 각 계산노드는 rcons(console server)를 사용하도록 정의했다.

```
# nodehm.tab
#
# node hardware management
#
# power      = mp,baytech,emp,apc,apcp,ipmi,NA
# reset      = mp,apc,apcp,ipmi,NA
# cad        = mp,NA
# vitals     = mp,ipmi,NA
# inv        = mp,NA
# cons       = conserver,tty,rtel,NA
# mpcons     = rcons,mp,NA
# eventlogs  = mp,ipmi,NA
# getmacs    = rcons,cisco3500,cisco2950,cisco3550,cisco3548,cisco3524
# netboot    = pxe,eb,elilo,nbgrub,file:,NA
# eth0       = eepro100,pcnet32,e100,bcm5700,e1000
# gcons      = vnc,NA
# sb         = N,Y (serial bios)
# beacon     = ipmi,NA
# bootseq
# serialbps
#
#node,power,reset,cad,vitals,
#                    inv,cons,mpcons,eventlogs,getmacs,
#
netboot,eth0,gcons,sb,beacon
#
#node001 mp,mp,NA,mp,
#                    mp,conserver,rcons,mp,rcons,
#                                                    pxe,bcm5700,NA,Y,NA,NA,def

node001 mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,bcm5700,NA,Y,NA,NA,def
node002 mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,bcm5700,NA,Y,NA,NA,def
```

```
...
node255  mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,bcm5700,NA,Y,NA,NA,def
node256  mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,bcm5700,NA,Y,NA,NA,def

db01      mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,bcm5700,NA,Y,NA,NA,def
...
db04      mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,bcm5700,NA,Y,NA,NA,def
lg01      mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,bcm5700,NA,Y,NA,NA,def
...
lg04      mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,bcm5700,NA,Y,NA,NA,def

storage01   mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,e1000,NA,Y,NA,NA,def
...
storage08   mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,e1000,NA,Y,NA,NA,def

sched     mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,e1000,NA,Y,NA,NA,def

spare01  mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,bcm5700,NA,Y,NA,NA,def
...
spare08  mp,mp,NA,mp,mp,conserver,rcons,mp,rcons,pxe,bcm5700,NA,Y,NA,NA,def

rsa01     NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,N,NA,NA,def
...
rsa17     NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,N,NA,NA,def

itch01   NA,NA,NA,NA,NA,NA,NA,NA,NA,file:NEMC_IR.SYS,NA,NA,N,NA,NA,def
...
itch09   NA,NA,NA,NA,NA,NA,NA,NA,NA,file:NEMC_IR.SYS,NA,NA,N,NA,NA,def
```

nodelist.tab:각 노드를 그룹별로 관리할 수 있도록 각 노드의 이름과 이 노드가 속한 그룹을 정의한다. 이 그룹 alias는 시스템의 설치 및 관리 시 각 노드를 해당 그룹별로 나누어서 작업을 할 경우 유용하게 사용된다.

```
# nodelist.tab
#
# This table must list all the nodes in the cluster and defines the groups
```

```
# that each node is part of.
#
# All elements of a cluster are nodes.
#
#<node> <group,group,...>
#

node001 node001,all,ia32,myri,rack2g,compute,ip,image,rsa,rsa04,stage
node002 node002,all,ia32,myri,child,rack2g,compute,ip,image,rsa04,stage01
node003 node003,all,ia32,myri,child,rack2g,compute,ip,image,rsa04,stage01
...
node255 node255,all,ia32,myri,child,rack2a,compute,ip,image,rsa16,stage13
node256 node256,all,ia32,myri,child,rack2a,compute,ip,image,rsa16,stage13

db01 db01,all,ia32,myri,rack2a,debug,ip,image,rsa17
...
lg04 lg04,all,ia32,myri,rack2a,login,ip,image,rsa17

master-eth2   nan

storage01 storage01,storage,all,rack1d,rsa,rsa03
storage02 storage02,storage,all,rack1d,rsa03
storage07 storage07,storage,all,rack1d,rsa03
storage08 storage08,storage,all,rack1d,rsa03

sched sched,schedule,all,rack1c,rsa,rsa01

spare01 spare01,all,ia32,myri,rack2a,spare,ip,image,rsa,rsa18
...
spare08 spare08,all,ia32,myri,rack2a,spare,ip,image,rsa18

rsa01    nan,rsaa
...
rsa18    nan,rsaa

myri01   nan,myris
...
myri06   nan,myris
```

```
itch01 nan,ts
...
itch09 nan,ts

cisco01 nan
cisco02 nan
```

noderes.tab: 이 설정파일에서는 각 계산노드의 그룹이 부팅 및 설치 시 필요한 서비스를 어느 노드로부터 받아야 할지를 정의한다. 아래의 경우 한 가지 예를 들면, stage01에 속한 노드(위의 nodelist.tab 파일을 참조하면 node002부터 node020까지 19개의 노드가 해당됨을 알 수 있다)가 설치 및 부팅 시 사용할 TFTP 서버와 NFS 서버는 node001로 정의되어 있음을 알 수 있다.

```
# noderes.tab
#
# TFTP          = Where is my TFTP server?
#                 Used by makedhcp to setup /etc/dhcpd.conf
#                 Used by mkks to setup update flag location
# NFS_INSTALL   = Where do I get my files?
# INSTALL_DIR   = From what directory?
# SERIAL        = Serial console port (0, 1, or NA)
# USENIS        = Use NIS to authencate (Y or N)
# INSTALL_ROLL  = Am I also an installation server? (Y or N)
# ACCT          = Turn on BSD accounting
# GM            = Load GM module (Y or N)
# PBS           = Enable PBS (Y or N)
# ACCESS        = access.conf support
# GPFS          = Install GPFS
# INSTALL NIC   = eth0, eth1, ... or NA
#
#node    TFTP,NFS_INSTALL,INSTALL_DIR,
#               SERIAL,USENIS,INSTALL_ROLL,ACCT,GM,PBS,ACCESS,GPFS,
#                   INSTALL_NIC
#group TFTP,NFS_INSTALL,INSTALL_DIR,
#               SERIAL,USENIS,INSTALL_ROLL,ACCT,GM,PBS,ACCESS,GPFS,
#                  INSTALL_NIC
```

```
#
stage          master,master,/install,0,N,Y,N,Y,N,Y,N,eth0,eth0
stage01        node001,node001,/install,0,N,N,N,Y,N,Y,N,eth0,eth0
stage02        node021,node021,/install,0,N,N,N,Y,N,Y,N,eth0,eth0
...
stage13        node241,node241,/install,0,N,N,N,Y,N,Y,N,eth0,eth0
compute        master,master,/install,0,N,N,N,Y,N,Y,N,eth0,eth0
debug          master,master,/install,0,N,N,N,Y,N,Y,N,eth0,eth0
login          master,master,/install,0,N,N,N,Y,N,Y,N,eth0,eth0
storage        master,master,/install,0,N,N,N,N,N,Y,N,eth0,eth0
sched          master,master,/install,0,N,N,N,N,N,Y,N,eth0,eth0
spare          master,master,/install,0,N,N,N,Y,N,Y,N,eth0,eth0
```

nodetype.tab: 이 파일은 각 노드가 설치시 사용할 Kickstart 파일의 template의 이름을 명시한다.

```
node001  rh73,x86,stage
node002  rh73,x86,compute
node003  rh73,x86,compute
...
node020  rh73,x86,compute
node021  rh73,x86,stage
....
node256  rh73,x86,compute

db01     rh73,x86,debug
...
db04     rh73,x86,debug

lg01     rh73,x86,login
...
lg04     rh73,x86,login

storage01    rh73,x86,file
...
```

passwd.tab: 시스템(각 노드, 스위치, RSA카드 포함)의 모든 패스워드를 저장하며, 이것은 설치과정 및 rpower를 이용한 시스템

on/off등 에서 사용된다.

```
# passwd.tab
#
# defines the passwords that xCAT needs to access certain devices and the
# root password it needs to set on all nodes

rootpw          password

# telnet and enable secret for Cisco IOS

cisco           ibm1350

# userid and password for the mpa and mp (ASMA, RSA, BC, Ranger)

asmauser        USERID
asmapass        PASSW0RD
# userid and password for the BC switch

bcnetuser   USERID
bcnetpass   PASSW0RD

# userid and password for LAN-based IPMI

ipmiuser        foo
ipmipass        bar

# userid and password for CPS

cpsuserid       Admin
cpspasswd       foobar

# userid and password for flash

flashuser       flash
flashpass       flashpass

# password for summit48i
```

```
#summit48i        password

# password for blackdiamond

#blackdiamond    password
```

postscripts.tab: Post Install Scripts 테이블은 특정 리눅스 OS 혹은 특정 노드 타입에 따른 Post 설치 스크립트들을 결정할 때, 그리고 그것을 실행할 때 사용된다. 자세한 내용은 파일 내에 주석으로 설명된 내용을 참조하기 바란다. 아래의 예에서 다른 버전의 gm을 설치하기 위해 변경한 내용을 확인할 수 있다. 물론 /install/post/kernel에 해당 gm 버전을 위치시켜야 한다.(참조 23. 및 26-2)

```
# Setup syslog
OSVER=rh.* or OSVER=sles8 or OSVER=suse8.* {
        syslog
}

# update/add packages
(OSVER=rh.* or OSVER=sles8 or OSVER=suse8.*) {
        updaterpms nodeps
        otherrpms
        forcerpms
        optcp
}

# update kernel
(OSVER=sles8 or OSVER=suse8.*) and (ARCH=x86 or ARCH=ia64) {
        updatekernel latest
}
OSVER=sles8 and ARCH=x86_64 {
        updatekernel latest
}
OSVER=rh73 {
        updatekernel 2.4.20-28.7 smp
}
```

```
# setup NFS mounts
OSVER=rh.* or OSVER=sles8 or OSVER=suse8.* {
        nfsmounts
}
....
# Setup Myrinet
#TABLE:noderes.tab:$NODERES:$noderes_gm=Y and OSVER=rh73 and ARCH=x86 {
#    myrinet 2.0.9_Linux 2.4.20-28.7smp
#}
TABLE:noderes.tab:$NODERES:$noderes_gm=Y and OSVER=rh73 and ARCH=x86 {
    myrinet 2.0.12_Linux 2.4.20-28.7smp
}
....
```

postdeps.tab: Post scripts의 실행 순서에 관한 dependency를 명시한다.

```
# Post Scripts Dependancies
#
#
# For each script define if any scripts that must be executed first.
#
# e.g. most xCAT post install scripts use logger to send messages to
# the management node, so syslog should be setup first to allow this.
#

bootmenu          syslog,syncdirs
mountpost         syslog
remoteshell       syslog,mountpost
updateflag        syslog,remoteshell
umountpost        syslog
hardeth0          syslog
hardeths          syslog
harddns           syslog
hardname          syslog,harddns
hardgate          syslog
hardnis           syslog
hardhosts         syslog,harddns
```

```
updaterpms        syslog,mountpost
otherrpms         syslog,mountpost
tarballs          syslog,mountpost
updatekernel      syslog,mountpost
##### custom script for copy customized opt dir ######
optcp             syslog,mountpost
#####################################################
nfsmounts         syslog
syncclock         syslog
setupntp          syslog
setuppam          syslog
setuppaths        syslog
setupservices     syslog
syncdirs          syslog,mountpost
```

snmptrapd.conf: 이것은 RSA 체인에 연결된 x335, 혹은 x345 장비의 s/n, model number를 얻을 때 사용된다.

```
traphandle default /opt/xcat/sbin/snmptraphandle
```

networks.tab: 클러스터에 사용될 네트워크의 IP 주소를 정의한다.

```
# networks.tab
#
#network          mask,gateway,dns,dns,dns,...
#
172.20.0.0        255.255.0.0,NA,172.20.100.1
172.30.0.0        255.255.0.0,NA,172.30.100.1
```

mac.tab: 이 테이블은 각 노드의 이더넷 MAC 어드레스를 저장한다. getmacs 스크립트 실행시 자동으로 생성된다.

```
db01-eth0         00:0D:60:1B:3C:DA
db01-eth1         00:0D:60:1B:3C:DB
...
```

```
db04-eth0          00:0D:60:1B:3C:F4
db04-eth1          00:0D:60:1B:3C:F5
lg01-eth0          00:0D:60:1B:3A:6E
...
lg04-eth0          00:09:6B:63:86:C5
lg04-eth1          00:09:6B:63:86:94
myri01             00:60:dd:49:92:02
...
myri06             00:60:dd:49:92:0a
node001-eth0       00:0D:60:1B:3C:AA
node001-eth1       00:0D:60:1B:3C:AB
node002-eth0       00:09:6B:63:A0:4A
node002-eth1       00:09:6B:63:A0:49
...
```

본 클러스터는 터미널 서버를 통해 각 노드의 serial 콘솔을 관리한다. 이를 위해 conserver.tab과 conserver.cf 이 두개의 파일을 정의해야 한다.

conserver.tab: 이 테이블에서 터미널서버의 conserver 대몬의 실행 노드(아래의 경우, localhost 즉 mgmt 노드)와 consol 명령어를 통해 접근할 각 노드(x335, x345의 hostname)의 이름을 정의한다.

```
# conserver.tab
#
# This table defines the relationship between nodes and console servers.
#
#   <conserver>   hostname of the console server
#   <console>     name of the console that corresponds to conserver.cf
#
#<node> <conserver>,<console>
```

```
node001 localhost,node001
node002 localhost,node002
node003 localhost,node003
node004 localhost,node004
node005 localhost,node005
...
```

conserver.cf : 터미널 서버의 각 포트와 이에 연결된 노드를 정의한다. 또한 시리얼 콘솔의 output을 로그파일로 저장하기 위한 경로를 지정한다. 터미널 서버의 포트 번호는 2100부터 시작하며, 100씩 증가한다. (port 1은 2100이며, port 2는 2200이다, port 40은 6000이다) 아래의 경우 node001은 터미널서버의 21번 포트에 연결되어 있다.

```
#
LOGDIR=/var/log/consoles
# The character '&' in logfile names are substituted with the console
# name.  Any logfile name that doesn't begin with a '/' has LOGDIR
# prepended to it.
node001:!itch01:4100:&:
node002:!itch01:4200:&:
node003:!itch01:4300:&:
...
node009:!itch01:4900:&:

node010:!itch01:5000:&:
node011:!itch01:5100:&:
node012:!itch01:5200:&:
...
node019:!itch01:5900:&:

node020:!itch01:6000:&:
node021:!itch01:2100:&:
node022:!itch01:2200:&:
...
node029:!itch01:2900:&:

node030:!itch01:3000:&:
```

```
node031:!itch01:3100:&:
...
node040:!itch01:4000:&:



node041:!itch02:4100:&:
node042:!itch02:4200:&:
node043:!itch02:4300:&:
node044:!itch02:4400:&:
node045:!itch02:4500:&:
...
```

각 x335, x345 노드에 있는Management Processor를 위해 다음의 두 가지 파일을 정의해야 한다.

mpa.tab: 이 테이블은 클러스터내의 Management Processor의 리스트를 명시한다.

```
# mpa.tab
#
# This table lists the Management Processor Adapters in this cluster
#
# <type>        asma,rsa
# <name>        internal name (*)
# <number>      internal number,unique and >10000
# <command>     telnet,mpcli,http(hawk only)
# <reset>       mpcli(hawk only),mpcli,NA
# <rvid>        telnet(asma only),NA
# <dhcp>        Y/N (rsa only)
# <gateway>     default gateway IP address or NA or using DHCP
# <dns>         DNS server or NA if using DHCP
#
# NOTE: command and reset method should not be the same, if the
#         command interface is locked of unavailable, a different
#         interface must be used.  Please read the
#         managementprocessor-HOWTO.html
#
# (*)internal name should be the node name if the mpa is the
# primary management adapter for that node
```

```
#
#<node>,<type>,<name>,<number>,
#          <command>,<reset>,<rvid>,<dhcp>,
#                                        <gateway>,<dns>
#
rsa01    rsa,rsa01,10001,http,mpcli,NA,N,NA,NA
rsa02    rsa,rsa02,10002,http,mpcli,NA,N,NA,NA
rsa03    rsa,rsa03,10003,http,mpcli,NA,N,NA,NA
..
rsa17    rsa,rsa17,10017,http,mpcli,NA,N,NA,NA
rsa18    rsa,rsa18,10018,http,mpcli,NA,N,NA,NA
```

mp.tab: 이 테이블은 Management Processor 네트웍을 정의한다. 즉 MPA(Management Processor Adaptor, a.k.a, RSA)을 통해 Management Processor 네트웍 안에 있는 개별 Management Processor에 접근할 수 있다.

```
# mp.tab
# This table describes the topology of the Management Processor Network.
# This enables xcat to contact the management processors via the mpa
# connected to the same chain.
#
# <mpa>     name of the mpa connected to this node
# <name>    internal name of the mp adapter (*)
#
# (*)this field should be NA if the mpa is the primary management
# adapter for that node
#
#<node> <mpa>,#<node>

node001 rsa04,NA
node002 rsa04,node002
node003 rsa04,node003
..
node020 rsa04,node020
node021 rsa05,NA
node022 rsa05,node022
node023 rsa05,node023
```

```
..
node256 rsa16,node256

db01 rsa17,NA
db02 rsa17,db02
...
lg04 rsa17,lg04

storage01        rsa03,NA
storage02        rsa03,storage02
...
storage08        rsa03,storage08

sched    rsa01,NA

spare01 rsa18,NA
spare02 rsa18,spare02
...
spare08 rsa18,spare08
```

7 계산노드의 펌웨어/바이오스 업데이트 및 설정

7-1 계산노드 시스템의 firmware 업데이트

웹에서 다운로드한 최신 firmware/bios를 windows의 Director console을 이용하여 업데이트 작업 진행한다. xCAT에서도 firmware 업데이트를 지원하나 제대로 동작하지 않는 경우가 많다.

7-2 BIOS 설정 업데이트

1. boot order가 network boot(PXE)가 HDD 보다 먼저 오도록 한다.
2. POST/BIOS 정보가 serial console를 통해 redirect되도록 한다.
3. 그 외 설정은 아래의 example 활용을 한다.

```
## BIOS configuration option ## from xCAT redbook p.98

### Under Devices and I/O Ports:
Serial Port A Port 3F8, IRQ 4
Serial Port B Disabled
```

```
- Under Remote Console Redirection:
    Active        Enabled
    Port          COM1
    Baud Rate  9600
    Data Bits    8
    Parity        None
    Stop Bits    1
    Emulation ANSI
    Active After Boot Enabled

### Under Start Options:
Boot Fail Count Disabled
Virus Detection Disabled
- Under Startup Sequence:
    First Startup Device        Diskette Drive 0
    Second Startup Device    CD ROM
    Third Startup Device       Network
    Fourth Startup Device     Hard Disk 0
```

굵은 선으로 표시된 값들은 BIOS의 초기값과 다르게 되어 있는 것들이다.

8 /etc/hosts 파일 설정

한 개 이상의 인터페이스가 있을 경우 node001-eth0, node002-eth2와 같이 '-' 기호를 사용한다. 이 파일은 dns의 DB를 만들 때 /opt/xcat/etc/site.tab과 더불어 참조된다.

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1                   localhost.localdomain localhost

172.20.100.1                master   master.kisti.re.kr
172.30.100.1                master-eth2 master-eth2.kisti.re.kr

172.20.100.2                sched    sched.kisti.re.kr

#### Cluster VLAN ####
172.20.0.1                  node001                 node001.kisti.re.kr
172.20.0.2                  node002                 node002.kisti.re.kr
```

```
172.20.0.3          node003          node003.kisti.re.kr
...
172.20.2.56         node256          node256.kisti.re.kr

172.20.10.1         db01             db01.kisti.re.kr
...
172.20.10.4         db04             db04.kisti.re.kr

172.20.20.1         lg01             lg01.kisti.re.kr
...
172.20.20.4         lg04             lg04.kisti.re.kr

172.20.30.1         spare01          spare01.kisti.re.kr
...
172.20.30.8         spare08          spare08.kisti.re.kr

#### Management VLAN ####
172.30.30.1         rsa01            rsa01.kisti.re.kr
...
172.30.30.18        rsa18            rsa18.kisti.re.kr

172.30.20.1         itch01           itch01.kisti.re.kr
...
172.30.20.9         itch09           itch09.kisti.re.kr

172.30.70.1         rcm01            rcm01.kisti.re.kr

172.30.90.1         raid01           raid01.kisti.re.kr
172.30.90.2         raid02           raid02.kisti.re.kr
172.30.91.1         raid03           raid03.kisti.re.kr
172.30.91.2         raid04           raid04.kisti.re.kr

172.30.80.1         cisco01          cisco01.kisti.re.kr
172.30.80.2         cisco02          cisco02.kisti.re.kr
172.30.50.1         cisco3550        cisco3550.kisti.re.kr
172.30.40.1         cisco3508        cisco3508.kisti.re.kr

172.30.85.1         foundry01        foundryw01.kisti.re.kr
```

```
172.30.60.1          sansw01          sansw01.kisti.re.kr
172.30.60.2          sansw02          sansw02.kisti.re.kr

172.30.61.1          mds01            mds01.kisti.re.kr

172.30.180.1         myri01           myri01.kisti.re.kr
...
172.30.180.6         myri06           myri06.kisti.re.kr

#### IP over Myrinet ####
172.10.0.1       node001-myri0       node001-myri0.kisti.re.kr
172.10.0.2       node002-myri0       node002-myri0.kisti.re.kr
...
172.10.2.56      node256-myri0       node256-myri0.kisti.re.kr

172.10.10.1          db01-myri0       db01-myri0.kisti.re.kr
...
172.10.10.4          db04-myri0       db04-myri0.kisti.re.kr

172.10.20.1          lg01-myri0       lg01-myri0.kisti.re.kr
...
172.10.20.4          lg04-myri0       lg04-myri0.kisti.re.kr

172.10.30.1      spare01-myri0       spare01-myri0.kisti.re.kr
...
172.10.30.8      spare08-myri0       spare08-myri0.kisti.re.kr

## Storage VLAN
172.20.50.1          storage01        storage01.kisti.re.kr
...
172.20.50.8          storage08        storage08.kisti.re.kr
```

9 DNS 서버를 설정

아래의 스크립트는 자동으로 xCAT의 site.tab 그리고 /etc/hosts 파일을 이용하여 DNS DB를 업데이트하고 대몬을 재시작시킨다.

```
# makedns
```

10 Non-collectable MACs를 /opt/xcat/etc/mac.tab에 추가

이러한 경우는 terminal server, switches, RSA등이 해당된다.

11 DHCP 서버 설정

redhat을 사용할 경우 /opt/xcat/etc/site.tab에 dhcpver이 2로 표기되어 있어야 한다. 또한 /opt/xcat/etc/networks.tab에 DNS가 지원하는 네트워크를 정확히 명시해야 한다.

```
# makedhcp --new --allmac
```

12 클러스터의 Ethernet switch 설정

클러스터내의 스위치를 설정시 VLAN 구성, 그리고 trunk 구성등이 필요할 것이다. 이중에서 일단 VLAN 구성에 대해서만 간단히 설명한다. 보다 자세한 내용은 Catalyst 3550 Multilayer Switch Software Configuration Guide 혹은 Creating Ethernet VLANs on Catalyst Switches 를 참조한다.

12-1 telnet을 이용한 스위치 접속

switch의 hostname은 /etc/hosts파일을 참조한다.

```
[root@master root]# telnet cisco3550
Trying 172.30.50.1...
Connected to cisco3550 (172.30.50.1).
Escape character is '^]'.


User Access Verification
Password:
```

/opt/xcat/etc/passwd.tab을 참조하여 적절히 passwd를 입력한다.

12-2 Privileged EXEC 모드로 전환

이 모드로 전환한 후, 기존의 vlan 구성을 체크한다.

```
cisco3500>enable
Password:
cisco3500#show vlan

VLAN  Name     Status     Ports
----- ---      -----      -----------------------
1     default    active     Fa0/1, Fa0/2, Fa0/3, Fa0/4, Fa0/5, Fa0/6, Fa0/7, Fa0/8,
                            Fa0/9, Fa0/10, Fa0/11, Fa0/12, Fa0/13, Fa0/14
                            Fa0/15, Fa0/16, Fa0/17, Fa0/18, Fa0/19, Fa0/20, Fa0/21,
                            Fa0/22, Fa0/23, Fa0/24, Fa0/25, Fa0/26, Fa0/27
                            Fa0/28, Fa0/29, Fa0/30, Fa0/31, Fa0/32, Fa0/33, Fa0/34,
                            Fa0/35, Fa0/36, Fa0/37, Fa0/38, Fa0/39, Fa0/40
                            Fa0/41, Fa0/42, Fa0/43, Fa0/44, Fa0/45, Fa0/46, Fa0/47,
                            Fa0/48, Gi0/1
20    VLAN0020   active     Gi0/2
1002  fddi-default          active
1003  token-ring-default    active
1004  fddinet-default       active
1005  trnet-default         active

VLAN Type  SAID   MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- --- ---- ----- ---- ---- ------ ---- -------- ------ ------
1    enet 100001 1500  -    -    -      -    -        0      0

20   enet 100020 1500  -    -    -      -    -        0      0

1002 fddi 101002 1500  -    -    -      -    -        0      0

1003 tr   101003 1500  -    -    -      -    -        0      0

1004 fdnet 101004 1500 -    -    -      ieee -        0      0

1005 trnet 101005 1500 -    -    -      ibm  -        0      0
```

12-3 기존의 startup-config 내용을 tftp 서버에 백업하기.

tftp서버 구성은 현재 Master 노드에 되어 있다. 기본적으로 /tftpboot 디렉토리로 복사되며, 이 디렉토리에 대해 group, others 에게도 쓰기 권한이 있어야 복사가 가능하다.

```
cisco3500#copy startup-config tftp
Address or name of remote host []? 172.30.100.1
Destination filename [cisco3500-confg]?
!!!
5741 bytes copied in 0.040 secs (143525 bytes/sec)
cisco3500#
```

12-4 Global configuration 모드로 전환

```
cisco3500#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
```

12-5 Config-vlan 모드로 전환

Config-vlan모드로 전환하면서 동시에 새로운 VLAN을 추가하거나 혹은 기존의 VLAN 수정할 수 있다 "vlan vlan_id" 형태로 명령을 내리면 된다

```
cisco3500(config)#vlan 30
cisco3500(config-vlan)#exit
cisco3500(config)#exit

##  ID가 30인 새로운 vlan이 생성된 것을 확인할 수 있다.
cisco3500#show vlan

VLAN Name Status    Ports
---- ----- ------ -------------------------------
```

```
1   default      active     Fa0/1, Fa0/2, Fa0/3, Fa0/4, Fa0/5, Fa0/6, Fa0/7, Fa0/8,
                            Fa0/9, Fa0/10, Fa0/11, Fa0/12
                            Fa0/13, Fa0/14, Fa0/15, Fa0/16, Fa0/17, Fa0/18, Fa0/19,
                            Fa0/20, Fa0/21, Fa0/22, Fa0/23
                            Fa0/24, Fa0/25, Fa0/26, Fa0/27, Fa0/28, Fa0/29, Fa0/30,
                            Fa0/31, Fa0/32, Fa0/33, Fa0/34
                            Fa0/35, Fa0/36, Fa0/37, Fa0/38, Fa0/39, Fa0/40, Fa0/41,
                            Fa0/42, Fa0/43, Fa0/44, Fa0/45
                            Fa0/46, Fa0/47, Fa0/48, Gi0/1
20   VLAN0020   ctive     Gi0/2
30   VLAN0030                    active
1002 fddi-default                 active
1003 token-ring-default           active
1004 fddinet-default              active
1005 trnet-default                active

VLAN Type  SAID   MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1
Trans2
---- ----- ----- ----- ------ ------ -------- ---- --------
------ --
1    enet  100001    1500  -      -      -        -    -        0      0
20   enet  100020    1500  -      -      -        -    -        0      0
30   enet  100030    1500  -      -      -        -    -        0      0
1002 fddi  101002    1500  -      -      -        -    -        0      0
1003 tr    101003    1500  -      -      -        -    -        0      0
1004 fdnet 101004    1500  -      -      -        ieee -        0      0
1005 trnet 101005    1500  -      -      -        ibm  -        0      0

Remote SPAN VLANs
------------------------------------------------------------
```

12-6 VLAN 추가

이제 스위치의 특정 포트(37 - 48포트: 12개)를 방금 전에 만든 VLAN에 추가해 보자

```
## 포트를 먼저 선택한다
```

```
cisco3500#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
cisco3500(config)#interface range Fa0/37 - 48

## 이제 "spanning-tree portfast" 모드로 동작하게 하고 channel group의
## 사용을 중지시키려면 다음의 명령을 내린다.
cisco3500(config-if-range)#switchport host
```

여기서 spanning-tree protocol은 일반적으로 네트웍의 경로가 loop를 이룰 때(즉 한 노드로 가는 경로가 여러 개 존재할 때) 이 경로들 중에서 최상의 경로를 찾아내고, 다른 경로는 block시키는데 사용된다. 하지만 이 프로토콜의 정보들은 새로운 노드가 링크될 때 마다 update되어야 하며, 약 1분 정도의 delay를 야기한다. channel group은 스위치 간에 trunk를 사용 하고자 할 때 사용 한다.

간단한 예로 spanning-tree protocol의 한계를 설명할 수 있다. 만일 위에서 언급한 네트워크의 루프가 세 개(이상의) 스위치에서 만들어질 경우, 또한 세 개(이상)의 스위치간의 connection이 모두 동시에 사용되어할 경우, spanning tree protocol은 해결책이 될 수 없다. 따라서 이 경우, 유일한 해결책은 vlan을 다르게 하는 것이다.

12-7 (참고사항) 다른 방법

앞의 "switchport host" 명령에서 “host"는 다음의 명령들을 실행하는 스크립트를 수행한다. 따라서 아래와 같이 직접 수행해도 된다. 이와 같은 스크립트 명령어는 모든 cisco 스위치에 공통적으로 적용되지 않는다. 따라서 매뉴얼을 참조하면서 작업을 해야 한다.

```
## 위의 포트에 대한 설정을 하기위해 스위치를 read-write를 허용
```

```
## 하는 access mode에 놓는다
cisco3500(config-if-range)#switchport mode access

## vlan30이 위의 포트로 들어오는 모든 traffic을 허용하도록 한다
cisco3500(config-if-range)#switchport access vlan 30

## 위의 포트가 forwarding state로 재빨리 바뀌도록 한다.
cisco3500(config-if-range)#spanning-tree portfast
cisco3500(config-if-range)#end
```

12-8 세팅 확인

```
cisco3500#show running-config interface Fa0/37
Building configuration...

Current configuration : 139 bytes
!
interface FastEthernet0/37
 switchport access vlan 30
 switchport mode access
 no ip address
 no cdp enable
 spanning-tree portfast
end


## VLAN 세팅을 확인한다
cisco3500#show vlan

VLAN  Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4, Fa0/5, Fa0/6, Fa0/7, Fa0/8,
```

```
                                Fa0/9, Fa0/10, Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                Fa0/15, Fa0/16, Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                Fa0/21, Fa0/22, Fa0/23, Fa0/24, Fa0/25, Fa0/26, Fa0/27
                                Fa0/28, Fa0/29, Fa0/30, Fa0/31, Fa0/32, Fa0/33, Fa0/34
                                Fa0/35, Fa0/36, Gi0/1
20   VLAN0020        active     Gi0/2
30   VLAN0030        active     Fa0/37, Fa0/38, Fa0/39, Fa0/40, Fa0/41, Fa0/42, Fa0/43
,                               Fa0/44, Fa0/45, Fa0/46, Fa0/47, Fa0/48
1002 fddi-default                active
1003 token-ring-default          active
1004 fddinet-default             active
1005 trnet-default               active

VLAN  Type   SAID        MTU    Parent  RingNo  BridgeNo  Stp   BrdgMode  Trans1
Trans2
----  -----  ----------  -----  ------  ------  ----------  --  ------
1     enet   100001      1500   -       -       -           -     -         0       0

20    enet   100020      1500   -       -       -           -     -         0       0

30    enet   100030      1500   -       -       -           -     -         0       0

1002  fddi   101002      1500   -       -       -           -     -         0       0

1003  tr     101003      1500   -       -       -           -     -         0       0

1004  fdnet  101004      1500   -       -       -           ieee  -         0       0

1005  trnet  101005      1500   -       -       -           ibm   -         0       0
```

12-9 설정 저장하기

현재의 설정을 스위치가 reboot되어도 항상 적용하도록 하기 위해서는 다음 명령을 내리면 된다.

```
cisco3500#copy running-config startup-config
```

12-10 원래대로 되돌리기

새로 구성한 vlan 및 port를 원래대로 되돌리기 위해 다음의 과정을 진행한다.

```
## 이제 만들어진 VLAN ID가 30 인 vlan을 제거해 보자
cisco3500#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
cisco3500(config)#no vlan 30
cisco3500(config)#end

## 확인
cisco3500#show vlan
VLAN Name      Status         Ports
---------------- --------- -------------------------------
1      default       active         Fa0/1, Fa0/2, Fa0/3, Fa0/4, Fa0/5, Fa0/6, Fa0/7, Fa0/8,
                                     Fa0/9, Fa0/10, Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                     Fa0/15, Fa0/16, Fa0/17, Fa0/18, Fa0/19, Fa0/20, Fa0/21,
                                     Fa0/22, Fa0/23, Fa0/24, Fa0/25, Fa0/26, Fa0/27
                                    Fa0/28, Fa0/29, Fa0/30, Fa0/31, Fa0/32, Fa0/33, Fa0/34,
                                    Fa0/35, Fa0/36, Gi0/1
20   VLAN0020                  active        Gi0/2
1002 fddi-default              active
1003 token-ring-default        active
1004 fddinet-default           active
1005 trnet-default             active

VLAN Type   SAID          MTU     Parent RingNo BridgeNo Stp   BrdgMode
Trans1 Trans2
---- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -       -       -        -    -        0      0

20   enet  100020     1500  -       -       -        -    -        0      0
```

```
1002 fddi   101002      1500   -       -       -         -     -         0       0

1003 tr     101003      1500   -       -       -         -     -         0       0

1004 fdnet 101004       1500   -       -       -         ieee  -         0       0

1005 trnet 101005       1500   -       -       -         ibm   -         0       0

## 이제 스위치의 각 포트의 세팅을 원래대로 되돌리자
cisco3500#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
cisco3500(config)#vlan 1
cisco3500(config)#interface range  Fa0/37 - 48
cisco3500(config-if-range)#switchport mode access
cisco3500(config-if-range)#switchport access vlan 1
cisco3500(config-if-range)#end

## 확인
cisco3500#show vlan

VLAN Name           Status     Ports
---- -------------------- --------- -------------------------------
1       default        active       Fa0/1, Fa0/2, Fa0/3, Fa0/4, Fa0/5, Fa0/6, Fa0/7, Fa0/8,
                                    Fa0/9, Fa0/10, Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                    Fa0/15, Fa0/16, Fa0/17, Fa0/18, Fa0/19, Fa0/20, Fa0/21,
                                    Fa0/22, Fa0/23, Fa0/24, Fa0/25, Fa0/26, Fa0/27
                                    Fa0/28, Fa0/29, Fa0/30, Fa0/31, Fa0/32, Fa0/33, Fa0/34,
                                    Fa0/35, Fa0/36, Fa0/37, Fa0/38, Fa0/39, Fa0/40
                                    Fa0/41, Fa0/42, Fa0/43, Fa0/44, Fa0/45, Fa0/46, Fa0/47,
                                    Fa0/48, Gi0/1
20    VLAN0020        active       Gi0/2
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
```

```
VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- --------- ------
1    enet  100001     1500  -      -      -        -    -         0      0

20   enet  100020     1500  -      -      -        -    -         0      0

1002 fddi  101002     1500  -      -      -        -    -         0      0

1003 tr    101003     1500  -      -      -        -    -         0      0

1004 fdnet 101004     1500  -      -      -        ieee -         0      0

1005 trnet 101005     1500  -      -      -        ibm  -         0      0
```

13 터미널 서버 셋업

각 터미널 서버의 포트에 각 노드의 시리얼 케이블이 순서대로 연결되어 있어야 하며, 앞서 설명한 conserver.tab과 conserver.conf 파일이 이에 맞게 적절히 설정이 되어 있어야 한다.

13-1 케이블 연결 및 /etc/hosts 파일 확인

각 터미널 서버의 랜포트가 이더넷 네트워크로 연결되어 있어야 하며, /etc/hosts에 적절한 hostname을 지정해야 한다.

```
172.30.20.1        itch01        itch01.kisti.re.kr
172.30.20.2        itch02        itch02.kisti.re.kr
....
172.30.20.9        itch09        itch09.kisti.re.kr
```

13-2 nodelist.tab 설정

앞서 언급된 nodelist.tab 파일에 해당 터미널 서버의 정보를 적절히 입력해야 한다.

13-3 DNS DB 업데이트(makedns)

위와 같은 과정을 거친 후 makedns 명령을 통해 dns DB를 업데이트한다.

```
[root@master root]# makedns
```

13-4 터미널 서버 셋업

터미널 서버를 셋업하기 위해 다음의 과정을 수행한다.
터미널 서버를 통해 각 계산노드의 OS에서 지원하는 remote serial console와 BIOS에서 지원하는 remote BIOS/POST console 등을 이용할 수 있다.

13-4-1 터미널 서버의 정보를 초기화 하기 위해 reset을 수행한다.

13-4-2 각 터미널 서버에 대해 IP 주소를 설정한다.

다음은 iTouch InReach IR series 터미널 서버를 기준으로 설명한다. 시리얼 케이블을 터미 널 서버의 가장 높은 번호를 가진 포트에 연결한다. 이 포트가 local management port를 겸한다. 반대쪽을 serial terminal, terminal emulator, 또는 server에 연결하고, 9600 baud로 세팅한후 IP 주소를 설정한다.

```
login>access
Enter username>root
iTouch>set priv
password>system
iTouch>define server ip address 10.1.1.1
iTouch>devine server ip subnet mask 255.255.0.0
iTouch>init server
```

13-4-5 다음의 명령을 통해 터미널서버의 부가적인 설정을 한다.

이 명령어는 각 터미널 서버 타입에 맞는 스크립트를 수행하며, expect를 사용하여 작성되었다.

```
[root@master root]# setupir [terminal server name]
```

터미널 서버에 관한 세부적인 설정 방법은 IBM의 내부 문서인 hardware maintenance 문서를 참조할 수 있다.

14 conserver 대몬 재 시작

```
[root@master root]# service conserver restart
```

15 stage boot image 생성

이때 만들어진 소형 kernel과 stage2.tmpl이라는 이름의 boot image는 각 계산노드가 PXE로 booting할 때, 무한루프로 ping을 수행하도록 한다.

```
[root@master root]# cd /opt/xcat/stage
[root@master root]# ./mkstage
```

[참고] xCAT에서 stage의 의미: xCAT은 시스템의 setup 단계에 따라 4 개의 stage로 나뉜다.
stage1, 하드웨어 셋업: 스위치, RSA, 터미널서버, 바이오스/펌웨어 셋업
stage2, Mac collection: 각 계산노드의 MAC 주소를 collection
stage3, Management Processor 셋업: MP의 펌웨어 설정
stage4, 노드 설치: kickstart 파일 생성, kernel image 생성, node OS 설치

stage1을 제외하고 각 단계에 해당되는 스크립트를 가진다. 즉 stage2.tmpl 스크립트에서는 무한루프로 ping을 하는 스크립트가, stage3.tmpl에서는 Management Processor의 펌웨어을 업데이트 하는 코드를 실행하는 스크립트가 실행되며, stage4에서는 이에 대응되는 스크립트가 바로 redhat의 kickstart 스크립트이다. 사실 각 stage별 스크립트는 kickstart 스크립트를 필요에 따라 적절히 만든 것으로 봐도 무방하다. 각 stage의 template 파일의 위치는 다음과 같다.

stage[2, 3].tmpl 파일의 위치: /opt/xcat/stage/stage[2,3].tmpl
kickstart 파일의 위치: /opt/xcat/install/rh73/x86/[compute.tmpl, stage.tmpl]

16 각 계산노드 수동으로 리부팅하기

그리고 master 노드에서 다음의 명령을 수행하여 각 계산노드의 Mac address를 수집한다.

바로 앞의 14 단계에서 만들어진 커널과 boot image로 각 계산노드가 부팅하면서 무한루프로 ping을 수행하면, terminal server의 시리얼 콘솔을 통해 이 정보를 볼 수 있으며, 여기에서 Mac address를 caputure하여 각 노드의 Mac address를 수집하게 된다. 따라서 이 단계를 수행하기 전에 반드시 terminal server의 각 포트에 conserver.cf에 명시된 순서대로 계산노드가 연결되어 있어야 한다.

```
[root@master root]# getmacs [node range]

node1-eth0 00:07:E9:93:F8:DD
node1-eth1 00:00:5A:9A:DB:7C
node2-eth0 00:07:E9:93:F8:DD
node2-eth1 00:00:5A:9A:DB:7C
...

Auto merge mac.lst with /opt/xcat/etc/mac.tab(y/n)? y
```

17 /etc/dhcpd.con 업데이트

이제 위에서 만들어진 Mac 주소의 데이터를 이용하여 DHCP서버의 설정파일인 /etc/dhcpd.conf를 업데이트한다.

```
[root@master root]# makedhcp --allmac
```

18 Management processor의 펌웨어 업데이트

Management processor를 갖고 있는 모든 xseries 노드를 위해 다음의 명령을 수행한다. 이 명령을 통해 각 PXE를 통해 각 노드에 전달될 bootloader 설정 파일에 명시되어 있는 kickstart 파일의 스크립트의 이름을 변경한다. 즉 이러한 trick을 통해 OS 설치 kickstart 스크립트 대신 management processor의 펌웨어 설정을 업데이트하는 stage3 스크립트를 수행한다.

```
[root@master root]# nodeset [node range] stage3
```

19 각 계산노드 수동으로 리부팅하기

앞서 말한 바와 같이 이 단계에서 각 계산노드는 TFTP를 통해 소형 커널과 stage3.tmp을 다운로드한 후 자동으로 Management processor의 펌웨어를 설정 옵션을 수정한다. wcons를 통해 각 계산노드에서 다음의 작업이 진행됨을 확인할 수 있다.

```
setallalertsforsnmp:
Enable Alerts Entry : Result = success
Setting Alert Text Name : Result = success
Setting SNMP Alerts : Result = success
Setting Temp Alerts : Result = success
Setting Voltage Alerts : Result = success
Setting Tamper Alerts : Result = success
Setting Multiple Fan Alerts : Result = success
Setting Power Alerts : Result = success
Setting HD Alerts : Result = success
Setting VRM Alerts : Result = success
Setting Redundant Power Alerts : Result = success
Setting One Fan Alerts : Result = success
.......
```

이에 관한 보다 자세한 내용은 Management Processor Howto를 참조한다.

20 Management processor 설정 테스트

Management processor가 적절히 설정이 되었는지 다음의 명령어를 통해 테스트할 수 있다.

```
[root@master tftpboot]# mpascan rsa01-rsa07
rsa01:  SN#99XA497      1
rsa03:  storage02       1
rsa03:  storage03       3
rsa03:  storage04       4
rsa03:  storage05       5
rsa03:  storage06       7
rsa03:  storage07       6
rsa03:  storage08       2
rsa04:  node002 10
rsa04:  node003 12
rsa04:  node004 16
rsa04:  node005 19
rsa04:  node006 1
rsa04:  node007 3
```

```
rsa04:  node008 5
rsa04:  node009 7
rsa04:  node010 9
rsa04:  node011 8
rsa04:  node012 14
rsa04:  node013 15
rsa04:  node014 17
rsa04:  node015 18
rsa04:  node016 2
..........
1067% [root@master tftpboot]# mpncheck node001-node256
node002: rsa04
node003: rsa04
node004: rsa04
........
node020: rsa04
node022: rsa05
node023: rsa05
node024: rsa05
.........
node038: rsa05
node039: rsa05
node040: rsa05
node042: rsa06
........
```

21 원격 전원 관리 테스트

이제 각 계산노드의 전원을 원격에서 관리할 수 있다. 다음의 명령을 통해 MP에 명령이 제대로 전달되어 수행되는 지 확인하자

```
[root@master root]# rpower [node range] stat
[root@master root]# rbeacon [node range] on
```

22 리눅스 배포판 복사

리눅스 배포판(여기서는 redhat 7.3)의 CD의 내용을 복사하기 위해 다음의 명령을 수행한다.

```
[root@master root]# rpower [node range] stat
[root@master root]# rbeacon [node range] on
```

23 xCAT의 post-installation 파일 복사

```
[root@master root]# cd /opt/xcat
[root@master root]# find post -print | cpio -dump /install
혹은
[root@master root]# cp -arpf /opt/xcat/post  /install/
```

post-installation file에 올 수 있는 파일은 다음과 같다. 이 파일들은 kickstart내의 %post 단계에서 /opt/xcat/bin/postage를 통해 수행된다. 이 명령어는 /opt/xcat/install/postscripts/에 위치한 (각 패키지들을 설치하는) 스크립트를 실행한다.

추가적인 RPM 패키지: 추가적인 RPM패키지를 설치하기 위해서는 /install/post/otherrpms 에 위치시키면 된다.

GPFS: 클러스터에서 GPFS를 설치하고자하면 GPFS 소프트웨어 패키지가 여기에 /install/post/gpfs에 위치해야 한다.

Redhat Errata: /install/post/updates/rh73/i686에 rehat 7.3을 기준으로 업데이트된 RPM을 위치시킬 수 있다.

Custom Kernels: 사용자가 수정한 kernel을 /install/post/kernel에 위치시킨 후 설치할 수 있다. 또한 미리넷의 RPM 패키지도 이 경로에 위치시킬 수 있다.(참조 26-2)

Custom binary: /install/post/opt에 master노드에서 설치한 HPC 관련 병렬 라이브러리 및 컴파일러등(e.g. mpich, intel compiler, absoft compiler등)을 복사해놓으면 계산노드 설치시 이

디렉토리의 모든 내용이 그대로 복사된다. 이 작업은 custom postscript인 /opt/xcat/install/postscripts/custom/optcp를 통해 수행된다.
(주의) custom postscript가 제대로 수행되기 위해서는 postscript의 실행에 관한 dependancy가 명시된 /opt/xcat/etc/postdeps.tab가 적절히 설정되어 있어야 한다(참조 6.5).

위의 post-script는 각 rpm 및 바이너리를 /install/post가 아닌 /post로부터 찾는다. 이것은 kickstart의 마지막에 실행되는 postscripts 들 중 /opt/xcat/install/postscripts/mountpost에서 master:/install/post를 /post로 마운트로 하도록 하기 때문이다.

24 root의 SSH keys 생성

```
[root@master root]# /opt/xcat/sbin/gensshkeys root
```

25 /etc/exports 파일업데이트

/etc/exports 파일에서 /install디렉토리도 export하도록 update한다.

```
[root@master root]# echo "/install *(ro,sync,no_root_squash)"
>> /etc/exports
```

redhat의 경우, 서비스 대몬을 추가하고 재시작시키기 위해 다음의 명령을 내릴 수 있다.

```
[root@master root]# chkconfig --add nfs
[root@master root]# service nfs restart
```

26 미리넷 GM RPM build

26-1 kernel 준비

현재 Master노드에 설치된 kernel 과 일치하는 kernel source가 적절한 경로에 위치해야 하며 다음의 과정을 통해 kernel을 준비해야 한다.

```
1. 이전에 컴파일된 binary와 kernel 설정파일등 이전의 작업내용을 지운다.(선택사항)
# cd /usr/src
# ln -s linux-2.4 linux
# cd /usr/src/linux-2.4
# make mrproper

2. Makefile 이 현재 실행중인 kernel 버전과 일치하는지 확인한다.

VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 19
EXTRAVERSION = -SMP

# uname -r
2.4.19-SMP

3. 커널 설정파일을 다시 만든다.
# cp configs/(matching config) .config

4. 커널 컴파일을 다음단계까지 진행한다.
# make oldconfig
# make dep
# make modules (ctrl-c after 30 seconds)
```

26-2 GM 소스를 RPM으로 빌드

www.myri.com에서 다운로드한 GM 소스를 /tmp에 복사한후, 다음의 과정을 통해 GM 소스를 RPM으로 build 한다. /usr/src/redhat/RPMS/i686/에 RPM 버전의 GM이 만들어진다. 이 만들어진 패키지를 /install/post/kernel에 위치시킨다.

```
[root@master root]# cd /opt/xcat/build/gm-2.0
[root@master root]# ./gmmaker gm-2.0.12_Linux.tar.gz
```

26-3 GM active mapper 지정

$XCATROOT/etc/site.tab에 GM이 mapping 정보를 초기화할 때, active mapper가 될 host를 지정해야 한다.

```
.......
mapperhost                    node200,node201,db01
.......
```

myrinet GM 2.0 이상은 더 이상 static mapper를 사용하지 않으며, 위와 같이 mapping process에서 잠정적인 active mapper의 역할을 할 host를 지정해야 한다. 참고로 위의 두개의 노드는 GM의 설치가 끝나면, /etc/gm_mapper에 두개의 파일을 가지게 된다. GM 2.0에 대해서는 http://www.myri.com/scs/mapper_gm2.html를 참조하기 바란다.

26-4 IP over Myrinet 세팅

xCAT에 의해 만들어진 GM RPM 패키지는 설치시 IP over Myrinet 세팅을 자동으로 시도한다. 따라서 /etc/hosts 파일에 이에 대한 hostname이 적절히 들어있는지 확인후 makedns 명령을 통해 DNS 정보를 업데이트해야 한다.

```
172.20.100.1  node001-myri0
172.20.100.2  node002-myri0
.......
```

27 Kickstart Template 파일 수정

/opt/xcat/installl/rh73/x86/[compute,stage].tmpl등의 kickstart template 파일을 적절히 수정한다. 6-6의 nodetype.tab에서 정의된 template 파일에 대해서만 이 작업을 수행하면 된다. 이것의 설정에 대한 자세한 내용은 nodeinstall-HOWTO를 참조하기 바란다.

28 계산노드 설치

이제 각 노드를 설치하면 된다. rinstall 또는 winstall등을 사용할 수 있다. 한번에 설치할 수 있는 노드 수는 32 노드 정도이며, 다른 방법으로 staging을 이용한 설치를 고려할 수 있다. staging을 이용하여 설치를 진행할 경우 다음과 같이 명령을 내릴 수 있다.

```
# winstall -t 8 stage

# winstall -t 8 rack2g,-stage
```

위의 -t 8 옵션은 터미널창을 통해 설치과정을 보고자 할때 사용한다. 터미널창의 수는 가로로 8 개 씩 보여 진다.

간단히 설치과정을 설명하면 다음과 같다.

28-1 "winstall node001"(즉, rinstall node001) 명령은 "nodeset node001 install"을 호출한다.

28-2 “nodeset node001 install"은 nodehm.tab의 다음의 “node001 mp,mp,NA, mp, mp,conserver, rcons,mp,rcons,pxe,bcm5700, NA,Y, NA,NA,def” 줄을 참조하여 해당노드가 pxe를 통해 설치해야 함을 알고 "nodeset.pxe node001 install"을 호출한다.

28-3 “nodeset.pxe node001 install"은 nodetype.tab에 있는 “node001 rh73,x86,stage”을 참조하여 해당 노드에 설치될 OS가 redhat 7.3-x86 버전임을 파악하며, template 파일을 참조하여 적절한 kickstart 설정파일과 bootloader 설정파일을 만든다. 다음은 kickstart 파일 및 bootloader. 설정파일을 만들기 위해 다음의 template 파일을 사용함을 보여준다.

```
/opt/xcat/install/rh73/x86/install.pxe
→ /tftpboot/xcat/rh73/x86/compute-stage01.pxe
→ /tftpboot/pxelinux.cfg/node001
→ /tftpboot/pxelinux.cfg/[Hex_IP_address of node001]

/opt/xcat/install/rh73/x86/[compute.tmpl, stage.tmpl]
→ /install/scripts/rh73/x86/compute-stage[00-12]
```

또한 /opt/xcat/install/rh73/x86/bin/mkinstall을 통해 kernel image를 /tftpboot/xcat/rh73/x86/로 복사하고 initrd.img를 적절히 수정한 후 같은 경로로 복사한다.

install시 사용하게 되는 node001의 노드정보를 알기 위해서는 다음의 명령어를 사용할 수 있다.

```
# nodels node001 install
node001: rh73-x86-stage-stage
```

nodename: OS Version-Architecture-Template_name (nodetype.tab)-Resource_group(noderes.tab)의 정보를 볼 수 있다.

28-4 "rinstall node001"은 "rpower node001 boot"를 호출하여 node001이 reboot하면서 자동으로 재 설치되도록 한다.

29 계산노드의 SSH hostkey 수집

설치후 다음의 명령을 통해 각 노드의 SSH hostkey를 수집한다. 이것은 ssh-keyscan 명령을 사용한다.

```
# makesshgkh [node range]
```

30. 계산노드의 system date 체크

psh를 이용한 다음의 명령을 사용하여 각 노드의 시간이 일치하는지 확인한다.

```
# psh [node range] date ; date
```

31 (GM 2.0 이하인 경우) GM routes 생성

미리넷 GM의 버전이 2.0 이상이 아닌 경우 GM routes를 다음의 명령으로 만든다.

```
# makegmroutes [node range]
```

32 (선택사항) PBS와 Maui 설치

```
cd /tmp
/opt/xcat/build/pbs/pbsmaker OpenPBS_2_3_16.tar.gz scp
/opt/xcat/build/maui/mauimaker maui-3.2.5.tar.gz
genpbs [noderange]
. /etc/profile.d/pbs.sh
showq (모든 노드를 볼 수 있어야 한다)
rpm -i /opt/xcat/src/perl-Curses-1.06-208.src/rpm
cd /usr/src/packages/SPECS
rpm -ba perl-Curses.spec
rpm -i /usr/src/packages/RPMS/ia64/perl-Curses-1.06-208.ia64.rpm
pbstop (모든 노드를 볼 수 있어야 한다)
```

본 테라클러스터의 경우 PBS Pro가 설치되어 있으며 따라서 위의 과정을 따르지 않고 별도의 과정을 통해 설치되었다.

33 사용자 계정 추가

(Modified) 사용자 계정을 추가하기 위해 다음의 과정을 따를 수 있다. 필요시 groupadd 명령을 통해 사용자가 속할 그룹을 먼저

만들고 이 과정을 진행해야 한다.

```
# addclusteruser
Enter username: bob
Enter group: users
Enter UID (return for next): 501
Enter absolute home directory root: /home
Enter password (blank for random): B0vHw0bL
cd /etc
cp passwd passwd.CYA
cp group group.CYA
cp /etc
prsync -craz passwd group noderange,-$(hostname -s):/etc
```

xCAT의 경우 별도의 사용자 login노드를 고려하지 않는다. 따라서 위의 경우 master노드에 사용자 계정을 추가한 후 passwd 및 group 파일을 각 계산노드에 sync하면 사용자 계정의 추가가 끝난다. 이 과정에서 ssh의 key를 생성하며, 모든 계산노드에는 ssh를 통해 password없이 access 가능하게 된다.

하지만 본 테라 클러스터는 4개의 로그인 노드를 master 노드와는 별도로 두고 있으며, 따라서 위의 사용자 추가과정에 약간의 수정을 해야 했다. 사용자의 추가는 4번째 로그인 노드인 lg04에서 해야 하며, 위의 “addclusteruser" 명령을 그대로 사용할 수 있다. 다음은 테라 클러스터에서 사용자를 추가하기 위한 수정된 방법이다.

33-1 (lg04노드에서 다음의 과정을 진행한다) 사용자가 사용할 uid를 결정하기 위해 /etc/passwd 파일의 마지막 user의 uid를 확인한다.

33-2 필요시 groupadd 명령을 통해 사용자 그룹을 만든다.

```
# groupadd -g 531 test001
```

위에서 확인한 uid의 다음의 uid를 이용하여 사용자 group을 만든다. 즉 여기에서는 uid와 같은 숫자를 gid로 사용했다.

33-3 addclusteruser를 사용하여 user를 추가한다.

```
# addclusteruser
Enter username: test001
Enter group: test001
Enter UID (return for next): 531
Enter absolute home directory root: /home1
Enter passwd (blank for random): dontdoit
Creating home directory /home1/test001
Adding user: useradd -u 531 -g test001 -d /home1/test001 test001
Setting passwd for test001 to dontdoit
Changing password for user test001.
passwd: all authentication tokens updated successfully.
Now, updating user info to the login, debug, sched, and storage nodes
db02:    User test001 added
............
```

여기에서 원래의 addclusteruser 스크립트에서 추가된 내용은 다음과 같다.

```
#    echo "₩nDon't forget to run pushuser [noderange] $USERNAME₩n"
     echo "Now, updating user info to the login, debug, sched, and storage nodes"
  pushuser4s -l lg01-lg03,debug $USERNAME                                  ①
  psh debug usermod -G vtune $USERNAME                                     ②
  pushuser compute,spare $USERNAME                                         ③
  pushuser storage $USERNAME                                               ④
  pushuser sched $USERNAME                                                 ⑤
  psh sched "/usr/sbin/changeshell -e admin,system 2>/dev/null"            ⑥
  psh storage "/usr/sbin/changeshell -e admin,system ₩
```

```
2>/dev/null"                                                    ⑦
```

① xCAT의 pushuser 명령어는 /etc/passwd /etc/group에서 사용자의 계정정보가 들어 있는 줄을 읽어서 다른 노드의 /etc/passwd, /etc/group에 첨가한다. 이를 통해 원격의 노드에 새로운 사용자 계정을 추가하게 되는 결과를 얻는다. 하지만 /etc/shadow에 들어 있는 해당 사용자의 정보는 복사하지 않으며, 이것은 외부에서 암호인증을 통해 억세스를 허가를 받아야 하는 로그인/디버깅 노드의 경우 문제가 된다. 따라서 여기에서 사용한 pushuser4s 명령어는 /etc/shadow의 정보도 복사하도록 하기 위해 수정된 스크립트이다.

② 디버깅 노드의 경우 Vtune이 설치되어 있다. 하지만 추가된 사용자가 Vtune를 사용하기 위해서는 vtune 그룹에 포함되어야 한다.

③, ④, ⑤ 위의 ①에서 설명한 바와 같이 추가된 사용자정보를 compute, spare, storage, sched 노드에 추가한다.

⑥, ⑦ 여기에서는 관리상 필요한 적절한 사용자를 제외하고 일반사용자의 sched, storage 노드에 억세스를 차단시킨다. 이를 위해 일반사용자의 /etc/passwd의 쉘 정보를 /sbin/nologin 으로 변경시킨다. 이것은 xCAT에서 제공하는 명령어가 아니다.

34 (선택사항) PBS/Maui를 테스트

이것은 xCAT에 채택한 기본 병렬 작업 큐잉 및 스케쥴링 시스템인 OpenPBS/Maui를 사용하는 경우에 해당된다. 일반 유저로 다음을 테스트한다.

```
bob@head01:~> qsub -l nodes=2,walltime=1:00:00 -I
qsub: waiting for job 0.head01.foobar.org to start
qsub: job 0.head01.foobar.org ready
```

```
----------------------------------------
Begin PBS Prologue Thu Dec 19 14:17:53 MST 2002
Job ID: 0.head01.foobar.org
Username: bob
Group: users
Nodes: node10 node9
End PBS Prologue Thu Dec 19 14:17:54 MST 2002
----------------------------------------

Note the Nodes: line. Try to ssh from node to node and back to the user
node that started qsub:

bob@node10:~> ssh node9
bob@node9:~> exit
logout
Connection to node9 closed.
bob@node10:~> ssh head01
bob@head01:~> exit
logout
Connection to head01 closed.
bob@node10:~> exit
logout

qsub: job 0.head01.foobar.org completed

Now try to ssh back to the nodes that were assigned, you should be
denied:

bob@head01:~> ssh node9
14653: Connection close by 199.88.179.209
```

참고로 위의 내용에 대해 간단히 설명하면, xCAT의 PBS/SSH 설정에서, 사용자의 계산노드에의 억세스는 사용자가 신청한 job이 할당된 특정 계산노드에만 가능하다. 즉 이러한 경우를 제외하고는 일반적으로 일반사용자의 계산노드로의 억세스는 차단된다.

35 컴파일러와 라이브러리 설치

xCAT HPC Benchmark HOWTO를 참조한다.

아래의 내용은 xCAT HPC Benchmark HOWTO에 나와 있는 설명과 다를 수 있다.

35-1 GNU compiler 설치

redhat 7.3의 패키지를 그대로 사용했다.gcc의 버전은 gcc-2.96-113이며, glibc는 redhat 7.3의 공식 업데이트 버전 사용했다.

```
glibc-2.2.5-44.i686.rpm
glibc-debug-2.2.5-44.i686.rpm
```

35-2 absoft 컴파일러 설치

35-2-1 특정 노드(본 클러스터의 경우, sched 노드)에 absoft의 RPM 버전을 설치해야 하며, ABSoft License Manager Server 대몬을 구동시켜야 한다. 이 대몬을 구동시키려면 absoft라는 계정을 만들어야 한다.

```
라이센스 정보: 256 node floating license/concurrent users: 2
설치경로: /opt/absoft
버전: ABSoft Pro Fortran95 9.0 32Bit(Intel, IA32)
RPM 버전: absoft_32bitfortran95-9.0-1.i386.rpm
라이센스: /opt/absoft/license.dat
Compiler 경로: /opt/absoft/bin
라이센스 매니저 대몬: /opt/absoft/bin/absoftlmd
```

35-2-2 lg01~04, db01~04, node001~256에도 역시 absoft의 RPM 버전을 설치한다.
/opt/absoft/license.dat 파일에 다음의 SERVER와 PORT 정보를 기재해야 한다.

```
SERVER: sched
PORT: 1287
```

35-2 Intel compiler 설치

Intel compiler로 다음의 패키지가 설치되어 있다.

C compiler 8.0 :	/opt/intel/cc_80	l_cc_p_8.0.055.tar.gz
Fortran compiler 8.0:	/opt/intel/fc_80	l_fc_p_8.0.034.tar.gz
MKL 6.1:	/opt/intel/mkl61	l_mkl_p_6.1.006.tar.gz
Vtune 2.0:	/opt/intel/vtune	vt_cli_2.0.348.tar.gz

Vtune package는 디버깅 노드에만 설치되어 있다.(Appx. 참조)

35-3 PGI compiler 설치

db01~04와 lg01~04까지 PGI Soft 컴파일러 및 라이센스를 설치했다. 각 계산노드에는 prsync를 이용해서 복사했다. pghpf등 PGI에서 제공하는 자동 병렬화 툴을 사용하고자 할 경우, 라이센스 제한에 걸릴 수 있다. 그러나 일반적인 mpich의 경우 256노드에서 병렬작업을 수행할 수 있다.

```
라이센스 정보: PGI workstation 10 packs
(single user/node-locked/up to 4 cpus)
설치경로: /opt/pgi
버전: PGI Linux86 5.1 (32bit)
라이센스 파일: /opt/pgi/license.dat
Compiler 경로: /opt/pgi/linux86/5.1/bin
라이센스 매니저 데몬 스크립트: /etc/rc.d/init.d/pgi_lmgrd 혹은
                                /opt/pgi/linux86/5.1/bin/lmgrd.rc
```

36 로그인노드에 MPICH-GM 설치

myrinet-HOWTO를 참조한다. 다음의 과정을 따르면 master 노드에서 간단히 설치할 수 있다.

36-1 먼저 mpe의 jumpshot을 사용하기 위해서는 java sdk를 설치해야 한다. 이를 위해 master노드에 j2sdk-1_4_1_07-fcs-linux-i586.rpm를 설치했다.

36-2 26번에서 Myrinet GM을 설치했다. 다음 단계를 진행하기

위해서는 반드시 적절한 GM이 설치되어 있어야 한다. 하지만 Master 노드에 Myrinet Card가 있을 필요는 없으며, 따라서 gm 모듈이 load되어 있지 않아도 된다.

36-3 mpich source 코드를 다운로드하여 /opt/xcat/build/mpich 혹은 /tmp 디렉토리에 복사한다.

36-4 다음의 과정을 통해 각 컴파일러 별 mpich-gm을 설치한다.

```
cd /opt/xcat/build/mpi
./mpimaker 1.2.5..10:2.0.12-2.4.20-28.7smp-i686 smp intel32  ssh
                                          //          gnu   //
                                          //          pgi   //
                                          //          absoft //
```

명령어의 format은 다음과 같다. 여기에서 gm version은 gm이 설치된 경로의 path를 참조하면 된다. 즉 위의 경우, gm 2.0.12가 /opt/gm-2.0.12-2.4.20-28.7smp-i686 에 설치되어 있다.

```
mpimaker [mpich version:gm version] [up|smp] [gnu|pgi|] [ssh|rsh]
```

위의 명령의 실행이 정상적으로 종료되면 다음의 경로에 mpich-gm이 설치가 된다.

```
/opt/mpich/1.2.5..10/gm-2.0.12-2.4.20-28.7smp-i686/smp/intel32/
                                          //          gnu
                                          //          pgi
                                          //          absoft
```

ethernet 기반의 mpich를 설치하기위해서는 위의 명령을 그대로 사용할 수 있다. 단 gm 정보를 입력할 필요는 없다.

```
/mpimaker 1.2.5.2 smp intel32  ssh
```

37 HPL benchmark 테스트 후 결과를 top500.org에 제출

xCAT HPC Benchmark HOWTO를 참조한다.

38 Enjoy your cluster

Appendix

A1 계산노드 전체 재설치

1. /opt/xcat/install/rh73/x86/stage.tmpl 을 요청한 파티션으로 수정
2. /opt/xcat/install/rh73/x86/compute.tmpl 을 요청한 파티션으로 수정
3. 전체 노드 shutdown
4. stage노드 설치 (winstall stage)
5. 각 랙별로 compute 노드 설치 (winstall rack2g,-stage)
6. 재설치 되지 않은 노드 검색 (파티션구성으로 구별)
7. 재설치 되지 않은 노드 설치
8. ping test

A2 시스템의 교체 후 계산노드의 재설치

node033를 재설치하는 경우를 예로 들어 설명 한다.

1. MAC update

1-1. 온보드 이더넷 MAC 메모 - F1키를 눌러 BIOS화면으로 들어가면 확인가능

1-2. /opt/xcat/etc/mac.tab 화일 편집 - 새로 교체된 MAC으로 수정
1-3. /etc/dhcpd.conf 화일편집 - 새로 교체된 MAC으로 수정
1-4. dhcp서비스 재시작: service dhcpd restart

2. MP setup
2-1 해당 노드를 관리하는 rsa로 Web 접속 (USERID/PASW0RD)
2-2 Acess remote ASM 메뉴의 해당노드로 들어간후 시스템 세팅에서 이름을
해당노드 이름으로 변경, 저장 후 web에서 로그아웃
3. winstall node033

4. ssh-key update
4-1. node033의 public key를 /opt/xcat/etc/gkh에서 지운다.
4-2. ssh node030 'ssh-keyscan -t rsa node030' >> /opt/xcat/etc/gkh
혹은 간단히 makesshgkh node030
login 노드의 경우 ssh host file의 update는 6-5의 과정이 끝난 후에 수행한다.

5. gm update
5-1. rpm -e --nodeps gm
5-2 rpm -ivh gm-2.0.12_Linux-2.4.20-28.7smp.i686.rpm
위의 gm 설치작업은 6-6의 postscripts.tab 설정과 26-2의 rpm 복사를 /install/post/kernel/에 했을 경우 할 필요가 없다.
5-3 scp node029:/etc/rc.d/init.d/gm node030/etc/rc.d/init.d/
5-4 service gm start

6. /opt/ 디렉토리, rsh 관련 설정 파일, 사용자 계정 파일 update
6-1 ssh node032
6-2 prsync -ravzc --delete /opt/[a-z]* node033:/opt/

이 작업은 23의 post-installation 설정에서 가장 최신의 /opt 디렉토리 복사를 미리 해놓은 경우 할 필요가 없다. 로그인/디버깅/master/sched 노드의 경우 PGI license를 update해야 한다. 이를 위해 https://www.pgroup.com/License에 접속하여 라이센스를 갱신한 후 작업을 해야 한다.

6-3 prsync /etc/pam.d/r* node033:/etc/pam.d/

6-3 prsync /etc/pam.d/ssh node033:/etc/pam.d/

6-4 prsync /etc/xinetd.d/r* node033:/etc/xinetd.d/

6-5 prsync /etc/passwd /etc/group node033:/etc/

login node의 경우 /etc/ssh/ssh_host* 도 sync 필요(즉 ssh hostkey file을 동일하게 해야 hamel.kisti.re.kr 대표주소를 이용한 접속시 hostkey 변경에 따른 경고 메시지를 피할 수 있음)

login, debug,sched,master node의 경우 /etc/shadow도 sync 해야 함.

6-6 ssh node033

6-7 service xinetd restart

7 pbs pro 설치

7-1 rpm -ivh pbs-mom-5.4.0.40152-0.i386.rpm

23에서 /install/post/otherrpms에 미리 복사를 해놓은 경우 이 작업을 할 필요는 없다

7-2 /etc/pbs.conf 수정: PBS_SERVER=sched

7-3 /var/spool/PBS/mom_priv/config 수정: $clienthost sched