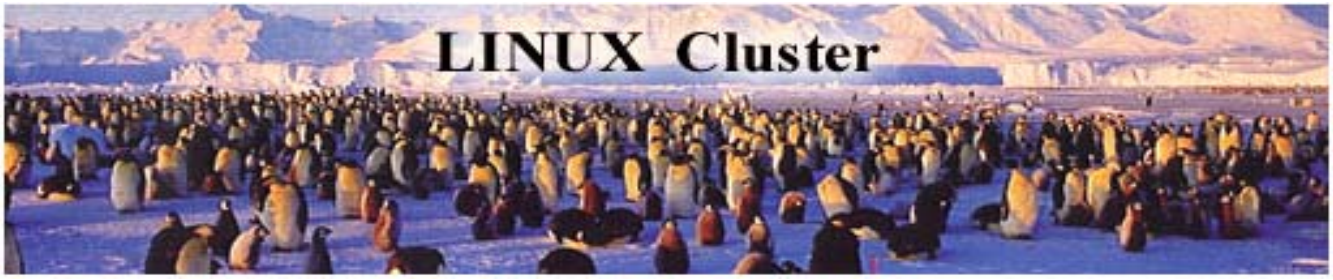
Linux Advanced Management

Kim Wan Hee , FTSS
Last Update 2007/07/24



Index

본 문서는 IBM Residency Program 에 참여하여 주신 Business Partner 분들께서 만드신 자료 입니다. 바쁘신 시간중에도 5일간 교육에 참석 해주신 Engineer 분들께 감사 드립니다.

1. xCAT (Extreme Cluster Administration Toolkit)

xCAT (Extreme Cluster Administration Toolkit)은 Linux Cluster와 같이 많은 동일한 하드웨어 컴포넌트를 사용하는 환경에서 관리자가 Build, Configure, Administer, 그리고 Maintain 작업을 중앙 집중화된 환경에서 효율적으로 관리 가능케 하는 Perl/Unix Shell Script Base의 Tool입니다. 아래와 같은 시스템 환경에서 구현이 가능하다.

1. High Performance (HPC): such as computing physics, seismic, CFD, FEA, weather, bioinformatics and other simulations.
2. Horizontal Scaling (HS): such as Web farms.
3. Administrative: A very convenient platform, although non-traditional, to install and administer a number of Linux machines.
4. Microsoft Windows and other Operating Systems: With xCAT's cloning and imaging support, it can be used to rapidly deploy and conveniently manage clusters with compute nodes that run Windows or any other operating system.

1. OS/Distribution support Any OS on compute nodes via OS agnostic imaging support.
2. Hardware Control Remote Power control (on/off state) through IBM Management Processor Network, BMC, and/or APC Master Switch.
3. Hardware Control Remote software reset (rpower).
4. Hardware Control Remote Network BIOS/firmware update and configuration on IBM hardware.
5. Hardware Control Remote OS console with pluggable support for a number terminal servers.
6. Hardware Control Remote POST/BIOS console through the IBM Management Processor Network and with terminal servers.
7. Boot Control Ability to remotely change boot type (network or local disk) with syslinux.
8. Automated parallel install using scripted RedHat kickstart, SUSE Linux autoyast, on ia32, x86_64, ppc, and ia64.
9. Automated parallel install using imaging with other Linux distributions, Windows, and other operating systems.
10. Automated network installation with supported PXE NICs, with etherboot or BootP on supported NICs without PXE.
11. Monitoring hardware alerts and email notification with IBM's Management Processor Network and SNMP alerts.
12. Monitoring remote vitals such as fan speed, temperature, and more with IBM's Management Processor Network.
13. Monitoring remote hardware event logs with IBM's Management Processor Network/IPMI Interface.
14. Administration utilities such as parallel remote shell, ping, rsync, and copy.
15. Administration utilities such as remote hardware inventory with IBM's Management Processor Network.
16. Software Stack PBS and Maui schedulers to build scripts, documentation, automated setup, extra related utilities, and deep integration.
17. Software Stack Myrinet to automate setup and installation.
18. Software Stack MPI to build scripts, documentation, and automated setup for MPICH, MPICH-GM, and LAM.
19. Usability command line utilities for all cluster management functions.
20. Usability single operations can be applied in parallel to multiple nodes with very flexible and customizable group/range functionality.
21. Flexible support for various user defined node types.
22. Diskless support using warewulf.

<xCAT을 사용하기 위한 기본 HPC 환경>

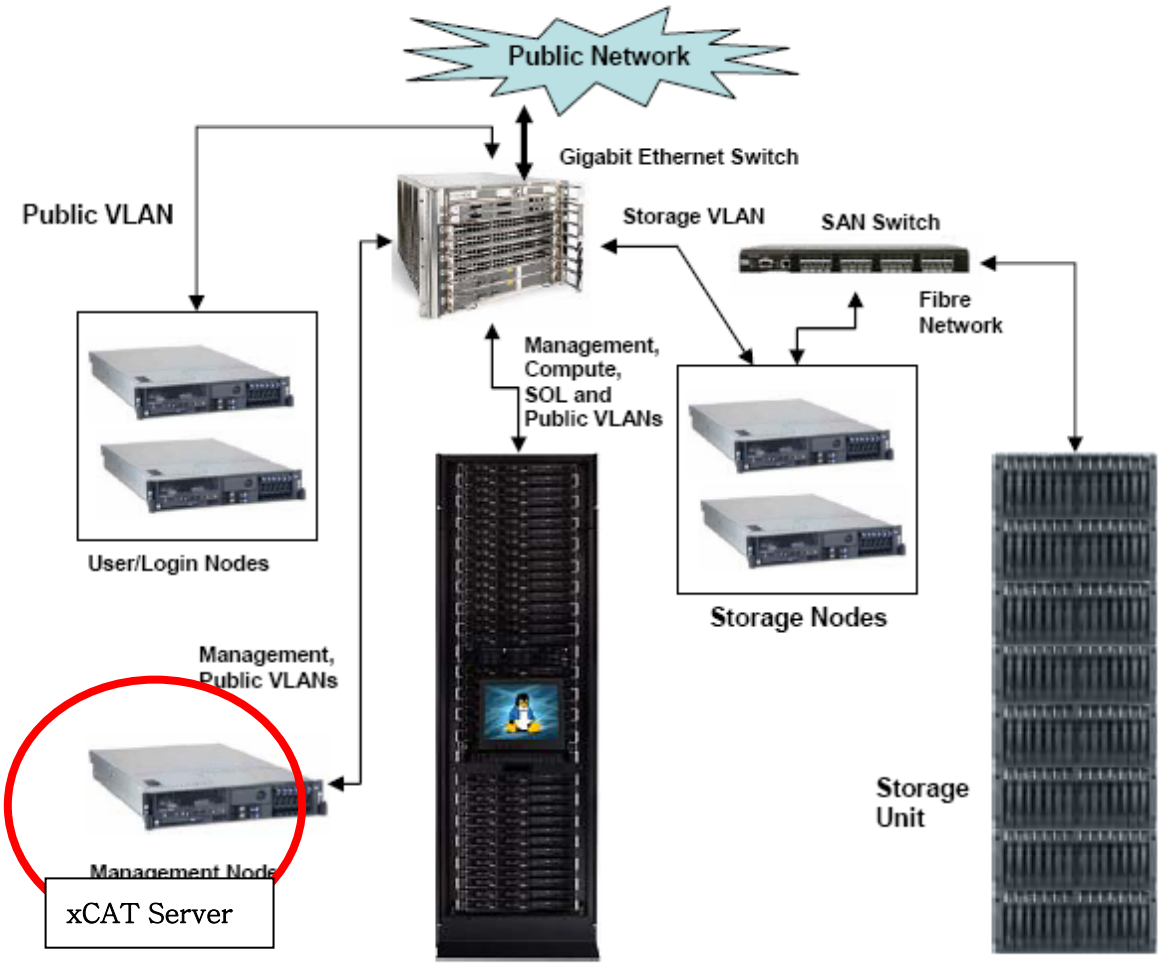


Figure (i) Conceptual view of a Cluster

<HPC를 구성하고 있는 중요 컴포넌트>

Management node -- Management node functionality can reside on a single server or multiple servers. In a single server environment, the management server operates in standalone mode. The management node is connected to the compute nodes over the Management VLAN on the switch. The management server is where the xCAT software is installed and is used exclusively to control the compute nodes using various xCAT functionalities: OS installation, remote power control, firmware updates, Serial-over-LAN, etc. In our conceptual cluster, there is one management node.

User/login nodes -- Ideally, the compute nodes of a cluster should not accept external connections and should only be accessible to system administrators through the management server. System users can log in to user nodes (or login nodes) in order to run their workloads on the cluster. Each user node consists of an image with full editing capabilities, the required development libraries, compilers, and everything else required to produce a cluster-enabled application and retrieve results.

Storage servers and disks -- You can connect several storage servers to a diskbased backend using various mechanisms. Connecting storage to the cluster can be direct or through a storage area network (SAN) switch, either by fiber, copper, or a mixture of the two. These servers provide shared storage access to the other servers within the cluster. If data backup is required, connect

the backup device to the storage server using an extra copper or fiber link. For the example cluster, the storage back end is a single entity, providing shared file system access across the cluster.

Compute nodes -- These nodes run the cluster workload, accepting jobs from the scheduler (if a job scheduler is used on the cluster). The compute nodes are the most disposable part of the cluster. The system administrator can easily reinstall or reconfigure them using the management server.

1.1. xCAT 설치

http://www.alphaworks.ibm.com/tech/xCAT 에서 Download Link 에서 관련 파일 Download함

- 받은 파일을 /opt 에 풀어줌

```
# 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 (extract only if you have IBM xSeries nodes)

cp /opt/xcat/samples/etc/* /opt/xcat/etc
```

1.2. xCAT 환경 설정

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

#copycds → 설치 할 OS image를 저장함.(xCAT설치 후 어느 단계에서나 진행 가능)
```

```
[root@x305 sbin]# cat /root/.bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin
BASH_ENV=$HOME/.bashrc
USERNAME="root"

export USERNAME BASH_ENV PATH

# xCAT Profile
export XCATROOT=/opt/xcat
export XCATPREFIX=/opt/xcat
export PATH=$PATH:$XCATROOT/bin:$XCATROOT/sbin
```

/etc/man.config 에 [MANPATH /opt/xcat/man] 추가

1.3. Time Service 등록

```
# mv -f /etc/ntp.conf /etc/ntp.conf.ORIG
# touch /etc/ntp.conf
# vim /etc/ntp.conf
-------------------------------------
server 127.127.1.0
fudge 127.127.1.0 stratum 10
driftfile /etc/ntp/drift
-------------------------------------

* Time서버에서의 Timezone 설정

setup
setclock OR hwclock -w
chkconfig --add ntpd
service ntpd start

* NTP 클라이언트 설정
server mgt.cluster.com
driftfile /etc/ntp/drift
multicastclient
broadcastdelay   0.008
authenticate no
keys             /etc/ntp/keys
trustedkey       65535
requestkey       65535
controlkey       65535
ntp서버에서 /etc/ntp디렉토리안에 있는 drift파일 복사

* NTP 데몬 테스트

[root@blade329 STREAM]# ntpstat
synchronised to NTP server (172.20.0.1) at stratum 12
   time correct to within 23 ms
   polling server every 128 s
```

1.4. Compute Node SOL (Serial Over LAN) Enable

For IBM System x 3550, 3650, and 3455
Flash the Management Processor (BMC) to the latest version.
Flash BIOS to the latest version.
Remove the power cord for 10 seconds.
Restore the power cord.
Reboot and press F1 to enter the BIOS configuration.
Configure the BIOS settings for optimum performance and then edit the Devices and I/O Ports as shown below:
Devices and I/O Ports
• Serial port A: Port 3F8, IRQ 4
• Serial port B: Disabled
• Remote Console Redirection
o Remote Console Active: Enabled
o Remote Console COM Port: COM 1
o Remote Console Baud Rate: 19200
To use Remote Console Text Emulation: VT100/VT220
Configure the text emulation settings as listed below:

• Remote Console Keyboard Emulation: VT100/VT220
• Remote Console After Boot: Enabled
• Remote Console Flow Control: Hardware
Configure the "Startup" settings as listed below:
From the main menu, select Startup Sequence and configure the settings as follows:
• First Startup Device: CD ROM
• Second Startup Device: Diskette Drive 0
• Third Startup Device: Network
• Forth Startup Device: Hard Disk
• Wale On LAN: Disabled
• Planer Ethernet PXE/DHCP: Planer Ethernet 1
43
• Boot Fail Count: Disabled

For IBM System BladeCenter Server
Flash the Management Processor (BMC) to the latest version.
Flash BIOS to the latest version.
Remove the power cord for 10 seconds.
Restore the power cord.
Reboot and press F1 to enter the BIOS configuration.
Configure the BIOS settings for optimum performance and then edit the Devices and I/O Ports as shown below:
Devices and I/O Ports
• Serial port A: Auto configured
• Serial port B: Auto configured
• Remote Console Redirection
o Remote Console Active: Enabled
o Remote Console COM Port: COM 2
o Remote Console Baud Rate: 19200
To use Remote Console Text Emulation: VT100/VT220
Configure the text emulation settings as listed below:
• Remote Console Keyboard Emulation: VT100/VT220
• Remote Console After Boot: Enabled
• Remote Console Flow Control: Hardware
Configure the "Startup" settings as listed below:
From the main menu, select Startup Sequence and configure the settings as follows:
• First Startup Device: CD ROM
• Second Startup Device: Diskette Drive 0
• Third Startup Device: Network
• Forth Startup Device: Hard Disk
• Wale On LAN: Disabled
• Planer Ethernet PXE/DHCP: Planer Ethernet 1
43
• Boot Fail Count: Disabled

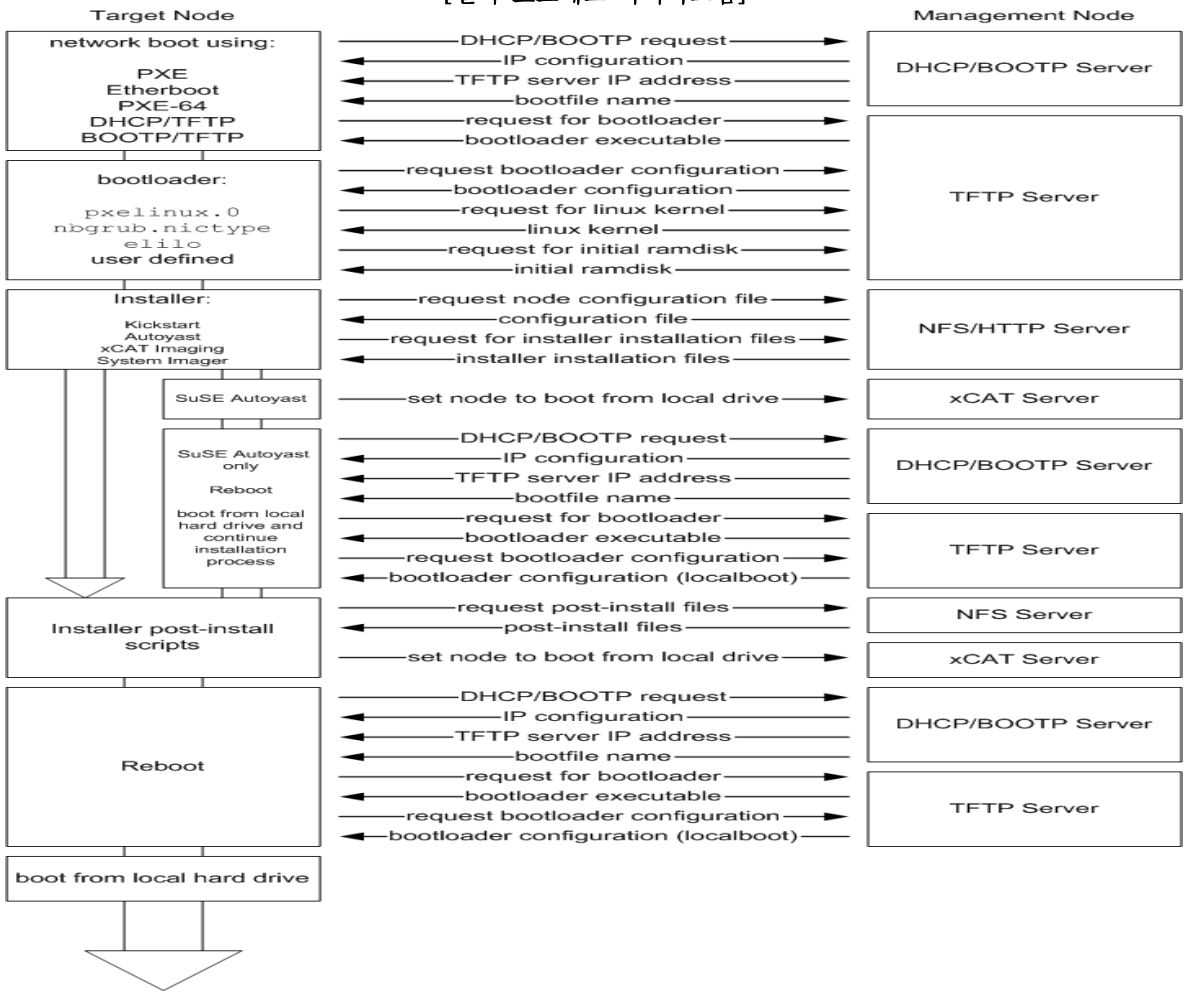
1.5. xCAT Config File

Cluster 정의, 환경 구성파일이 있는 경로는 /opt/xcat/etc/* 에 있어야 하는데 압축을 풀면 etc Directory 는 없습니다. (이때 /opt/xcat/samples/etc/* 에 예제 파일이 있으니 참고해서 만들어 사용을 하면 됩니다.) 환경설정의 더 자세한 내용을 원한다면 아래의 Site 의 Redbook 을 참고 하면 됩니다.
환경 구성 Directory 에는 각 Node,Switch,Terminal server, Node NIC MAC 등 xCat 내부 노드의 모든 정보들이 들어 가게 됩니다.

*각 구성 파일 이름 목록 site.tab / nodehm.tab / nodelist.tab / nodepos.tab / noderes.tab / nodetype.tab / passwd.tab / postscripts.tab / postdeps.tab / snmptrapd.conf / networks.tab mac.tab (loaded with non-collectable MACs, e.g. terminal servers, switches, RSAs, etc...)	
*Terminal server 관련 구성 Table 이름 목록	conserver.tab conserver.cf
*MAC Address 정보를 Cisco Switch 에서 수집을 하기 위한 구성 정보	cisco.tab summit48i.tab blackdiamond.tab
*IBM eServer xSeries Management Processor(예 RSA Card)	mpa.tab mp.tab
*APC Master Switch 를 위한 테이블	apc.tab
*APC Master Switch Plus 를 위한 테이블	apcp.tab
*xCAT flash support 를 위한 테이블	nodemodel.tab
*EMP 지원 구성 테이블	emp.tab
*Baytech 지원 구성 테이블	baytech.tab
*xCat GPFS 지원 구성 테이블	gpfs.tab
*IPMI 지원 구성 테이블 (BMC를 이용한 node management)	ipmi.tab
*Blade서버 슬롯위치와 switch port번호 mapping 테이블	bcnet.tab

1.6. xCAT Node Installation Process

[설치 프로세스 다이어그램]



1.7. 모든 노드의 구성

IANS:

Update all firmware to latest levels.
Configure firmware/BIOS/CMOS to NEVER prompt for anything.
Configure network to boot before HD.
Enable management processor.
Redirect POST/BIOS out serial if possible.

* rbootseq 명령으로 Bladecenter boot order변경
* rbootseq noderange n,c,hd0,f

1.8. /etc/hosts 파일 정의

```
[root@mgt pxelinux.cfg]# vi /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1               localhost.localdomain localhost   → 여기 hostname이 들어가지 않도록 주의

192.168.12.74    mgt       mgt.cluster.com

192.168.70.125   bc1
192.168.70.127   sw1
192.168.70.128   sw2

192.168.12.1     blade1
192.168.12.2     blade2
192.168.12.3     blade3
192.168.12.4     blade4
```

1.9. Blade Center Node Installation

1) /opt/xcat/etc 의 구성파일 편집

/opt/xcat/etc/site.tab → Cluster 전체 환경 구성 편집
/opt/xcat/etc/nodelist.tab → 각 Node 의 Group 편집
/opt/xcat/etc/noderes.tab → 각 노드들이 설치 될때 tftp,nfs등의 서버 경로지정
/opt/xcat/etc/nodetype.tab → 각 Node 의 Type 설정
(node를 group으로 정의할때, nodelist 및 nodetype에 등록을 해야 한다.)
/opt/xcat/etc/nodehm.tab → Node Hardware Management 설정
/opt/xcat/etc/nodemodel.tab → 각 노드의 machine 정의
/opt/xcat/etc/passwd.tab → 각 Management Node 의 Password 설정
/opt/xcat/etc/mac.tab → 각 노드의 Mac Address 설정
/opt/xcat/etc/conserver.cf → 각 노드의 conserver 설정
/opt/xcat/etc/conserver.tab → conserver의해 관리될 노드목록 설정
/opt/xcat/etc/mpa.tab → Management processor Adapter 및 module 정의

/opt/xcat/etc/mp.tab → Management processor Network의 토폴로지 정의

/opt/xcat/etc/bcnet.tab → blade서버 slot number와 blade switch번호 정의

/opt/xcat/etc/networks.tab → dhcp에서 정의하는 모든 네트워크를 정의

* /opt/xcat/etc/안에 파일 수정 후 setupxcat명령을 실행해야 변경된 설정이 반영됨

```
[root@mgt etc]# cat site.tab
sshkeyver                1
rsh                      /usr/bin/ssh
rcp                      /usr/bin/scp
gkhfile                  /opt/xcat/etc/gkh
tftpdir                  /tftpboot
tftpxcatroot             xcat
domain                   cluster.com
dnssearch                cluster.com
nameservers              192.168.12.74
forwarders               NA
nets                     192.168.12.0:255.255.255.0,192.168.70.0:255.255.255.0
dnsdir                   /var/named
dnsallowq                NA
domainaliasip            NA
mxhosts                  mailhosts.cluster.com
mailhosts                mgt
master                   mgt
homefs                   mgt:/home
localfs                  mgt:/usr/local
pbshome                  /var/spool/pbs
pbsprefix                /opt/pbs
pbsserver                mgt
scheduler                maui
xcatprefix               /opt/xcat
keyboard                 us
timezone                 US/Eastern
offutc                   -5
mapperhost               mgt
serialmac                0
serialbps                19200
snmpc                    public
snmpd                    192.168.12.74
poweralerts              Y
timeservers              mgt
logdays                  7
installdir               /install
clustername              wopr
dhcpver                  3
dhcpconf                 /etc/dhcpd.conf
dynamicr                 eth0,ia32,192.168.12.74,255.255.255.0,192.168.12.1,192.168.12.254
usernodes                NA
usermaster               mgt
nisdomain                NA
nismaster                NA
nisslaves                NA
homelinks                NA
chagemin                 0
chagemax                 60
chagewarn                10
```

```
    chageinactive           0
    mpciroot                /opt/xcat/lib/mpcli
    ipmimaxp                16
    ipmitimeout             30
    dpciroot                /opt/xcat/i686
    ipmitool                /opt/xcat/i686/sbin/ipmitool
    myrimask                255.255.0.0
    mpicli2root             /opt/xcat/lib/mpicli2
    ipmiretries             2
xcatdport                   3001
```

```
[root@mgt etc]# cat nodelist.tab
#node1   all,compute,opteron,rack1
blade1   all,compute,blade,box1
blade2   all,compute,blade,box1
blade3   all,compute,blade,box1
blade4   all,compute,blade,box1
sw1      sw

bc1      bc
```

```
[root@mgt etc]# cat noderes.tab
all     mgt,mgt,/install,1,N,N,N,N,N,N,N,eth0,eth0
```

```
[root@mgt etc]# cat nodetype.tab
blade1          rhas4,x86_64,blade
blade2          rhas4,x86_64,blade
blade3          rhas4,x86_64,blade
blade4          rhas4,x86_64,blade
```

```
[root@mgt etc]# cat nodehm.tab
#node1   openipmi,openipmi,NA,openipmi,NA,conserver,NA,openipmi,rcons,pxe,bcm5700,vnc,Y,openipmi,NA,9600

blade1   mp,mp,NA,mp,mp,conserver,mp,mp,bcmm,pxe,bcm5700,NA,N,mp,mp,NA → Blade Center용
blade2   mp,mp,NA,mp,mp,conserver,mp,mp,bcmm,pxe,bcm5700,NA,N,mp,mp,NA
blade3   mp,mp,NA,mp,mp,conserver,mp,mp,bcmm,pxe,bcm5700,NA,N,mp,mp,NA
blade4   mp,mp,NA,mp,mp,conserver,mp,mp,bcmm,pxe,bcm5700,NA,N,mp,mp,NA

node001  ipmi,ipmi,ipmi,ipmi,ipmi,conserver,NA,ipmi,switch,pxe,bcm5700,vnc,Y,ipmi,NA,19200→BMC용

node01 mp,mp,mp,mp,mp,conserver,rcons,mp,rcons,pxe,eepro100,vnc,N → RSA용

bc1      NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,N,NA,NA
```

```
[root@mgt etc]# cat nodemodel.tab
blade1          blade
blade2          blade
blade3          blade
blade4          blade
```

```
[root@mgt etc]# cat mac.tab
blade1-eth0     00:11:25:9E:A7:BA
blade1-eth1     00:11:25:9E:A7:BB
blade2-eth0     00:11:25:9E:A7:B0
blade2-eth1     00:11:25:9E:A7:B1
```

```
blade3-eth0       00:11:25:9E:A7:DE
blade3-eth1       00:11:25:9E:A7:DF
blade4-eth0       00:11:25:9E:AA:A2
blade4-eth1       00:11:25:9E:AA:A3
```

```
[root@mgt etc]# cat conserver.cf
LOGDIR=/var/log/consoles

blade1:|sol.bc blade1::&: → Blade Center용
blade2:|sol.bc blade2::&:
blade3:|sol.bc blade3::&:
blade4:|sol.bc blade4::&:

node001:|sol.e326 node001::&: → BMC용

node01:!ts01:3001:&: → RSA용

%%
trusted: 127.0.0.1
```

```
[root@mgt etc]# cat conserver.tab
# 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>
blade1              localhost,blade1
blade2              localhost,blade2
blade3              localhost,blade3
blade4              localhost,blade4


conserver.tab~   mpa.tab         nodehm.tab~    nodetype.tab     site.tab.ORG
[root@mgt etc]# cat mpa.tab
# mpa.tab
#
# This table lists the Management Processor Adapters and Modules.
#
# <type>        asma,rsa,rsa2,bc
# <name>        internal name (*)
# <number>      internal number,unique and >10000 (asma, rsa only)
# <command>     telnet,mpcli,http(hawk only) (bc must be http)
# <reset>       http(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>
#
 bc1        bc,bc1,NA,http,http,mpcli,NA,NA,NA
```

```
[root@mgt etc]# cat mpa.tab
# mpa.tab
#
# This table lists the Management Processor Adapters and Modules.
#
# <type>         asma,rsa,rsa2,bc
# <name>          internal name (*)
# <number>       internal number,unique and >10000 (asma, rsa only)
# <command>     telnet,mpcli,http(hawk only) (bc must be http)
# <reset>        http(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>
#
bc1        bc,bc1,NA,http,http,mpcli,NA,NA,NA → Blade Center용

bmc001    bmc,bmc001,NA,mpcli,mpcli,mpcli,NA,172.29.0.1,NA → BMC용

rsa01 rsa,rsa01,10001,mpcli,mpcli,NA,N,NA → RSA용
```

```
[root@mgt etc]# cat mp.tab
# 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>,<name>
#node10 rsa1,node10
#
# This table maps nodes to node chassis.
#
# <bc>         hostname of the node chassis management module.
# <bay>       position of the node.   DO NOT PAD WITH ZEROS.
#
```

```
#<node> <bc>,<bay>

blade1   bc1,1 → Blade Center용
blade2   bc1,2
blade3   bc1,3
blade4   bc1,4

node001 bmc001,node001 → BMC용

node01 rsa01,node01 → RSA용
```

```
[root@mgt etc]# cat bcnet.tab
blade1          sw1,1
blade2          sw1,2
blade3          sw1,3
blade4          sw1,4
```

```
[root@mgt etc]# cat networks.tab
# networks.tab
#
#network          mask,gateway,dns,dns,dns,...
192.168.12.0      255.255.255.0,NA,192.168.12.74
192.168.70.0      255.255.255.0,NA,192.168.12.74

169.254.0.0       255.255.0.0,NA,NA
```

2) blade서버 kickstart file 생성(kicstart template파일을 blade.tmpl이름으로 복사하여 사용한다.)

```
[root@mgt base]# cd /opt/xcat/install/rhas4/x86_64/base
[root@mgt base]# cp all.tmpl blade.tmpl
```

3) makedns -- site.tab 및 host파일을 참조하여 named.conf파일 생성

```
[root@mgt xcat]# makedns
Stopping named: [   OK   ]
named: no process killed
Starting named: [   OK   ]
```

4) makedhcp – site.tab,networks.tab,hosts파일을 참조하여dhcp.conf파일 생성

```
[root@mgt xcat]# makedhcp -new
updating existing /opt/xcat/etc/networks.tab
skipping 172.30.0.0 (entry exists)
skipping 172.29.0.0 (entry exists)
skipping 169.254.0.0 (entry exists)
skipping 172.20.0.0 (entry exists)
Saving orginal /etc/dhcpd.conf as /etc/dhcpd.conf.ORIG
added network eth0
added network eth1
updated /etc/sysconfig/dhcpd with DHCPD_INTERFACE or DHCPDARGS="eth0 eth1"
added network all
added subnet 172.20.0.0/255.255.0.0
added subnet 172.29.0.0/255.255.0.0
added subnet 172.30.0.0/255.255.0.0
added subnet 169.254.0.0/255.255.0.0
adding dynamic eth0
Shutting down dhcpd: [   OK   ]
Starting dhcpd: [   OK   ]
```

5) mkstage -- stage1

```
[root@mgt stage]# cd /opt/xcat/stage/
[root@mgt stage]# ./mkstage
- mkstage명령으로 /tftpboot/xcat디렉토리안에 pxelinux으로 os를 설치하기 위한 kernel및 ramdisk image, 설정파일들이 만들어짐
```

6) getmacs <nodereange> -- stage2

* blade서버의 mac address를 수집하여 mac.tab에 추가한다.

```
[root@mgt stage]# getmacs box1
Please reset nodes:  blade1 blade2 blade3 blade4 blade5 blade6 blade7 blade8 blade9
blade10 blade11 blade12 blade13 blade14

Press [Enter] when ready...
Saving output to mac-16754.lst in current directory /opt/xcat/stage.

blade1-eth0      00:14:5E:D5:D0:46
blade1-eth1      00:14:5E:D5:D0:48
blade10-eth0     00:14:5E:D5:E3:54
blade10-eth1     00:14:5E:D5:E3:56
blade11-eth0     00:14:5E:D5:E4:F2
blade11-eth1     00:14:5E:D5:E4:F4
blade12-eth0     00:14:5E:D5:CF:86
blade12-eth1     00:14:5E:D5:CF:88
blade13-eth0     00:14:5E:D5:D3:2E
blade13-eth1     00:14:5E:D5:D3:30
blade14-eth0     00:14:5E:D5:E2:28
blade14-eth1     00:14:5E:D5:E2:2A
blade2-eth0      00:14:5E:D5:DB:86
blade2-eth1      00:14:5E:D5:DB:88
blade3-eth0      00:14:5E:D5:D9:E8
blade3-eth1      00:14:5E:D5:D9:EA
blade4-eth0      00:14:5E:D5:DB:E6
blade4-eth1      00:14:5E:D5:DB:E8
blade5-eth0      00:14:5E:D5:D3:B2
blade5-eth1      00:14:5E:D5:D3:B4
blade6-eth0      00:14:5E:D5:E4:08
blade6-eth1      00:14:5E:D5:E4:0A

Auto merge mac-16754.lst with /opt/xcat/etc/mac.tab(y/n)? y
```

7) makedhcp --allmac

* 이전에 만들었던 dhcpd.conf파일에 각 host의 mac address를 추가하여 재 생성

```
[root@mgt stage]# makedhcp --allmac
updating existing /opt/xcat/etc/networks.tab
skipping 172.30.0.0 (entry exists)
skipping 172.29.0.0 (entry exists)
skipping 169.254.0.0 (entry exists)
skipping 172.20.0.0 (entry exists)
updated /etc/sysconfig/dhcpd with DHCPD_INTERFACE or DHCPDARGS="eth0 eth1"
```

8) mpacheck -l <mm range>(BladeCenter mm 체크)

```
[root@mgt stage]# mpacheck -l bc1
bc1: Host IP Address: 172.30.101.5
bc1: Subnet mask: 255.255.0.0
```

```
bc1: Gateway IP Address: 0.0.0.0
bc1: Data Rate: Auto
bc1: Duplex Mode: Auto
bc1: SNMP Traps: Enabled
bc1: SNMP Agent: Enabled
bc1: SNMP Community Name: public
bc1: SNMP IP Address 1: 0.0.0.0
```

9) mpname <noderange> -- stage3

* 각 blade들의 hardeware management processor를 rename 한다.

```
[root@mgt stage]# mpname box1
blade1: blade bc1,1 renamed from blade1 to blade1
blade10: blade bc1,10 renamed from blade10 to blade10
blade11: blade bc1,11 renamed from blade11 to blade11
blade12: blade bc1,12 renamed from blade12 to blade12
blade13: blade bc1,13 renamed from blade13 to blade13
blade14: blade bc1,14 renamed from blade14 to blade14
blade2: blade bc1,2 renamed from blade2 to blade2
blade3: blade bc1,3 renamed from blade3 to blade3
blade4: blade bc1,4 renamed from blade4 to blade4
blade5: blade bc1,5 renamed from blade5 to blade5
blade6: blade bc1,6 renamed from blade6 to blade6
blade7: blade bc1,7 renamed from blade7 to blade7
blade8: blade bc1,8 renamed from blade8 to blade8
blade9: blade bc1,9 renamed from blade9 to blade9
```

10) mpascan <mm range>

* 모든 blade서버 이름들이 설정tab에 정의되었는지를 확인

```
[root@mgt stage]# mpascan bc1
bc1: blade1          1
bc1: blade10         10
bc1: blade11         11
bc1: blade12         12
bc1: blade13         13
bc1: blade14         14
bc1: blade2          2
bc1: blade3          3
bc1: blade4          4
bc1: blade5          5
bc1: blade6          6
bc1: blade7          7
bc1: blade8          8
bc1: blade9          9
```

11) RHEL AS4 CD 내용을 Management node에 복사한다.

```
[root@mgt etc]# copycds
Please insert CD 1 and press [Enter]...
- 자동적으로 /install/rhas4/x86_64 디렉토리 생성과 함께 CD내용을 디렉토리에 복사한다.
```

12) post 디렉토리 복사

```
[root@mgt /]# cd /opt/xcat
[root@mgt xcat]# find post -print | cpio -dump /install
```

13) management node ssh key생성

```
[root@mgt xcat]# gensshkeys root
[root@mgt xcat]# cp /root/.ssh/* /install/post/.ssh/ -f
- * management node 의 ssh - key 를 생성하여 /install 에 놓고 각 계산 노드들이 설치를 할때에
~/.ssh/디렉토리안에 복사를 해서 사용할 수 있도록 함
```

14) conserver 데몬 시작

```
[root@mgt /]# service conserver start
Starting Console Server: [  OK  ]
```

15) NFS 서버 설정

```
[root@mgt /]# vi /etc/exports
/opt/xcat *(ro,no_root_squash,sync)
/install *(ro,no_root_squash,sync)
/tftpboot *(ro,no_root_squash,sync)
[root@mgt /]# chkconfig nfs on
[root@mgt /]# service nfs start
Starting NFS services:  [  OK  ]
Starting NFS quotas: [  OK  ]
Starting NFS daemon: [  OK  ]
Starting NFS mountd: [  OK  ]
[root@mgt /]# exportfs
/opt/xcat       <world>
/tftpboot       <world>
/install        <world>
```

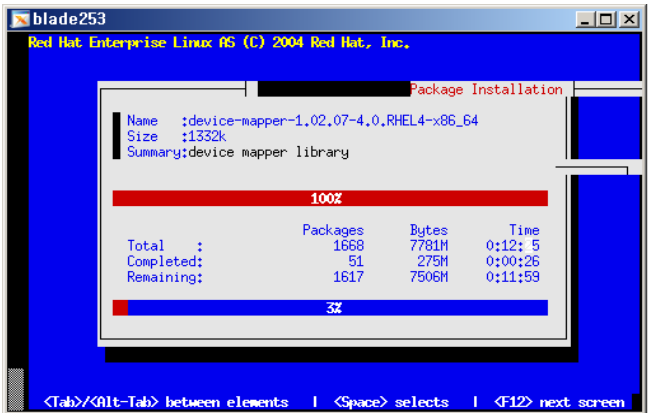
16) rinstall <node range>

* /tftpboot/pxelinux.cfg/안에 node이름별로 pxelinux관련 파일들이 생성되면서 node설치가 진행된다.

```
[root@mgt /]# rinstall blade253
blade253: install rhas4-x86_64-blade-all
blade253:  ********
[root@mgt pxelinux.cfg]# ls
AC14       AC147709  AC14770D  AC147C02  AC147C06  blade256
AC140A08   AC14770A  AC14770E  AC147C03  AC147C07  blade243
AC14760A   AC14770B  AC147908  AC147C04  AC147C08  blade254
AC147708   AC14770C  AC147C01  AC147C05  AC147C09  blade255 default
```

17) tty console를 사용하여 node설치 확인

[root@mgt ~]# wcons -t blade253 (console open)



[root@mgt ~]# wkill blade253 (console close)

- console font 및 size조절

* ctrol key + mouse right button 클릭

18) makesshgkh <noderange>

* 설치 완료 후 node들의 ssh key값을 스캔하여 정합성을 체크한다.

```
[root@mgt ~]# makesshgkh blade1-blade14
Scanning keys, please wait...
```

1.10 xCAT 중요 명령어

xCAT 관련한 모든 명령어들은 /opt/xcat/sbin 또는 /opt/xcat/bin 디렉토리 안에 정의되어 있다. 하지만 몇몇 명령어들은 xCAT 스크립트에서만 사용되며, 그 외 다른 명령어들은 클러스터 환경을 관리하는 담당자나 클러스터를 이용하여 Job을 수행하고자 하는 User에 의해 사용될 수 있다. 여기서는 주로 사용되는 명령어에 대해 설명하고자 한다.

1. rpower - Remote power control

xCAT의 설정파일인 nodelist.tab 파일에 정의되어 있는 node들의 전원관리를 원격에서 On/Off/Reset 할 수 있도록 함

Synopsis	rpower [noderange] [on|off|stat|state|reset|boot|cycle]	
Options	on	Turn power on
	off	Turn power off
	stat | state	Return the current power state
	reset	Send a hardware reset
	boot	If off, then power on. If on, then hard reset
	cycle	Power off, then on
Examples	[root@mgt ~]# rpower blade1-blade14 stat ➔ blade1~14까지 전원상태 확인 [root@mgt ~]# rpower blade7 on ➔ blade7 서버 전원을 On 시킴 [root@mgt ~]# rpower box22 reset ➔ Chassis 22번 전 서버의 전원을 Reboot 시킴 [root@mgt ~]# rpower all off ➔ 전 시스템의 전원을 Off 시킴	

2. rinstall - Remote network install

xCAT의 설정파일인 nodetype.tab 파일에 정의되어 있는 사항에 맞게 node들의 설치를 원격에서 진행 시킴. (단, 설치 진행되는 화면은 확인할 수 없으며, 진행사항을 보고자 할 경우 console server를 통해서 볼 수 있음.)

Synopsis	rinstall [noderange]
Examples	[root@mgt ~]# rinstall blade1-blade14 ➔ blade1~14까지 원격 설치 진행 [root@mgt ~]# rinstall box10 ➔ Chassis 11번 전 서버의 원격 설치 진행

3. wcons - Windowed remote console

xCAT의 설정파일인 conserver.tab 파일에 정의되어 있는 node들의 Serial 연결 화면을 제공하여 줌.

Synopsis	rpower [noderange] [on|off|stat|state|reset|boot|cycle]	
Options	-t|-tile|--tile n	창의 왼쪽 위에서부터 왼쪽에서 오른쪽으로 해당 node들의 xWindow 창을 보여준다. 'n' 은 한 행에서 보여줄 수 있는 tile의 수를 정의함.
	-f|-font|--font font	wcons 화면들의 font를 지정함. (주로 화면창에서 ctrl + right click으로 조절이 가능하다.) wcons가 지원하는 font alias는 아래와 같다. verysmall | vs = nil2 small|s = 5x8 medium | med|m = 6x13

		large | l | big | b = 7x13 verylarge | vl | verybig | vb = 10x20
Examples	[root@mgt ~]# wcons blade7 ➔ blade7의 Serial Console화면을 보여줌. [root@mgt ~]# wcons -t 3 box1➔ Chassis 1번의 Serial Console화면을 3개 단위로 보여줌.  * 화면 크기 조절 (wcons 화면에서 ctrl + mouse right click) 	

4. wkill - Windowed remote console kill

wcons 명령어를 통해 띄운 창을 없애고자 할 때 사용.

Synopsis	wkill [noderange]
Examples	[root@mgt ~]# wkill blade1-blade14 ➔ blade1~14의 Console 화면을 없앤다. [root@mgt ~]# wkill box10 ➔ Chassis 10번에 해당하는 Console 화면을 없앤다.

5. winstall - Windowed remote network install

xCAT의 설정파일인 nodetype.tab 파일에 정의되어 있는 사항에 맞게 node들의 설치를 원격에서 진행 시키며, winstall 명령어 실행 시 wcons 명령어를 동시에 실행시켜 설치 화면을 Serial Console화면을 통해서 확인 가능케 함.

Synopsis	winstall [{-t|-tile|--tile} n | {-t|-tile|--tile}=n] [{-f|-font|--font} font | {-f|-font|-

	-font}=font] noderange	
Options	-t|-tile|--tile n	창의 왼쪽 위에서부터 왼쪽에서 오른쪽으로 해당 node들의 xWindow 창을 보여준다. 'n' 은 한 행에서 보여줄 수 있는 tile의 수를 정의함.
	-f|-font|--font font	wcons 화면들의 font를 지정함. (주로 화면창에서 ctrl + right click으로 조절이 가능하다.) wcons가 지원하는 font alias는 아래와 같다. verysmall | vs = nil2 small|s = 5x8 medium | med|m = 6x13 large | l | big | b = 7x13 verylarge | vl | verybig | vb = 10x20
Examples	[root@mgt ~]# winstall blade1-blade14 ➔ blade1~14까지 원격 설치를 진행하고 Serial Console 화면을 띄워줌 [root@mgt ~]# winstall box22 ➔ Chassis 22번 전 서버에 대한 설치를 진행하고 Serial Console 화면을 띄워줌	

6. psh - Parallel remote shell

Cluster User들이 가장 많이 사용하는 명령어로, xCAT에 설정되어 있는 node들에 동시에 동일한 명령어를 실행하고자 할 때 사용 (rsh를 사용할지 ssh를 사용할지는 site.tab 파일에 정의되어있음)

Synopsis	psh [-s] [noderange|me|pbs job id] [command]	
Options	-s	Issues the commands serially.
Examples	[root@mgt ~]# psh box1-box5 ls ➔ Chassis 1~5번에 해당되는 서버들에게 동시에 ls명령실행 [root@mgt ~]# psh blade3,blade5 date ➔ Blade 3,5번 서버에게 동시에 date 명령실행	

7. pping - Parallel ping

xCAT에 설정되어 있는 node들에 동시에 ping 명령어를 실행하고자 할 때 사용

Synopsis	pping [-s] [noderange]	
Options	-s	Issues the commands serially.
Examples	[root@mgt ~]# pping blade1-blade20 ➔ Blade 1~20번에 해당되는 서버에 ping명령 동시실행 [root@mgt ~]# pping box25 ➔ Chassis 25번 서버에 ping명령 동시실행	

8. prcp - Parallel remote copy

Cluster User들이 가장 많이 사용하는 명령어로, xCAT에 설정되어 있는 node들에 동시에 동일한 명령어를 실행하고자 할 때 사용 (rsh를 사용할지 ssh를 사용할지는 site.tab 파일에 정의되어있음)

9. rinv - Remote hardware inventory

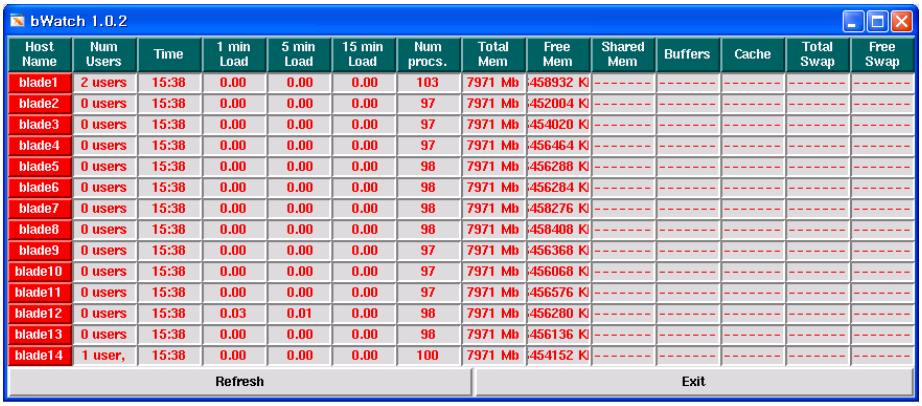
Onboard에 있는 관리 프로세스를 통해 하드웨어 구성 정보를 확인.

Synopsis	rinv [noderange] [pci|config|model|serial|asset|vpd|bios|mprom|all]	
Options	pci	PCI bus 정보 추출.
	config	Processor 수량, speed, total memory, DIMM locations 정보 추출
	model	model number 추출.
	serial	serial number 추출.
	asset	Eth0 MAC address 추출
	vpd	BIOS level, management processor, firmware level 추출

	bios	BIOS level 정보 추출.
	mprom	management processor firmware level 정보 추출
	all	위의 모든 정보 추출
Examples	[root@mgt ~]# rinv blade5 bios ➔ blade5의 BIOS 정보 추출 [root@mgt ~]# rinv box23 all ➔ Chassis 23번의 모든 하드웨어 구성 정보 추출	

10. bwatch – Blade Watch

각 Blade Server에 대한 간단한 시스템 사용 정보를 GUI화면을 통해 확인

Synopsis	bwatch [noderange]
Examples	[root@mgt ~]# bwatch box1 ➔ Chassis 1번에 해당되는 서버들의 시스템 정보 확인  [root@mgt ~]# bwatch blade3 ➔ Blade 3번 서버의 시스템 정보 확인

bWatch 1.0.2

Host Name	Num Users	Time	1 min Load	5 min Load	15 min Load	Num procs.	Total Mem	Free Mem	Shared Mem	Buffers	Cache	Total Swap	Free Swap
blade1	2 users	15:38	0.00	0.00	0.00	103	7971 Mb	458932 Kb	-------	-------	-------	-------	-------
blade2	0 users	15:38	0.00	0.00	0.00	97	7971 Mb	452004 Kb	-------	-------	-------	-------	-------
blade3	0 users	15:38	0.00	0.00	0.00	97	7971 Mb	454020 Kb	-------	-------	-------	-------	-------
blade4	0 users	15:38	0.00	0.00	0.00	97	7971 Mb	456464 Kb	-------	-------	-------	-------	-------
blade5	0 users	15:38	0.00	0.00	0.00	98	7971 Mb	456288 Kb	-------	-------	-------	-------	-------
blade6	0 users	15:38	0.00	0.00	0.00	98	7971 Mb	456284 Kb	-------	-------	-------	-------	-------
blade7	0 users	15:38	0.00	0.00	0.00	98	7971 Mb	458276 Kb	-------	-------	-------	-------	-------
blade8	0 users	15:38	0.00	0.00	0.00	98	7971 Mb	458408 Kb	-------	-------	-------	-------	-------
blade9	0 users	15:38	0.00	0.00	0.00	97	7971 Mb	456368 Kb	-------	-------	-------	-------	-------
blade10	0 users	15:38	0.00	0.00	0.00	97	7971 Mb	456068 Kb	-------	-------	-------	-------	-------
blade11	0 users	15:38	0.00	0.00	0.00	97	7971 Mb	456576 Kb	-------	-------	-------	-------	-------
blade12	0 users	15:38	0.03	0.01	0.00	98	7971 Mb	456280 Kb	-------	-------	-------	-------	-------
blade13	0 users	15:38	0.00	0.00	0.00	98	7971 Mb	456136 Kb	-------	-------	-------	-------	-------
blade14	1 user,	15:38	0.00	0.00	0.00	100	7971 Mb	454152 Kb	-------	-------	-------	-------	-------

Refresh Exit

11. rbootseq – Remote boot sequence

BIOS상에서 boot sequence를 변경하지 않고 명령어를 통해 boot priority를 변경.

Synopsis	Rbootseq [noderange] [{f|floppy,c|cdrom,n|network,hd{0,1}|harddisk{0,1}}|list]	
Options	f|floppy	Floppy disk
	c|cdrom	CD-Rom
	n|network	PXE Network
	hd{0,1}|harddisk{0,1}	Hard Disk.
	list	현재 boot sequence 확인
Examples	[root@mgt ~]# rbootseq box1 n,c,hd0,f ➔ Chassis 1번의 Boot Sequence를 네트워크, CD-Rom, Hard Disk 0번, Floppy Disk으로 변경 [root@mgt ~]# rbootseq box23 list➔ Chassis 23번의 모든 서버들의 Boot Sequence 확인	

2. HPC Software Stack Installation Guide

2.1 Intel C Compiler

intel CD를 삽입하고 CD에 있는 바이너리를 특정 디렉토리로 복사한다.
tar 압축을 푼후 인스톨 프로그램을 실행시킨다.

```
[root@dmgt tmp]# tar xfz l_cc_c_9.1.045.tar.gz
[root@dmgt tmp]# cd l_cc_c_9.1.045
[root@dmgt l_cc_c_9.1.045]# ./install.sh
```

install 메뉴에서 1번을 선택한다음, 다시 한번 1번을 선택하고, 라이센스파일의 위치를 적어준다.

```
Please provide the license file name with full path (/<path>/<name>.lic)
        b. Go Back.
        x. Exit.

License file path  :   /opt/intel/cpplicense.lic
```

이후 accept를 입력하면 컴파일러가 설치된다.
설치를 마친후, /opt/intel/cc라는 디렉토리가 설치되었는지 확인한 다음 아래의 명령어로 컴파일러가 정상적으로 설치되었는지 확인한다.

```
[root@dmgt l_cc_c_9.1.045]source /opt/intel/cce/9.1.042/bin/iccvars.sh
[root@blade329 ~]# icc
icc: Command line error: no files specified; for help type "icc -help
```

2.2 Intel Fortran Compiler

intel CD를 삽입하고 CD에 있는 바이너리를 특정 디렉토리로 복사한다.
tar 압축을 푼후 인스톨 프로그램을 실행시킨다.

```
[root@dmgt tmp]# tar xfz l_fc_c_9.1.040.tar.gz
[root@dmgt tmp]# cd l_fc_c_9.1.040
[root@dmgt l_fc_c_9.1.040]# ./install.sh
```

install 메뉴에서 1번을 선택한다음, 다시 한번 1번을 선택하고, 라이센스파일의 위치를 적어준다.

```
Please provide the license file name with full path (/<path>/<name>.lic)
      b. Go Back.
      x. Exit.

License file path  :   /opt/intel/fclicense.lic
```

이후 accept를 입력하면 컴파일러가 설치된다.
설치를 마친후, /opt/intel/fc라는 디렉토리가 설치되었는지 확인한 다음 아래의 명령어로 컴파일러가 정상적으로 설치되었는지 확인한다.

```
[root@dmgt tmp]source / source /opt/intel/cce/9.1.042/bin/iccvars.sh
[root@ dmgt tmp]# ifc
ifc: warning: The Intel Fortran driver is now named ifort.  You can suppress this message with '-quiet'
Ifort: Command line error: no files specified; for help type "ifort -help"
```

2.3 Intel MPI

intel CD를 삽입하고 CD에 있는 바이너리를 특정 디렉토리로 복사한다.
tar 압축을 푼후 인스톨 프로그램을 실행시킨다.

```
[root@dmgt tmp]# tar xfz l_mpi_p_3.0.033.tar.gz
[root@dmgt tmp]# cd l_mpi_p_3.0.033
[root@dmgt l_mpi_p_3.0.033]# ./install.sh
```

install 메뉴에서 1번을 선택한다음, 다시 한번 1번을 선택하고, 라이센스파일의 위치를 적어준다.

```
Please provide the license file name with full path (/<path>/<name>.lic)
      b. Go Back.
      x. Exit.

License file path  :    /opt/intel/mpilicense.lic
```

이후 accept를 입력하면 mpi가 설치된다.
설치를 마친후, /opt/intel/mpirun라는 디렉토리가 설치되었는지 확인한 다음 아래의 명령어로 mpi가 정상적으로 설치되었는지 확인한다.

```
[root@dmgt tmp]source source /opt/intel/mpi/3.0/bin64/mpivars.sh
[root@blade329 bin]#   mpiexec | more
usage: mpiexec [-h or -help or --help]      # get this message
```

2.4 Intel MKL (Math Kernel Library)

intel CD를 삽입하고 CD에 있는 바이너리를 특정 디렉토리로 복사한다.
tar 압축을 푼후 인스톨 프로그램을 실행시킨다.

```
[root@dmgt tmp]# tar xfz l_mkl_p_9.0.018.tar.gz
[root@dmgt tmp]# cd l_mkl_p_9.0.018
[root@dmgt l_mkl_p_9.0.018]# ./install.sh
```

install 메뉴에서 1번을 선택한다음, 다시 한번 1번을 선택하고, 라이센스파일의 위치를 적어준다.

```
Please provide the license file name with full path (/<path>/<name>.lic)
      b. Go Back.
      x. Exit.

License file path  :    /opt/intel/mkllicense.lic
```

이후 accept를 입력하면 MKL이 설치된다.
설치를 마친후 /opt/intel/mkl이 생성되었나를 확인한다.

2.5 PBS Pro

설치를 원하고자하는 노드에 dmgt의 /root/pspace/PBSPro_7.1를 복사해 준다.

```
root@dmgt#> scp /root/pspace/PBSPRo_7.1 blade1:/tmp
```

복사한 노드에 로그인한후 복사한 디렉토리로 이동한다.

```
root@dmgt#> ssh blade1
```

복사된 디렉토리로 이동한후 설치 명령을 수행한다.

```
root@blade1# PBSPro_7.1> ./INSTALL < install_mom
```

같은 디렉토리의 pbs.conf(디버그 노드일 경우 pbs_debug.conf)를 /etc/아래로 복사한후 pbs restart시켜준다.

```
root@blade1# PBSPro_7.1> cp pbs.conf /etc/pbs.conf
root@blade1# PBSPro_7.1> service pbs restart
```

2.6 PBS Pro Check

Pbs server와의 통신확인

```
[root@blade328 scripts]# qstat -Bf
Server: schedule1
    server_state = Active
    server_host = schedule1
    scheduling = True
    total_jobs = 1
```

```
     state_count = Transit:0 Queued:0 Held:1 Waiting:0 Running:0 Exiting:0 Begun
          :0
     acl_roots = root@schedule2,root@schedule1
     default_queue = workq
     log_events = 511
     mail_from = adm
     query_other_jobs = True
     resources_default.ncpus = 1
     default_chunk.ncpus = 1
     scheduler_iteration = 600
     FLicenses = 0
     resv_enable = True
     node_fail_requeue = 310
     max_array_size = 10000
     pbs_version = PBSPro_7.1.4.63140
```

pbs 데몬 상태확인

```
[root@blade328 scripts]# pbsnodes blade328
blade328
      Host = blade328
      ntype = PBS
      state = free
      license = l
      pcpus = 4
      resources_available.arch = linux
      resources_available.host = blade328
      resources_available.mem = 8162496kb
      resources_available.ncpus = 4
      resources_assigned.mem = 0kb
      resources_assigned.ncpus = 0
      resv_enable = True
```

3. HPL Compile, Installation and Running BMT

3.1 HPC 시스템의 성능수치 - FLOPS

슈퍼컴퓨터에서의 성능은 FLOPS(Floationg-point Operations Per Second : 초당 실수연산 회수) 1초에 덧셈, 뺄셈, 곱셉, 나눗셈 등의 실수 계산을 총 몇 번 할 수 있는지를 나타내는 값이다. 만약 1초에 실수끼리 곱셈을 2번씩 할 수 있다면 그때의 계산 속도는 2FLOPS가 되는 것이다.

3.2 이론성능 vs 실제성능 Rpeak vs Rmax

그럼 이제 성능에 대해 알아봤으니 어떻게 성능을 측정하는가?에 대한 해답이 나와야 한다. 성능은 크게 이론성능과 실제성능으로 나뉜다. 이론성능(Rpeak)은 말 그대로 이론적인 성능이고 실제성능(Rmax)은 여러가지 테스트를 통해 얻어낸 실제로 측정된 성능이다. 그럼 실제성능을 구하기 전에 내 하드웨어(CPU)가 낼 수 있는 이론적인 성능을 계산해 보도록 하자. KIST 슈퍼컴퓨팅센터의 QnA에 올라온 것을 인용하자면, 이론 성능치, peak performance(Rpeak)는 아래의 식을 통해 구할 수 있다.

Flop/s=(cycle/s)*(Flop/cycle)*(Number of pipes)

Pentium-IV 3GHz일 경우 ia32(Itaium을 제외한 intel, AMD의 대부분의 CPU) 아키텍처 계열의 CPU로서 1 cycle당 (double precision) floating point 연산은 최대 2번 수행할 수 있으며, vector 계열이 아니므로 number of pipes는 1이다. 따라서 위식을 적용하면

3(GHz/sec) * 2(Flop/Hz) * 1 = 6 GFlops 따라서 위와 같은 노드가 32개 일 경우 32*6 = 192 Gflops 가 나온다.

CPU가 Woodcrest 모델일 때는 3(GHz/sec) * 4(Flop/Hz) * 1 = 12 GFlops로 계산이 된다. 마이크로 프로세스의 아키텍쳐 변화로 인하여 한 클럭당 4개의 fp명령어를 처리 가능하게 되었다.

1. 명령어(instruction, mictor-op code)를 1Clock당 3개에서 4개 fetch, decode, issue, execution & retire 2. 과거에는 fp 연산시 add or mul/div 였는데 지금은 add + mul/div 인구조로변경되었습니다. 즉 과거에는 add 와 mul/div 를 동시에 처리할 수 없었는데(둘중에하나만처리), 지금은 add와 mul/div를 동시에 처리 가능합니다. 따라서 행렬연산 과 같은 HPC application에서 4개의 fp 처리가 가능합니다.

3.3 HPL (High Performance Linpack)

흔히 클러스터 시스템에서 벤치마크는 LINPACK 을 사용하거나 HPL(High-Performance Linpack Benchmark)을 통해서 시스템의 실제 성능을 측정하게 된다. LINPACK 벤치마크 에서 중점적으로 사용되는 루틴들은 Gauss 소거법을 이용한 N 개의 선형방정식 의 해를 구하는 것으로 BLAS (Basic Linear Algebra Subprograms) 에 포함되어 있다. BLAS 는 LINPACK 벤치마크 에서 가장 기본이 되는 라이브러리로써 기본적인 선형대수 연산함수 들을 구현해놓은 집합이다. 이 BLAS 를 이용해 벤치마킹을 할수도 있지만 ATLAS (Automatically Tuned Linear Algebra Software) 를 이용하여 해당 플랫폼에 최적화된 루틴 라이브러리를 생성 할 수도 있다. HPL 을 사용하기 위해서는 MPI implementation 과 BLAS implementation 을 필요로 한다.

3.4 선형대수 BLAS , ATLAS

BLAS (Basic Linear Algebra Subprograms)

선형대수(Linear Algebra) 문제의 해를 효율적으로 구하기 위한 방법의 하나는 Basic Linear Algebra Subprograms(BLAS)를 이용하는 것이다. BLAS 는 blocking 기법을 바탕으로 하여 기본적인 vector 와 matrix 연산을 수행하는 역할을 한다. BLAS 에는 연산의 종류에 따라 Level 1, 2, 3 BLAS 로 나뉘어진다. Level 1 BLAS 는 vector-vector 연산을 수행하고, Level 2 BLAS 는 matrix-vector 연산, Level 3 BLAS 는 matrix-matrix 연산을 하는데 사용되어진다. BLAS 는 뛰어난 효율과 우수한 이식성을 바탕으로 LAPACK 과 같은 Linear Algebra Software 의 개발에 사용되고 있다. 현재 각각의 architecture 에 hand-optimized BLAS 가 software vendor 들에 의해서 만들어져 있다. 그러나 새로운 architecture 마다 최적화된 BLAS 를 만드는 작업은 시간이 많이 걸릴 뿐만 아니라 금전적인 문제도 가지고 있 다. 이로 인해 Pentium/Linux architecture 를 위한 효율적인 matrix multiply 를 할 수 있는 BLAS 가 아직 만들어지지 않았다. 따라서 오늘날의 다양한 Architecture 에서 효율적으로 사용될 수 있는 basic linear algebra routines 를 만들어주는 software 가 필요하게 되었다. 이러한 목적을 달성하기 위하여 만든 소프트웨어가 바로 ATLAS(Automatically Tuned Linear Algebra Software)이다.

ATLAS (Automatically Tuned Linear Algebra Software)

ATLAS 는 그 이름에서도 알 수 있듯이 오늘날의 microprocessor 에서 고도의 효율성을 달성할 수 있는 basic linear algebra routine 을 스스로 만들 수 있는 software 이다. 따라서 프로그램의 특성에 따라, 강한 이식성과 architecture 의 특성에 맞는 BLAS implementation 을 제공하고 있다. 지금도 꾸준히 새로운 버전의 ATLAS 가 만들 어지고 있으며, C 와 Fortran77 interface 를 제공하고 있다. 현재 안정버전은 3.2.1 까지, 개발자버전은 3.3.1 까지 만들어져 있다. ATLAS 3.2.1 에서는 Intel 의 SSE 와 AMD 의 3DNOW 를 이용하여 library 를 만들고 있으며, ATLAS 3.3.1 에서는 Intel 의 SSE, SSE2 와 AMD 의 3DNOW, 3DNOW2 를 각각 이용하도록 되어있다. 따라서 ATLAS 3.2.1 을 사용한 Pentium III 에서는 단정도 부동소수점(single precision floating point) 연산의 성능이 확연히 좋아지고, ATLAS 3.3.1 을 사용한 Pentium4 시스템에서 SSE2 기능을 탐지하여 사용함으로서 배정도 부동소수점(double precision floating point) 연산에서의 성능이 획기적으로 개선되는 것을 확인할 수 있다.

3.5 MPI (Message Passing Interface)

MPI 의 종류에는 여러 가지가 있지만, Linux 용은 크게 두가지 이다. 그 하나가 LAM/MPI 이며, 또 다른 하나는 MPICH 이다. 이러한 MPI Library 의 설치는 GNU Compiler 를 설치할 때 와 같은 configure , make, make install 의 일련의 과정을 거치며, 다른 점은 configure 할 때의 option 의 차이이다. 표 1 은 사용되어진 compiler 그리고 사용되어진 option 에 따라 설치되어진 CUP 에 대한 MPI 를 나타내었다.

MPI	OPTION	Compiler	비고
lam-gcc (lam-6.5.6)	--prefix=/usr/local/lam-gcc	gcc-2.95.3	Smp machine: --with-

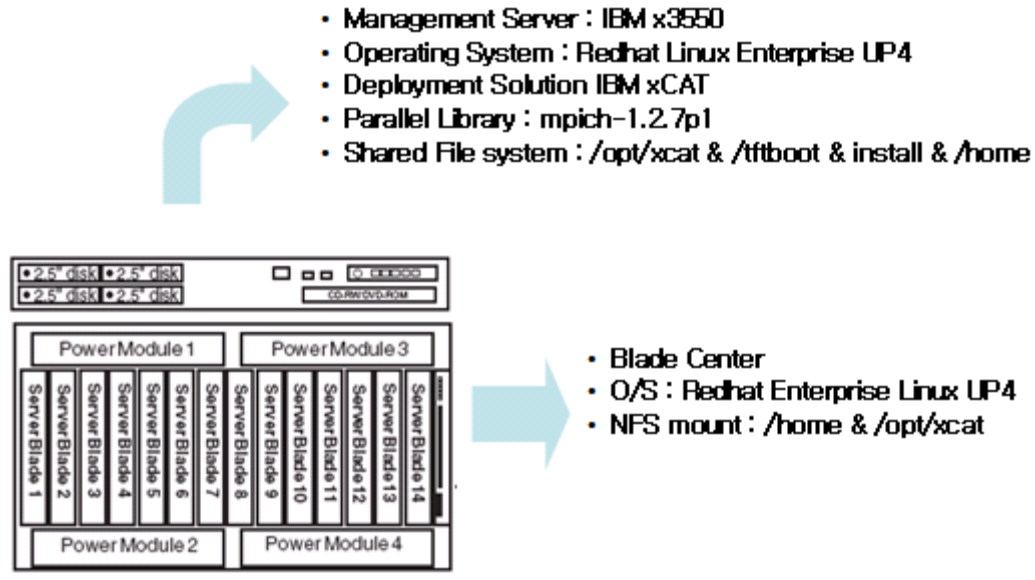
lam-intel (lam-6.5.6)	--prefix=/usr/local/lam --with-cc=icc --with-fc=ifc --with-cflags='-unroll -axW -tpp7 -align' --with-fflags='-unroll -axW -tpp7 -align'	Intel Compiler	rpi=usysv 추가
mpich-gcc (mpich-1.2.4)	--prefix=/usr/local/mpich-gcc	gcc-2.95.3	
mpich-intel (mpich-1.2.4)	export CC=icc export FC=ifc export F90=ifc --prefix=/usr/local/mpich-intel –with-comm=shared	Intel Compiler	

표 1 MPI Library

3.6 MPICH and Library Installation

3.6.1 사전 시스템 구성 환경

시스템 운영환경은 x86_64를 기반으로 하는 리눅스 시스템으로 MPI용 네트웍은 기가비트 이더넷을 사용하여 운영체제는 Redhat Enterprise Linux Update 4를 이용하여 시스템을 구성하였다. 시스템 운영환경에 대한 구성은 다음과 같다.



3.6.2 MPICH Installation

GNU Compiler인 GCC 컴파일러에 대한 환경변수 및 설치위치 Path등에 대한 환경변수가 적절하게 구성되어 있는지 확인하여 MPICH 에 대한 설치를 진행한다

```
# gcc -v

Reading specs from /usr/lib/gcc/x86_64-redhat-linux/3.4.6/specs
Configured with: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --
```

```
enable-shared --enable-threads=posix --disable-checking --with-system-zlib --enable-
__cxa_atexit --disable-libunwind-exceptions --enable-java-awt=gtk --host=x86_64-redhat-linux
Thread model: posix
gcc version 3.4.6 20060404 (Red Hat 3.4.6-3)
```

* 컴파일러에 대한 기본환경 구성 및 환경변수 항목들이 정상적으로 확인되었으면 아래와 같이 MPICH 병렬 라이브러리에 대한 기본환경 변수를 구성해 준다

```
# vi ~/.bashrc
.....
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

export MPICH="/usr/local/mpich/1.2.7p1/ip/x86_64/smp/gnu64/ssh"
export MPICH_PATH="${MPICH}/bin"
export MPICH_LIB="${MPICH}/lib"
export PATH="${MPICH_PATH}:${PATH}"
export LD_LIBRARY_PATH="${MPICH_LIB}:${LD_LIBRARY_PATH}"
```

MPICH 병렬라이브러리에 대한 기본환경 구성

* xCAT 기본설치 디렉토리인 /opt/xcat/build/mpi에 mpich tar를 파일을 복사하여 압축을 해제 (ex: NFS로 공유된 /opt/xcat/build/mpi에 mpimaker 명령어를 이용, 환경변수 자동구성 가능)

```
# source ~/.bash_profile [환경 프로파일 적용]
```

다운로드 받은 mpich소스의 압축을 풀고, 해당 소스의 루트로 이동합니다. 아래서 심볼릭 링크를 사용하는 이유는, 혹 다른 버전의 mpich로 변경하더라도 링크만 변경해주면, 사용자들의 환경변수를 변경할 필요없이 사용하기 위해서 입니다.

```
# wget http://www - unix.mcs.anl.gov/mpi/mpich/downloads/mpich.tar.gz
설치
# cd /usr/local
# tar xvfz mpich.tar.gz
# cd mpich
# ./configure --with-device=ch_p4 -prefix=/usr/local/mpich-1.2.7p1 -c++=pgCC ₩
                -cc=pgcc -fc=pgf77 -clinker=pgcc -flinker=pgf77 -c++linker=pgCC ₩
                -f90=pgf90 -f90inc=/usr/pgi/linux86/6.0/include -f90linker=pgf90 ₩
                -f90libpath=/usr/pgi/linux86/6.0/lib -opt="-fast" -rsh=ssh
# make
# make install
# ln -s /usr/local/mpich-1.2.7p1 /usr/local/mpich
mpich/util/machines/machines에 hostname을 넣어준다. 또는 mpirun -machinefile filename로 가능.
MPICH 테스트
# cd mpich/examples/basic
# make cpi
# mpich/bin/mpirun -np 4 cpi
(e.g. mpicc -g -o matrix matrix.c)
```

path는 /etc/profile.d/mpi.sh 에 다음과 같이 입력해두면 별도의 환경변수 셋팅없이 사용이 가능하다.

```
export MPICH=/usr/local/mpich
export PATH=$PATH:$MPICH/bin
```

mpimaker 를 이용한 MPICH 컴파일 방법

```
./mpimaker mpich-1.2.7p1.tar.gz smp gnu64 ssh
```

* mmpimaker는 tar압축해제 및 Configure, make, make install까지 자동으로 실행시켜준다

설치 확인

```
# mpicc -V

mpicc for 1.2.7 (release) of : 2005/11/04 11:54:51
Reading specs from /usr/lib/gcc/x86_64-redhat-linux/3.4.6/specs
Configured with: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --
enable-shared --enable-threads=posix --disable-checking --with-system-zlib --enable-
__cxa_atexit --disable-libunwind-exceptions --enable-java-awt=gtk --host=x86_64-redhat-linux
Thread model: posix

gcc version 3.4.6 20060404 (Red Hat 3.4.6-3)
 /usr/libexec/gcc/x86_64-redhat-linux/3.4.6/collect2 --eh-frame-hdr -m elf_x86_64 -dynamic-
linker /lib64/ld-linux-x86-64.so.2 /usr/lib/gcc/x86_64-redhat-linux/3.4.6/../../../../lib64/crt1.o
/usr/lib/gcc/x86_64-redhat-linux/3.4.6/../../../../lib64/crti.o /usr/lib/gcc/x86_64-redhat-
linux/3.4.6/crtbegin.o -L/usr/local/mpich/1.2.7p1/ip/x86_64/smp/gnu64/ssh/lib -
L/usr/lib/gcc/x86_64-redhat-linux/3.4.6 -L/usr/lib/gcc/x86_64-redhat-linux/3.4.6 -
L/usr/lib/gcc/x86_64-redhat-linux/3.4.6/../../../../lib64 -L/usr/lib/gcc/x86_64-redhat-
linux/3.4.6/../../.. -L/lib/../lib64 -L/usr/lib/../lib64 -lmpich -lpthread -lrt -lgcc --as-needed -lgcc_s
--no-as-needed -lc -lgcc --as-needed -lgcc_s --no-as-needed /usr/lib/gcc/x86_64-redhat-
linux/3.4.6/crtend.o /usr/lib/gcc/x86_64-redhat-linux/3.4.6/../../../../lib64/crtn.o
/usr/lib/gcc/x86_64-redhat-linux/3.4.6/../../../../lib64/crt1.o(.text+ 0x21): In function `_start':
: undefined reference to `main'
collect2: ld returned 1 exit status
```

각 계산 노드에도 같은 위치에 인스톨 하거나, NFS등을 이용하여 같은 위치에 mpich및 PGI가 위치할 수 있도록 해야 합니다.

예제 실행

```
# cd /usr/local/mpich/examples
# make cpi
# ./cpi

Process 0 on master
pi is approximately 3.1416009869231254, Error is 0.0000083333333323
wall clock time = 0.000105
```

mpirun을 통하여 실행

먼저 노드를 지정하기 위해, 머신파일을 생성합니다. 머신파일의 지정은 다음과 같으며 CPU의 지정타입은 hostname:n (n은 물리적인 CPU의 갯수를 지정한다)

```
# vi /usr/local/mpich/1.2.7p1/ip/x86_64/smp/gnu64/ssh/share/machines.LINUX
-----------------------------------------------------------------
```

```
node001:2 ← 각 노드들에 대한 물리적인 CPU의 개수를 지정해 한다.
node002:2
node003:2
node004:2
```

각 노드당 cpu개수 2개, 총4개의 CPU를 머신파일 마스터노드에서는 실행하지 않는 옵션으로 mpirun을 실행합니다.

```
# mpirun -np 4 -machinefile ./share/machines.LINUX -nolocal ./cpi
Process 0 on node001
Process 1 on node002
Process 2 on node003
pi is approximately 3.1416009869231241, Error is 0.0000083333333309
wall clock time = 0.000546
```

3.6.3 ATLAS Installation

ATLAS (Auto Tuning Linear Algebra Subprogram)는 수치 연산에 있어 사용되어지고 있는 BLAS (Basic Linear Algebra Subroutine)을 그 해당 Computer 에 맞게 자동으로 Tuning 하여 BLAS Library 를 만들어 주는 Program 이다. 이것으로 만들어진 BLAS Library 는 Standard BLAS Library 보다 성능이 월등하기 때문에 Linpack Benchmarking Test 를 할 경우에도 이 것을 Link 하여 사용한다. ATLAS 를 Install 하기 전에 우선적으로 고려해야 할 것이 Compiler 의 종류이다. 어떤 Compiler 로 Library 를 만들기 쉬운 반면에, 어떤 Compiler 의 경우에서는 Compile 이 쉽게 되지 않는 경우가 있다. 따라서 되도록이면, gcc-2.95 version 을 사용할 것을 권장한다.

BLAS Library 를 만들기 위한 일련의 과정을 아래에 나타내었다.

각 질문들의 자세한 내용은 www.netlib.org 를 참고하기 바란다.

```
$make
$make config CC=<ANSI C compiler>
(if you have other Compiler. Default is  gcc Compiler )
```

```
Enter number at top left of screen [0]:
Have you scoped the errata file? [y]:
Are you ready to continue? [y]:
Are you using a cross-compiler? [n]:
use express setup? [y]:
Enter Architecture name (ARCH) [Linux_P4SSE2]:
Enter Maximum cache size (KB) [512]:
Enter File creation delay in seconds [0]:
Enter f77 compiler [/usr/local/gcc-2.95.3/bin/g77]:
Enter F77 Flags [-O]:
Use supplied default values for install? [y]:
```

```
$make install arch=<ARCH>
( Linux_P4SSE2 is Default <ARCH> Name for Pentium 4 )
It will take for a while
```

```
Checking ~/ATLAS/lib/<ARCH>
( libatlas.a, libf77blas.a, libptcblas.a, libstatlas.a,
libcblas.a, liblapack.a, libptf77blas.a )
```

3.6.4 Cache Size

ATLAS 의 성능을 높이기 위한 하나의 방법으로는 Cache Edge Size 를 변경하는 것이있다. 보통 Pentium 4 의 경우 Cache Edge Size 가 209K 로 되어져 있는데, 이것의 크기를 변경하면 Cluster 의 성능을 높일 수 있다.

~/ATLAS/include/<arch>/atlas_cacheedge.h File 수정

```
#ifndef ATLAS_CACHEEDGE_H
   #define ATLAS_CACHEEDGE_H
   #define CacheEdge 210944                -> 393216 (384K)
#endif
```

~/ATLAS/bin/l3blast.c File 수정

```
 *
 * By default the mono-threaded  ATLAS  routines are tested. To test the
 * multi-threaded ATLAS routines, define the following macro:
 *    USE_L3_PTHREADS  : multi-threaded ATLAS implementation.
 */
//#define USE_F77_BLAS
#define USE_L3_REFERENCE                  // Line 79
```

~/ATLAS/bin/<arch>/ : make xsl3blastst, xdl3blastst, xcl3blastst, xzl3blastst 실행

3.6.5 GotoBLAS Installation

GotoBLAS 코드는 현재 선형대수 알고리즘의 서브루틴을 가장 빠르게 구현하고 다양한 아키텍쳐를 지원하는 알고리즘 소스 코드이다.

down load http://www.tacc.utexas.edu/resources/software/

GotoBLAS-1.10.tar.gz 다운로드후 /opt/폴더에 압축해제

```
[root@mgt01 opt]# tar xzvf GotoBLAS-1.10.tar.gz
[root@mgt01 opt]# cd GotoBLAS
[root@mgt01 opt]# vi Makefile.rule       GotoBLAS 선형알고리즘 아키텍쳐 룰셋정의
```

Configuration File 수정

```
#
#   Beginning of user configuration
#

# This library's version
REVISION = -r1.06

# Which do you prefer to use for C compiler? Default is gcc.
# I recommend you to use GCC because inline assembler is required.
C_COMPILER = GNU       ## C compiler 정의 → 리눅스 GNU 컴파일러 정의 64비트
# C_COMPILER = INTEL

# Which do you prefer to use for Fortran compiler? Default is GNU G77.
F_COMPILER = G77       ## FORTRAN compiler 정의
# F_COMPILER = G95
# F_COMPILER = GFORTRAN
# F_COMPILER = INTEL
# F_COMPILER = PGI
# F_COMPILER = PATHSCALE
# F_COMPILER = IBM
# F_COMPILER = COMPAQ
# F_COMPILER = SUN
# F_COMPILER = F2C

# If you want to build threaded version.
# You can specify number of threads by environment value
# "OMP_NUM_THREADS", otherwise, it's automatically detected.
# SMP = 1 -> 시스템환경이 SMP 환경이거나 Thread Library를 요구할 때 옵션 설정

# You may specify Maximum number of threads. It should be minimum.
# MAX_THREADS = 4 → Maximum threads 라이브러리의 개수를 정의함

# If you need 64bit binary; some architecture can accept both 32bit and
# 64bit binary(EM64T, Opteron, SPARC and Power/PowerPC).
# BINARY64    = 1

# If you want to drive whole 64bit region by BLAS. Not all Fortran
# compiler supports this. It's safe to keep comment it out if you
# are not sure.
INTERFACE64 = 1     ## 64bit binary only

# If you need Special memory management;
# Using HugeTLB file system(Linux / AIX / Solaris)
# CCOMMON_OPT     += -DALLOC_HUGETLB

# Using static allocation instead of dynamic allocation
```

```
# You can't use it with ALLOC_HUGETLB together
# CCOMMON_OPT     += -DALLOC_STATIC

# If you want to use CPU affinity
# CCOMMON_OPT     += -DUSE_CPU_AFFINITY

# If you want to use memory affinity (for NUMA)
# You can't use it with ALLOC_STATIC together
# NUMA_AFFINITY = 1

# If you want to use pure thread server model.
# Default is only OMP_NUM_THREADS - 1 threads are spawned to reduce
# thread overhead. This is not implemented yet.
# CCOMMON_OPT     += -DALL_THREADED

# Use busy wait instead of pthread_lock implemenation
# BLAS perforamnce should be better, but total performance would be
# worse due to extra usage of CPU
# CCOMMON_OPT     += -DUSE_BUSYWAIT

# If you have special compiler to run script to determine architecture.
GETARCH_CC          = gcc
GETARCH_FLAGS       =
#   End of user configuration
```

```
[root@mgt01 GotoBLAS]# make   # generating normal library
[root@mgt01 GotoBLAS]# make prof   # generating profile enabled library (linux only)
[root@mgt01 GotoBLAS]# make clean
[root@mgt01 GotoBLAS]# cd exports; make so # Shared library 생성
[root@mgt01 GotoBLAS]# ls
```

libgoto.a, libgoto_core2-r1.06.a, libgoto_core2-r1.06.so 파일 생성 확인

Sample Installation Script

```
#!/bin/bash
export CC=gcc
export FC=g77
rm -rf *.log
rm -rf *.so
tar xfz GotoBLAS-1.14.tar.gz
echo "Complete .. Unarchive"
cp Makefile.rule GotoBLAS/
cd GotoBLAS
make -j4 1>../make.log 2>../make_err.log
echo "Complete .. Make"
make prof -j4 1>../prof.log 2>../prof_err.log
echo "Complete .. Profile"
cd exports
make so 1>../../so.log 2>../../so_err.log
echo "Complete .. Build Dynamic Library"
cd ..
cp *.so ../
cd ..
#rm -rf GotoBLAS
echo "Complete .. Build GotoBLAS"
```

3.6.7 HPL Installation

HPL은 많은 사용자용 option이 있기 때문에 이 option에 따라서도 성능의 차이가 많이 날 수도 있다. 또한 Network의 성능차이 또한 Cluster Computer의 성능에 주요한 요인이 된다. HPL을 Install한다는 이야기는 곳 Compile하다는 이야기와 같다. 즉 hpl을 Compile하여 execution file을 만들고 이것의 input file인 HPL.dat file 을 수정하여 Cluster Super Computer의 성능을 평가 할 수 있다. hpl Compile은 ATLAS의 Compile과 유사하게 ~/hpl/setup Directory에 있는 적당한 Make.<arch> file을 선택하여, 수정한 후에, make arch=<arch> 명령을 실행 시키면 ~/hpl/bin/<arch> Directory에 HPL.dat file 과 execution file (xhpl)이 생긴다.
다음은 Make.<arch>file의 예이다.

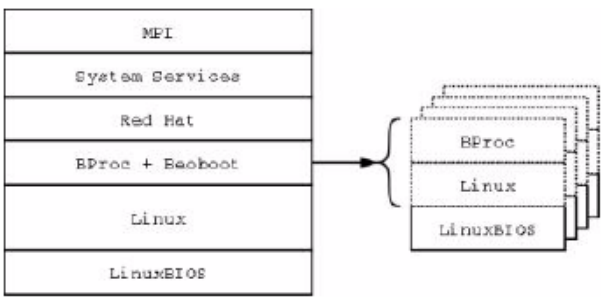
goforit@evergreen:~/Util/LINUX_src/hpl - 셸 - Konsole

Session 고치기 보기 설정 도움말

```
# ----------------------------------------------------------------------
# - shell --------------------------------------------------------------
# ----------------------------------------------------------------------
SHELL        = /bin/sh
#
CD           = cd
CP           = cp
LN_S         = ln -s
MKDIR        = mkdir
RM           = /bin/rm -f
TOUCH        = touch
# ----------------------------------------------------------------------
# - Platform identifier ------------------------------------------------
# ----------------------------------------------------------------------
ARCH         = Linux_TEST
# ----------------------------------------------------------------------
# - HPL Directory Structure / HPL library ------------------------------
# ----------------------------------------------------------------------
TOPdir       = $(HOME)/hpl
INCdir       = $(TOPdir)/include
BINdir       = $(TOPdir)/bin/$(ARCH)
LIBdir       = $(TOPdir)/lib/$(ARCH)
#
HPLlib       = $(LIBdir)/libhpl.a
# ----------------------------------------------------------------------
# - Compilers / linkers - Optimization flags ---------------------------
# ----------------------------------------------------------------------
CC           = /usr/local/lam-6.5.6-intel/bin/mpicc
NOOPT        =
CCFLAGS      = -O3
#
LINKER       = /usr/local/lam-6.5.6-intel/bin/mpif77
LINKFLAGS    = $(CCFLAGS)
#
ARCHIVER     = ar
ARFLAGS      = r
RANLIB       = echo
# ----------------------------------------------------------------------
# - Message Passing library (MPI) --------------------------------------
# ----------------------------------------------------------------------
MPdir        =
MPinc        =
MPlib        =
#
F2CDEFS      =
# ----------------------------------------------------------------------
# - Linear Algebra library (BLAS or VSIPL) -----------------------------
# ----------------------------------------------------------------------
LAdir        = /home/plab/ATLAS/lib/Linux_P4SSE2
LAinc        =
#LAlib        = $(LAdir)/libcblas.a $(LAdir)/libatlas.a
LAlib        = -L$(LAdir) -lgoto_p4_512k -lf77blas -lcblas -latlas
# ----------------------------------------------------------------------
# - HPL includes / libraries / specifics -------------------------------
# ----------------------------------------------------------------------
HPL_INCLUDES = -I$(INCdir) -I$(INCdir)/$(ARCH) $(LAinc) $(MPinc)
HPL_LIBS     = $(HPLlib) $(LAlib) $(MPlib)
#
HPL_OPTS     = -DHPL_CALL_CBLAS
#HPL_OPTS     = -DHPL_CALL_CBLAS
# ----------------------------------------------------------------------
#
HPL_DEFS     = $(F2CDEFS) $(HPL_OPTS) $(HPL_INCLUDES)
#
# ----------------------------------------------------------------------
```

MPI PATH & Compile OPTION

BLAS LIBRARY PATH & Link



HPL의 설치와 관련된 항목들은 상위단인 병렬라이브러리인 MPICH가 정상적으로 설치되었음을 가정하고 진행한다.

Step 1.

HPL의 소스코드를 다운로드 받아 설치하고자 하는 Path에 위치하여 압축화일을 해제해 준다
http://www.netlib.org/benchmark/hpl/hpl.tgz
일반적인 HPL의 소스코드의 압축해제의 위치는 /opt안에 설치한다.

Step 2.

압축해제된 HPL 디렉토리 이동하여 Machine config 파일을 복사하여 상위디렉토리에 위치후에 Config를 설정한다. /opt/hpl/setup 디렉토리에 Make.Linux_PII_FB_LAS를 Make.Linux로 Rename 하여 상위디렉토리에 복사해 넣은다.

[root@hpl/setup]cp Make.Linux_PII_FBLAS ./make.Linux

Step 3.

복사한 Make.Linux config 파일을 시스템의 설정에 맞게 재구성 한다. HPL의 설정은 GotoBLAS 라이브러리가 정상적으로 설치가 완료되었다는 가정하에 진행한다

```
* HPL 라이브러리 정의
----------------------------------------------------------------------
# - HPL Directory Structure / HPL library
# ----------------------------------------------------------------------
#

TOPdir       = /opt/hpl  → hpl이 설치된 디렉토리위치 정의
INCdir       = $(TOPdir)/include
BINdir       = $(TOPdir)/bin/$(ARCH)
LIBdir       = $(TOPdir)/lib/$(ARCH)
#
HPLlib       = $(LIBdir)/libhpl.a → /opt/hpl/lib 디렉토리를 확인하여 파일존재 확인

* MPICH 라이브러리 정의
# ----------------------------------------------------------------------
# - Message Passing library (MPI)
# ----------------------------------------------------------------------
# MPinc tells the  C  compiler where to find the Message Passing library
# header files,  MPlib  is defined  to be the name of  the library to be
# used. The variable MPdir is only used for defining MPinc and MPlib.
#
MPdir        = /usr/local/mpich/1.2.7p1/ip/x86_64/smp/gnu64/ssh → MPICH 디렉토리 정의
MPinc        = -I$(MPdir)/include
MPlib        = $(MPdir)/lib/libmpich.a
#

*Goto 라이브러리 정의
----------------------------------------------------------------------
```

```
# - Linear Algebra library (BLAS or VSIPL)
# -----------------------------------------------------------------------
# LAinc tells the  C  compiler where to find the Linear Algebra  library
# header files,  LAlib  is defined  to be the name of  the library to be
# used. The variable LAdir is only used for defining LAinc and LAlib.
#
LAdir          = /home/Goto/GotoBLAS → GotoBLAS 디렉토리 정의
LAinc          =
LAlib          = -L$(LAdir)/libgoto.a $(LAdir)/libgoto.a
```

*GNU Compiler 정의

```
-----------------------------------------------------------------------
# - Compilers / linkers - Optimization flags ---------------------------
# -----------------------------------------------------------------------
#
CC     = /usr/local/mpich/1.2.7p1/ip/x86_64/smp/gnu64/ssh/bin/mpicc
```

→ mpicc 컴파일러에 대한 정의

```
CCNOOPT        = $(HPL_DEFS)
CCFLAGS        = $(HPL_DEFS) -fomit-frame-pointer -O3 -funroll-loops
#
# On some platforms,  it is necessary  to use the Fortran linker to find
# the Fortran internals used in the BLAS library.
#

LINKER         = /usr/local/mpich/1.2.7p1/ip/x86_64/smp/gnu64/ssh/bin/mpif77
```

→ mpi Fortran77에 대한 정의를 지정

```
LINKFLAGS      = $(CCFLAGS) -lm
#
ARCHIVER       = ar
ARFLAGS        = r
RANLIB         = echo
#

# -----------------------------------------------------------------------
# - Platform identifier ------------------------------------------------
# -----------------------------------------------------------------------
#
ARCH           = Linux  →  리눅스 플랫폼에 대한 정의를 지정하여 준다.
```

Step4.

모든 설정이 완료된 후에 make arch=Linux 명령을 이용하여 지정된 GotoBLAS 라이브러리와 HPL 라이브러리에 대한 컴파일을 진행한다.

3.7 Run HPL

3.7.1 Compile

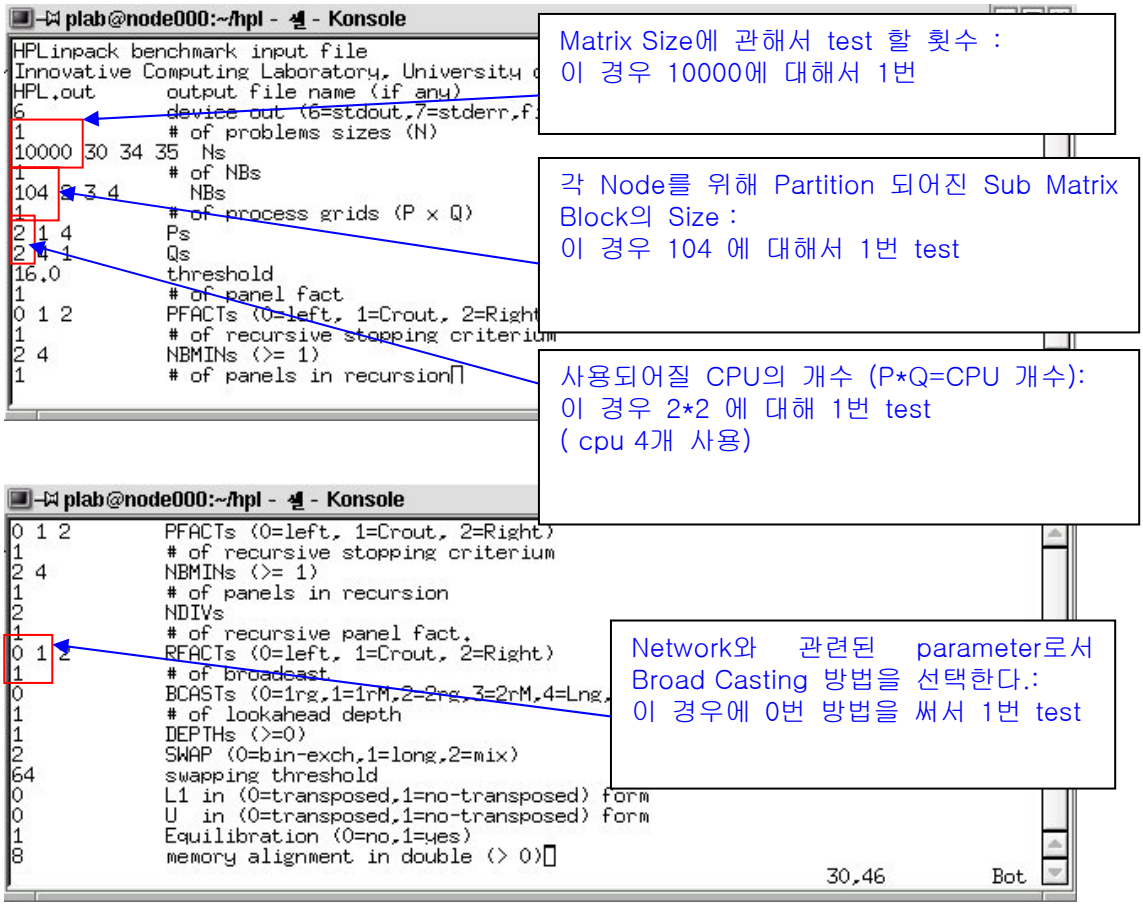
Linpack Benchmarking 은 위에서도 설명했던 것과 같이 Cluster의 성능을 평가 할 수 있다. MPICH를 이용하여 MPI Job을 실행시키기 위해서는 먼저 Compile을 mpich의 mpicc Command를 사용하여야 한다. 그 이외의 Compile 과정은 동일하다.

3.7.2 Send the Execution file

Linpack Benchmarking에서 각 Node에서 필요로 하는 execution file은 xhpl뿐이다. 따라서 scp나 psh 명령어를 이용하여 각 Node의 동일 Directory로 복사를 한다.

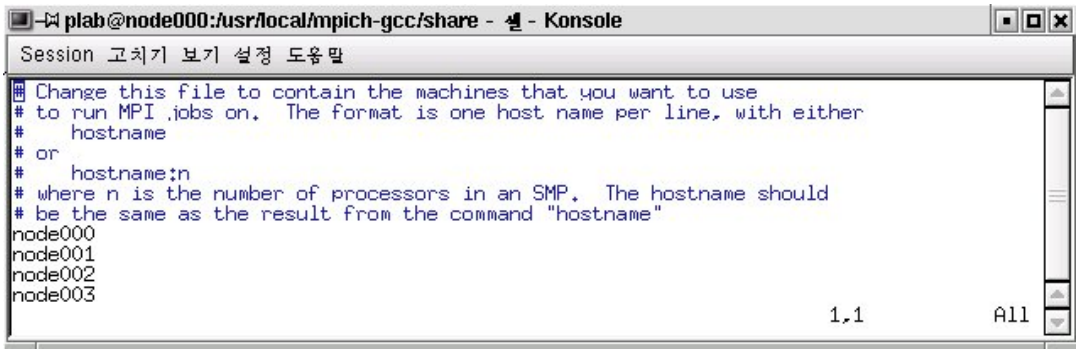
3.7.3 Preparation for MPI Job

Linpack Benchmarking test를 하기 위해서는 HPL의 input file인 HPL.dat file의 각 parameter들에 대해서 알아야 하는데, 자세한 내용은 hpl Directory내의 TUNING file에 잘 적혀져 있으므로 여기서는 Linpack Benchmarking test의 성능에 주요하게 영향을 주는 몇 가지 요소에 대하여만 설명 하도록 하겠다. 다음은 HPL.dat 의 예이다.



HPL.dat file의 parameter 들은 Linpack Benchmarking의 성능에 지대한 영향을 준다. 따라서 User가 가지고 있는 Cluster의 환경에 맞게 잘 수정해 주어야만 좋은 성능을 낼 수 있다.

MPICH가 LAM/MPI와 다른 점 중의 하나는 lamd를 실행시키기 위해 lamboot를 실행 시킬 필요가 없으며, 사용될 Node의 정보는 lamhosts file을 이용하는 것이 아니라 mpi Directory/share/machines.LINUX file을 이용한다. 아래는 그 file의 예이다.



3.7.4 xhpl 실행

```
# mpirun -np 7 xhpl
================================================================================
HPLinpack 1.0  --  High-Performance Linpack benchmark  --  September 27, 2000
Written by A. Petitet and R. Clint Whaley,  Innovative Computing Labs.,  UTK
================================================================================

An explanation of the input/output parameters follows:
T/V    : Wall time / encoded variant.
N      : The order of the coefficient matrix A.
NB     : The partitioning blocking factor.
P      : The number of process rows.
Q      : The number of process columns.
Time   : Time in seconds to solve the linear system.
Gflops : Rate of execution for solving the linear system.

The following parameter values will be used:

N      :   10000
NB     :      85
P      :       1
Q      :       7
PFACT  :   Crout
NBMIN  :       4
NDIV   :       2
RFACT  :   Right
BCAST  :  1ringM
DEPTH  :       1
SWAP   : Mix (threshold = 64)
L1     : transposed form
U      : transposed form
EQUIL  : yes
ALIGN  : 8 double precision words
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
W11R2C4        10000    85     1     7              70.49             9.460e+00
--------------------------------------------------------------------------------
||Ax-b||_oo / ( eps * ||A||_1  * N      ) =        0.0646673 ..... PASSED
||Ax-b||_oo / ( eps * ||A||_1  * ||x||_1  ) =        0.0153022 ..... PASSED
||Ax-b||_oo / ( eps * ||A||_oo * ||x||_oo ) =        0.0034203 ..... PASSED
================================================================================
```

위의 결과는 N = 10000 , NB = 85 일때 9.46Gflops 가 나왔다. LINPACK 과 마찬가지로 여러분의 시스템 환경에 맞게 problem size 와 NB 를 적절히 수정해 가면서 시스템이 수행할 수 있는 최고성능을

이끌어 내보자. HPL.dat 파일을 수정 한 다음 컴파일을 다시 한다.

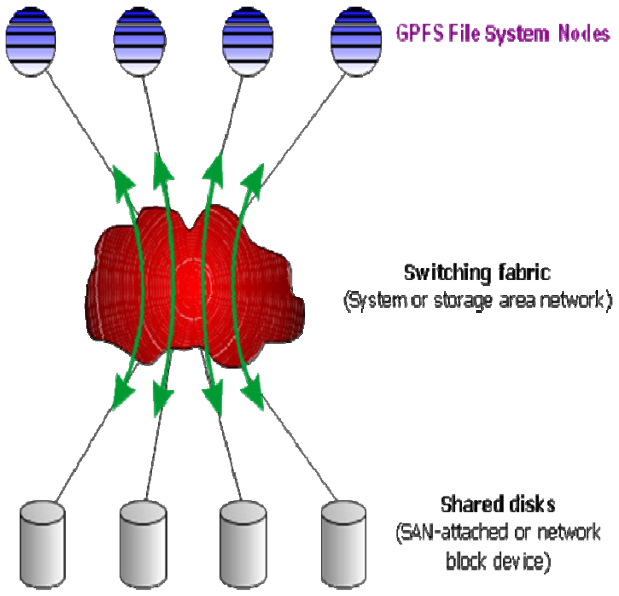
```
# rm -f ./xhpl
# cd ../../
# make clean
# make all
```

4. GPFS (General Parallel File System)

4.1 GPFS 개요

- GPFS(General Parallel File System)란?

General Parallel File System(GPFS)은 클러스터 내의 모든 노드로 부터 데이터 접근을 가능하게 하는 고성능 공유 디스크 파일 시스템이다. 병렬 또는 직렬 등 어떤 유형의 어플리케이션을 사용하더라도 표준 유닉스 파일 시스템 인터페이스를 통해 공유 파일에 쉽게 접근할 수 있으며, 복수 노드 에서도 같은 파일에 동시에 접근할 수 있다. GPFS는 로깅(logging)과 복제뿐만 아니라, failover 구성을 통해 디스크 또는 서버 오작동 시에도 고가용성을 보장한다.



- 확장성
- File 정합성
- Data 관리성
- 시스템 유연성 향상
- Data 가용성 향상
- 관리의 집중성
- 성능 향상

- 공유 파일 시스템 솔루션

복수개의 노드가 파일시스템의 파일을 공유할 수 있도록 지원

- Shared SAN Filesystem

Shared SAN을 통해 공유 노드들이 데이타와 물리적으로 직접 연결 함으로써 높은 성능과 안정성 제공 가능

- 표준 UNIX® 파일 시스템 인터페이스 제공

기존 유닉스 파일시스템과 동일하게 파일을 생성/변경/삭제. GPFS내의 파일을 관리하기 위해 별도 방법을 습득할 필요 없음.

4.2 GPFS 환경

GPFS short history & evolution

1996년 Tiger Shark File system – 멀티미디어 video server로 구현됨
1998년 GPFS 1.1 출시 – IBM SP switch이용한 고속 스위치 네트워크에 구현
2001년 최대 32 노드 지원, 리눅스 지원하는 Linux용 GPFS 1.1 출시
2002년 GPFS 2.1 출시 , GPFS 1.3 for Linux 출시
2003년 12월 GPFS 2.2 출시 – xSeries Linux, pSeries Linux, pSeries AIX
2004년 12월 GPFS 2.2 기능향상 – AIX-Linux interoperability 기능 추가
2004년 새로운 하드웨어 지원과 기능 향상된 GPFS 2.3 출시
2006년 GPFS V3.1 출시

- AIX와 Linux system의 Concurrent Sharing 지원

GPFS architecture는 AIX와 Linux 서버가 SAN based storage에 동시 연결하여 파일 공유하도록 지원
하드웨어적으로 Linux와 AIX에서 동시 접속/공유가 가능한 경우 적용 가능

- GPFS for AIX

- H/W

IBM system p5 server, IBM eServer JS20 BladeCenter server JS21 BladeCenter server.

- S/W

GPFS V3.1.0.1 : AIX V5.3 or later

GPFS V2.3.0.12 : AIX V5.2 or later

- GPFS for Linux

- H/W

IBM x86 xSeries® server, IBM BladeCenter server, IBM for AMD processor-based

Server Linux on POWER™ product (System p5 servers, BladeCenter server, OpenPower™ server)

- S/W

구축하고자 하는 system type (POWER, x86_64, IA_32)에 따라 지원 가능 커널 레벨 결정됨.

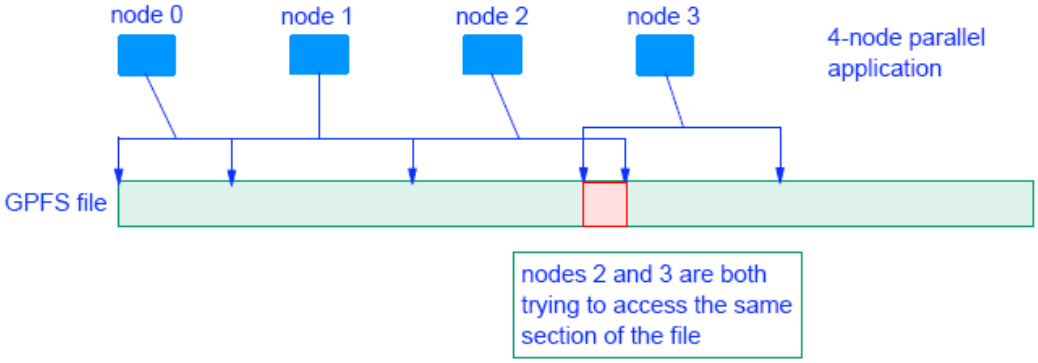
Red Hat EL 4.0 (Update 3), Red Hat EL 3.0 (Update 7)

SUSE LINUX ES 9.0 (SP3), SUSE LINUX ES 8.0 (SP4)

4.3 GPFS 특성

High Scalable File system	수백개의 노드 동시 접속과 1000개 이상의 디스크로 구성된 수백 TB 용량을 지원 이론적 최대 용량 2^99 bytes까지 지원 가능하며 현재 200 Terabytes까지 검증 Filesystem당 최대 2,147,483,648개 파일까지 생성 가능(구성에 따라 상이)
High availability	일부 노드 장애시에도 다운타임 없이 파일시스템 공유로 지속적으로 서비스 가능
High performance	Shared SAN을 통해 공유 노드들이 데이타와 물리적으로 직접 연결 함으로써 높은 성능과 안정성 제공 가능
High Flexibility	NSD (Network Shared Disk)기반의 디스크 제공으로, 직접적인 SAN 연결 없이도 GPFS 기반하의 파일시스템 공유 가능. 따라서 서버 확장이 용이하며, 용도별, I/O 용량별 유연한 구성 가능
High stability	HACMP 필요없이 GPFS 솔루션 만으로 파일시스템 공유가 가능해 짐으로써, 구성의 단순화, 관리의 단순화로 인한 안정성 향상 (GPFS v2.3이상)
Automatic Journaling(Logging) of Metadata	파일시스템 메타데이타를 각 공유 디스크에 저장하고, 장애시 신속한 복구가 가능하도록 지원
Concurrent File Access 및	정교한 Lock 관리로 데이타 접근 충돌/상충을 예방하고, 하나의 파일 여

Block Level Locking 제공	러 클라이언트가 동시에 접근할 수 있도록 함. Byte range Lock관리로, 한 파일내 겹치지 않는 부분에 대한 동시 업데이트 가능. 겹치는 영역에 대한 write/update요청시 요청 순서에 따라 serializing 하여 Lock 권한 제공 GPFS는 최대 300,000개의 토큰을 동시에 관리할 수 있음.
Data Protection	GPFS 파일시스템과 metadata에 대한 미러링 기능 제공으로 안정성 확보
Application에 최적화된 block-size 지정 가능	운영 어플리케이션의 특성에 맞도록 16K ~ 1024K까지 다양한 block size를 지정하여 최적화 할 수 있음
Block Level striping	파일시스템 생성시 지정한 block size단위로 자동 Striping함으로써 I/O 성능 최대화 제공
Online 상에서의 Disk add / remove 가능	시스템 운영중에 디스크 추가/삭제가 가능하고, 추가/삭제시 동적으로 데이타를 재분배 해 줌으로써, I/O balancing을 유지
표준 UNIX® 파일 시스템 인터페이스 제공	기존 유닉스 파일시스템과 동일하게 파일을 생성/변경/삭제. GPFS내의 파일을 관리하기 위해 별도 방법을 습득할 필요 없음



4.4 GPFS 설치

4.4.1 GPFS 설치 전 준비사항

GPFS에 포함될 모든 nodes 에 Linux OS 설치
각 node들 /etc/hosts파일 작성
root 사용자가 각 nodes에게 password prompt 없이 접속 가능하게 SSH 구성
GPFS nodes 시간 동기화를 위해 NTP서버 구성 권장
GPFS nodes에 외장 스토리지 볼륨 인식

4.4.2 GPFS 설치 및 Portability layer 빌드

- GPFS 설치 디렉토리 생성

mkdir /gpfslpp

- CD-ROM으로부터 GPFS 제품 이미지 복사 gpfs_install-3.1*

```
[root@blade1 gpfslpp]# ./gpfs_install-3.1.*
[root@blade1 gpfslpp]# ls
gpfs.base-3.1*.rpm gpfs.gpl-3.1*.noarch.rpm
gpfs.msg.en_US-3.1*.noarch.rpm          gpfs.docs-3.1*.noarch.rpm
[root@blade1 gpfslpp]# rpm –ivh gpfs*
```

- node profile 에 환경 변수 추가

PATH=$PATH:$HOME/bin:/usr/lpp/mmfs/bin

MANPATH=$MANPATH:/usr/lpp/mmfs/messages

- Portability layer Build 하기

cd /usr/lpp/mmfs/src

export SHARKCLONEROOT=/usr/lpp/mmfs/src

cd /usr/lpp/mmfs/src

make Autoconfig

make World

make InstallImages

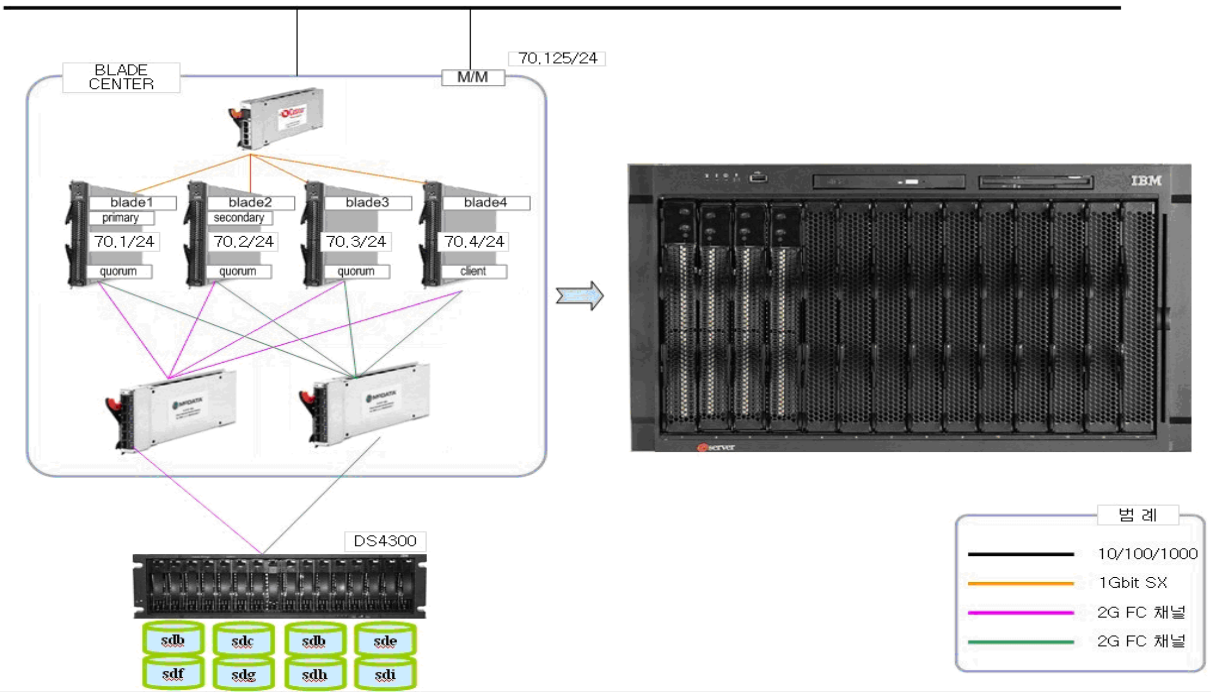
```
[root@blade1 ~]# make InstallImages
cd gpl-linux; /usr/bin/make InstallImages;
make[1]: Entering directory `/usr/lpp/mmfs/src/gpl-linux'
mmfslinux
mmfs26
lxtrace
dumpconv
tracedev
make[1]: Leaving directory `/usr/lpp/mmfs/src/gpl-linux'
```

- 각 GPFS nodes에 생성된 파일 복사

```
[root@blade1 ~]# cd /usr/lpp/mmfs/bin
[root@blade1 bin]# scp mmfslinux mmfs26 lxtrace dumpconv tracedev gpfs2:/usr/lpp/mmfs/bin
/* Copy to all the other nodes */
```

4.5 GPFS 구성

4.5.1 GPFS Test 시스템 구성도



4.5.2 GPFS구성 시 고려사항

- GPFS node 정의

NodeName:NodeDesignations:AdminNodeName

optional, "-" node roles 구분
manager | client – GPFS node 를 정의 할 때 manager node 혹은 client node에 대한 node pool을 정의한다.
quorum | nonquorum – Node pool에서 node가 quorum 아니면 non-quorum node로 동작 할 것인지에 대한 node role 정의. 정의 하지 않으면 non-quorum node로 동작
GPFS Cluster primary와 secondary node는 반드시 quorum nodes 구성 할 것을 권장한다.

- GPFS NSD 생성 옵션 확인

DiskName:PrimaryServer:BackupServer:DiskUsage:FailureGroup:DesiredName:StoragePool

DiskName – NSD로 정의될 block device 이름 ex) /dev/sdb, /dev/hdisk1
PrimaryServer – Primary NSD 서버 이름
BackupServer – Backup NSD 서버 이름
DiskUsage – Disk Usage 정의
- dataAndMetadata – disk가 data와 metadata 둘 다 가진다. default값
- dataOnly – disk가 data만 가진다.
- metadataOnly – disk가 metadata만 가진다
- descOnly – disk가 data 및 metadata 모두를 가지지 않고 단지 file system descriptor의 복사본만 가진다. DR 구성시에 third failure group으로 사용됨
FailureGroup – disk replication을 위한 failureGroup number 지정
DesiredName – NSD disk 이름 지정 지정하지 않으면 gpfsNNnsd형식으로 자동생성
StoragePool – NSD가 할당된 Storage Pool 이름 지정

ex) /dev/sdb:gpfs_primary:gpfs_secondary:dataAndMetadata:1::stgpool01

- GPFS file system 생성 옵션 확인

File System 생성 options	Default value
-F DiskDesc 파일 시스템 디스크정의	none
-A {yes | no | automount} yes – gpfs 데몬이 시작될 때 자동 마운트 no – 수동 마운트 automount – 파일시스템 처음으로 access될 때 마운트	yes
-B BlockSize 16K,64K,256K,512K,1024K or 1M data block size	256K
-E {yes | no} mtime 값 기록	yes
-m DefaultMetadataReplicas	1

metadata replication 값 {1 | 2} 1 – metadata replication 하지 않음 2 – metadata replication	
-M MaxMetadataReplicas metadata replication 값 {1 | 2} 1 – metadata replication 하지 않음 2 – metadata replication	1
-n NumNodes 마운트 되는 nodes 숫자	32
-o MountOptions	none
-Q {yes | no} to active quota	no
-r DefaultDataReplicas data replication 값 {1 | 2} 1 – data replication 하지 않음 2 – data replication	yes
-R MaxDataReplicas data replication 값 {1 | 2} 1 – metadata replication 하지 않음 2 – metadata replication	yes
-v {yes | no} disk usage 검증	yes

4.5.3 GPFS Cluster 구성

- GPFS Cluster 생성 및 확인

```
[root@blade1 gpfs]# mmcrcluster –n nodefile –p blade1 –s blade2 –r /usr/bin/ssh –R /usr/bin/scp –
C gpfs_test
! -p primary server –s secondary server
nodefile 의 내용 >
        blade1:manager-quorum
        blade2:manager-quorum
        blade3:manager-quorum
        blade4:client
[root@gpfs1 gpfslpp]# mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         test.gpfs
  GPFS cluster id:           13882422822655908522
  GPFS UID domain:           test.gpfs
  Remote shell command:      /usr/bin/ssh
  Remote file copy command:  /usr/bin/scp

GPFS cluster configuration servers:
-----------------------------------
  Primary server:    gpfs1
  Secondary server:  blade2.cluster.net

 Node  Daemon node name      IP address       Admin node name          Designation
-------------------------------------------------------------------------------
   1   blade1                192.168.70.1     blade1                   quorum-manager
   2   blade2.cluster.net    192.168.70.2     blade2.cluster.net       quorum-manager
   3   blade3.cluster.net    192.168.70.3     blade3.cluster.net       quorum-manager
   4   blade4.cluster.net    192.168.70.4     blade4.cluster.net
```

- NSD Disk 생성 및 확인

```
[root@blade1 gpfslpp]# mmcrnsd -F diskfile -v no
diskfile 의 내용 >
/dev/sdb:blade1:blade2:dataAndMetadata:1::
/dev/sdc:blade1:blade2:dataAndMetadata:1::
/dev/sdd:blade1:blade2:dataAndMetadata:1::
/dev/sde:blade1:blade2:dataAndMetadata:1::
/dev/sdf:blade2:blade1:dataAndMetadata:2::
/dev/sdg:blade2:blade1:dataAndMetadata:2::
/dev/sdh:blade2:blade1:dataAndMetadata:2::
/dev/sdi:blade2:blade1:dataAndMetadata:2::
[root@blade1 gpfslpp]# mmlsnsd

 File system    Disk name      Primary node              Backup node
---------------------------------------------------------------------------
 (free disk)    gpfs1nsd       blade1                    blade2.cluster.net
 (free disk)    gpfs2nsd       blade1                    blade2.cluster.net
 (free disk)    gpfs3nsd       blade1                    blade2.cluster.net
 (free disk)    gpfs4nsd       blade1                    blade2.cluster.net
 (free disk)    gpfs5nsd       blade2.cluster.net        blade1
 (free disk)    gpfs6nsd       blade2.cluster.net        blade1
 (free disk)    gpfs7nsd       blade2.cluster.net        blade1
 (free disk)    gpfs8nsd       blade2.cluster.net        blade1

* 아직 filesystem 이 만들어 지지 않아 File system부분이 free disk로 나타난다
```

- File System 생성 및 mount

file system을 생성 하기 전에 먼저 mmstartup -a 명령으로 gpfs 데몬을 시작해야 한다.

```
nsd 디스크 생성 이후 diskfile 의 내용 >
# /dev/sdb:blade1:blade2:dataAndMetadata:1::
gpfs1nsd:::dataAndMetadata:1::
# /dev/sdc:blade1:blade2:dataAndMetadata:1::
gpfs2nsd:::dataAndMetadata:1::
# /dev/sdd:blade1:blade2:dataAndMetadata:1::
gpfs3nsd:::dataAndMetadata:1::
# /dev/sde:blade1:blade2:dataAndMetadata:1::
gpfs4nsd:::dataAndMetadata:1::
# /dev/sdf:blade2:blade1:dataAndMetadata:2::
gpfs5nsd:::dataAndMetadata:2::
# /dev/sdg:blade2:blade1:dataAndMetadata:2::
gpfs6nsd:::dataAndMetadata:2::
# /dev/sdh:blade2:blade1:dataAndMetadata:2::
gpfs7nsd:::dataAndMetadata:2::
# /dev/sdi:blade2:blade1:dataAndMetadata:2::
# mmcrnsd로 nsd생성 후 diskfile 내용이 파일시스템 생성에 필요 값으로 바뀐다.

[root@blade1 gpfslpp]# mmcrfs /gpfs1 gpfs1 -F diskfile  -B 1024K -m1 -M1 -r1 -R1
# replication 기능을 사용 하기위해서는 -m2 -M2 -r2 -R2 로 옵션을 변경해야 한다.
[root@blade1 gpfslpp]# mmmount gpfs1 -a
[root@blade1 gpfslpp]# mmlsmount all -L

File system gpfs1 is mounted on 4 nodes:
  192.168.70.1     blade1
  192.168.70.2     blade2.cluster.net
  192.168.70.3     blade3.cluster.net
  192.168.70.4     blade4.cluster.net
```

```
File system gpfs2 is mounted on 4 nodes:
   192.168.70.2      blade2.cluster.net
   192.168.70.3      blade3.cluster.net
   192.168.70.1      blade1
   192.168.70.4      blade4.cluster.net

[[root@blade1 gpfslpp]# mmlsnsd

 File system    Disk name     Primary node              Backup node
---------------------------------------------------------------------------
 gpfs1          gpfs1nsd      blade1                    blade2.cluster.net
 gpfs1          gpfs2nsd      blade1                    blade2.cluster.net
 gpfs1          gpfs3nsd      blade1                    blade2.cluster.net
 gpfs1          gpfs4nsd      blade1                    blade2.cluster.net
 (free disk)    gpfs5nsd      blade2.cluster.net        blade1
 (free disk)    gpfs6nsd      blade2.cluster.net        blade1
 (free disk)    gpfs7nsd      blade2.cluster.net        blade1
 (free disk)    gpfs8nsd      blade2.cluster.net        blade1
```

4.6 GPFS (GENERAL PARALLEL FILE SYSTEM) 운영 및 관리

4.6.1 GPFS 파일시스템 확인 및 관리

파일시스템 생성 확인 명령 mmlsdisk, mmlsfs

```
[root@blade1 gpfslpp]# mmlsdisk gpfs1
disk         driver   sector failure holds    holds                          storage
name         type       size   group metadata data  status        availability   pool
------------ -------- ------ ------- -------- ----- ------------- ------------ -
-----------
gpfs1nsd     nsd         512       1 yes      yes   ready         up           system
gpfs2nsd     nsd         512       1 yes      yes   ready         up           system
gpfs3nsd     nsd         512       1 yes      yes   ready         up           system
gpfs4nsd     nsd         512       1 yes      yes   ready         up           system

[root@blade1 gpfslpp]# mmlsfs gpfs1
flag value            description
---- -------------- -----------------------------------------------------
 -s  roundRobin       Stripe method
 -f  32768            Minimum fragment size in bytes
 -i  1024             Inode size in bytes
 -I  32768            Indirect block size in bytes
 -m  1                Default number of metadata replicas
 -M  1                Maximum number of metadata replicas
 -r  1                Default number of data replicas
 -R  1                Maximum number of data replicas
 -j  cluster          Block allocation type
 -D  posix            File locking semantics in effect
 -k  posix            ACL semantics in effect
 -a  1048576          Estimated average file size
 -n  32               Estimated number of nodes that will mount file system
 -B  1048576          Block size
 -Q  none             Quotas enforced
     none             Default quotas enabled
 -F  417792           Maximum number of inodes
 -V  9.03             File system version. Highest supported version: 9.03
 -u  yes              Support for large LUNs?
 -z  no               Is DMAPI enabled?
```

```
 -E   yes                Exact mtime mount option
 -S   no                 Suppress atime mount option
 -K   whenpossible       Strict replica allocation option
 -P   system             Disk storage pools in file system
 -d   gpfs8nsd;gpfs9nsd;gpfs10nsd;gpfs11nsd   Disks in file system
 -A   yes                Automatic mount option
 -o   none               Additional mount options
 -T   /gpfs1             Default mount point
```

- File System 사용공간 확인하기

mmdf 명령어를 이용하여 모든 GPFS 디스크들의 용량 확인

ex) mmdf gpfs1

- file system mount 및 umount

GPFS 파일시스템을 생성할 때 자동으로 마운트되는 옵션(디폴트값)으로 생성하면 GPFS 데몬이 시작될 때 자동으로 마운트된다.

mmmount gpfs1 -a (전 nodes에 mount 할 때 -a옵션 사용)

mmumount gpfs1 -a

- File System 제거하기

GPFS 파일시스템을 제거하기 위해서는 먼저 파일시스템을 unmount해야 한다.

GPFS 파일시스템은 mmdelfs 명령어로 제거할 수 있다.

mmdelfs 명령어

mmdelfs gpfs1 -p (여기서 p 옵션은 파일시스템정보가 잘못되었거나 디스크가 available한 상태가 아니라도 강제로 삭제할 수 있다.)

- File System 오류 체크 및 복구하기

gpfs의 mmfsck 명령은 online, offline 두가지 모드에서 사용할 수 있다.

파일시스템이 마운트되어 있는 online상태에서는 -o 옵션을 사용하여 파일시스템을 체크할 수 있다. 반대로 파일시스템이 unmount된 상태에서는 offline으로 파일시스템을 체크할 수 있다.

Online 모드는 파일시스템이 마운트된 상태에서 배치되지 못한 블럭들을 체크하고 복구한다. 이외의 문제는 체크하여 보고만 하고 복구는 하지 않는다.

위와 다른 오류들을 해결하기 위해서는 파일시스템이 unmount된 offline모드에서 mmfsck명령을 실행하여야 한다.

Disk상태가 down상태라면 mmfsck 명령을 사용할 수 없다. 먼저 mmlsdisk명령어로 디스크 상태를 확인하고 down된 디스크가 있으면 mmchdisk명령으로 unrecovered 나 up상태로 만든다.

mmfsck 명령어

ex) mmfsck gpfs1 -n (n옵션은 파일시스템 복구를 시도하지 않고 체크만한다.)

- File System 속성 변경하기

mmchfs 명령어를 사용하여 GPFS 파일시스템의 속성을 바꿀 수 있다.

Usage:

```
mmchfs Device [-A {yes | no | automount}] [-D {posix | nfs4}] [-E {yes | no}]
              [-F MaxNumInodes] [-k {posix | nfs4 | all}]
              [-K {no | whenpossible | always}]
              [-m DefaultMetadataReplicas] [-o MountOptions]
              [-Q {yes | no}] [-r DefaultDataReplicas] [-S {yes | no}]
              [-T Mountpoint] [-V] [-z {yes | no}]
  or
mmchfs Device [-W NewDeviceName]
```

- File System Restriping 하기

GPFS는 파일을 write하면서 striping을 한다. 그런데 디스크를 추가한 후에 performance를 향상시키기 위해서는 mmrestripefs 명령어를 이용하여 파일을 restripe할 수 있다. 디스크 복제(replication)를 하지 않은 상태라면 –m 옵션을 사용하여 fail된 디스크로부터 데이터를 안전하게 복구할 수 있다. 마찬가지로 –r 옵션도 같은 효과를 낼 수 있다. 디스크 복제(replication)를 한 상태라면 –b(rebalance) 옵션으로 replication data도 restripe할 수 있다.

mmrestripefs 명령어

ex) mmrestripefs gpfs1 –r

- File System 단편화 정보 보기 및 최적화

파일시스템에 데이터를 기록할 때 fragmented block으로 조각나는 것은 피할 수 없다. 그러나 이런 조각들중에 데이터를 가득채우지 못하고 block의 일부만 사용하는 단편들도 생기게 된다. 이런 단편들은 디스크 공간을 낭비하게 되는 요인중의 하나가 된다. 이렇게 디스크 공간을 낭비하는 fragmented block에 조각모음을 하여 디스크 공간의 효율을 높이기 위해 mmdefragfs 명령어를 사용한다.

파일시스템의 단편화 정도를 알기 위해서는 mmdefragfs 명령어에 –i 옵션을 사용한다.

- File System 최적화

Block의 사용율을 높이기 위한 조각모음을 하기 위해서는 mmdefragfs 명령어를 실행한다.

-u 옵션으로 목표로 하는 block 사용율 (Block utilization)을 지정할 수 있다.

mmdefragfs 명령어

ex) mmdefragfs gpfs1

- File System 디스크 추가

파일시스템에 파일들이 증가하여 파일시스템 공간을 늘려야 할 때 GPFS는 빠르게 디스크를 추가하여 파일시스템 공간을 늘릴 수 있다. 이때 mmadddisk 명령어를 사용한다.

```
root@blade1 gpfslpp]# mmlsnsd -F

 File system   Disk name    Primary node             Backup node
---------------------------------------------------------------------------
 (free disk)   gpfs5nsd     blade2.cluster.net       blade1
 (free disk)   gpfs6nsd     blade2.cluster.net       blade1
 (free disk)   gpfs7nsd     blade2.cluster.net       blade1
 (free disk)   gpfs8nsd     blade2.cluster.net       blade1
```

*** free disk 로 남아있는 gpfs5nsd 디스크를 gpfs1 파일 시스템에 추가 할 예정**

```
[root@blade1 gpfslpp]# mmadddisk gpfs1 gpfs5nsd:::dataAndMetadata:1:: -r -v no
```

*** 여기서 r 옵션은 디스크를 추가한 다음 rebalance 하라는 옵션임**

```
[root@blade1 gpfslpp]# mmlsnsd
 File system   Disk name    Primary node             Backup node
---------------------------------------------------------------------------
 gpfs1         gpfs1nsd     blade1                   blade2.cluster.net
 gpfs1         gpfs2nsd     blade1                   blade2.cluster.net
 gpfs1         gpfs3nsd     blade1                   blade2.cluster.net
 gpfs1         gpfs4nsd     blade1                   blade2.cluster.net
 gpfs1         gpfs5nsd     blade2.cluster.net       blade1
 (free disk)    gpfs6nsd     blade2.cluster.net       blade1
 (free disk)    gpfs7nsd     blade2.cluster.net       blade1
 (free disk)    gpfs8nsd     blade2.cluster.net       blade1
```

* mmlsnd 명령으로 gpfs1 파일 시스템에 디스크가 추가 된 것을 확인 할 수 있다.

- File System에서 디스크 제거

디스크를 제거하기 전에 mmdf 명령어로 디스크를 삭제해도 파일시스템에 여유공간이 남는지 파일시스템의 남아있는 용량을 확인해야 한다. 디스크를 삭제하려면 남아있는 용량이 삭제할 디스크 용량의 150%이상이어야 한다.

*** 앞에서 추가한 gpfs5nsd디스크를 gpfs1 파일 시스템에서 제거**

```
[root@blade1 gpfslpp]# mmdeldisk gpfs1 gpfs5nsd -r
[root@blade1 gpfslpp]# mmlsnsd

 File system   Disk name    Primary node             Backup node
---------------------------------------------------------------------------
 gpfs1         gpfs1nsd     blade1                   blade2.cluster.net
 gpfs1         gpfs2nsd     blade1                   blade2.cluster.net
 gpfs1         gpfs3nsd     blade1                   blade2.cluster.net
 gpfs1         gpfs4nsd     blade1                   blade2.cluster.net
 (free disk)    gpfs5nsd     blade2.cluster.net       blade1
 (free disk)    gpfs6nsd     blade2.cluster.net       blade1
 (free disk)    gpfs7nsd     blade2.cluster.net       blade1
 (free disk)    gpfs8nsd     blade2.cluster.net       blade1
```

*** gpfs5nsd 디스크가 free disk로 변경 된 것 을 확인 할 수 있다.**

4.6.2 GPFS Node 관리

- GPFS Cluster에 nodes 추가 및 제거

mmaddnoe명령으로 gpfs cluster에 새로운 node를 추가 시킬 수도 있고 mmdelnode 명령으로 삭제 할 수 도 있다.

ex) mmaddnode blade4

mmdelnode blade4

mmlsnode -a 명령으로 cluster nodes 확인

- File System 관리 nodes 변경 및 확인

파일시스템을 관리하는 노드 정보는 mmlsmgr 명령어로 볼 수 있다.

mmlsmgr

mmlsmgr gpfs1

위 명령어를 실행하면 아래와 같이 파일시스템의 device name과 해당 파일시스템을 관리하는 노드가 어떤 노드인지를 보여준다.
또한 파일시스템을 관리하는 노드를 mmchmgr 명령어로 원하는 노드로 바꿀 수 있다.
mmchmgr
mmchmgr gpfs1 blade3
위 명령어를 실행하면 아래와 해당 파일시스템을 관리하는 노드가 blade1에서 blade2로 바뀐 것을 알 수 있다.

- GPFS Daemon 시작 및 중지

GPFS는 mmstartup 명령어를 사용하여 시작할 수 있다.
mmstartup -a
위 명령어를 실행하면 cluster에 있는 모든 노드에서 GPFS 데몬이 시작된다.
mmstartup -N NodeName
위 명령이 실행되면 명령을 실행한 노드의 GPFS 데몬만 올라온다.

GPFS는 mmshutdown 명령어로 중지시킬 수 있다.
mmshutdown -a
위 명령어를 실행하면 모든 노드에서 GPFS 데몬이 중지된다.
mmshutdown -N NodeName
위 명령어를 실행하면 해당 노드의 GPFS 데몬만 중지된다.

mmgetstate 명령으로 gpfs 서비스를 확인 할 수 있다.
mmgetstate -a 명령으로 전체 nodes 서비스 확인
mmgetstate -aL 명령으로 quorum node확인

4.6.3 GPFS Config 변경

GPFS Cluster configuration을 변경 할 때는 gpfs 데몬을 내리고 작업해야 한다.

- GPFS Cluster configuration data의 변경

mmchcluster
usage : mmchcluster -p ngstfe1_gpfs (primary server를 바꿀때 사용)
mmchcluster -s ngdtfe1_gpfs (secondary server를 바꿀때 사용)
mmchcluster -C cluster_name (cluster name을 바꿀대 사용)
mmchcluster -p LATEST (primary server config 정보를 동기화 할 때)

- GPFS cluster configuration parameter 의 변경

mmchconfig
usage : mmchconfig attribute=value -i (실행후 즉시 반영되고 restart후에도 지속됨)
mmchconfig attribute=value -I (실행후 즉시 cluster에 반영됨 restart후 적용 안됨)

Attribute

autoload - cluster node가 rebooting 시 자동으로 GPFS start

maxFilesToCache - 최근에 사용한 inode 개수 default (1000)

maxMBpS - Data 전송속도 default (150MB/s)

maxStatCache - stat cache에 유지할 inode개수

pagepool - cache on each node default (64MB) > 500M

tiebreakerDisks - node quorum과 같이 사용

umountOnDiskFail - disk에 장애가 발생했을 때 강제로 file system을 umount 시킴 (yes or no)

```
* GPFS test 서버의 Configuration 값 확인
[root@blade1 gpfslpp]# mmlsconfig
Configuration data for cluster gpfstest.blade1:
------------------------------------------
clusterName gpfstest.blade1
clusterId 13882422822655914773
clusterType lc
autoload no
useDiskLease yes
maxFeatureLevelAllowed 906
maxMBpS 512
maxblocksize 2048k
pagepool 256M
[blade1]
takeOverSdrServ yes

File systems in cluster gpfstest.blade1:
---------------------------------------
/dev/gpfs1
```

4.7 GPFS(GENERAL PARALLEL FILE SYSTEM) 구성변경 및 장애복구

4.7.1 GPFS cluster IP 및 host name 변경

GPFS IP address와 hostnames을 변경하려면 아래의 단계를 따라야 한다.

전체 nodes IP addresses 와 hostnames 변경

```
IP addresses 와 hostnames 변경 전에
1.primary GPFS cluster 서버에서 /var/mmfs/gen/mmsdrfs file백업
2.configuration parameter설정을 기록한다.
3.mmshutdown -a 명령으로 전 nodes gpfs 데몬 중지
        ex) [root@blade1 gpfslpp]# mmexportfs all -o gpfs_backup
        Usage:
        mmexportfs {Device | all} -o ExportfsFile
4.mmexportfs all 명령으로 file system 정보를 outfile로 만들어 내보낸다.
  ! mmexportfs 명령을 실행하면 기존 cluster에서 filesystem 및 disk정보가 사라짐
5.IP addresses 와 hostnames 변경
6.mmcrcluster 명령으로 cluster 재생성
7.mmchconfig 명령으로 configuration parameter 복구
8.mmimportfs all 명령으로 file system 및 disk 정보를 복구
        ex) [root@gpfs1 gpfslpp]# mmimportfs all -i gpfs_backup
        Usage:
        mmimportfs {Device | all} -i ImportfsFile [-S ChangeSpecFile]
```

Client nodes IP addresses 와 hostnames 변경

```
IP addresses 와 hostnames 변경 전에
```

```
1.primary GPFS cluster 서버에서 /var/mmfs/gen/mmsdrfs file백업
2.mmshutdown -a 명령으로 전 nodes gpfs 데몬 중지
3.mmdelnode 명령으로 변경될 node 제거
4.mmaddnode 명령으로 기존 cluster에 포함 시킨다
  primary or secondary configuration서버 IP addresses 와 hostnames 변경
5.mmchcluster 명령으로 primary or secondary 서버 role 변경
  mmchcluster -p PrimaryNode  primary로 변경 될 node명
  mmchcluster -s SecondaryNode  secondary로 변경 될 node명

ip addresses 및 hostnames 변경 후 configuration 서버 role 변경
```

primary or backup NSD 서버 IP addresses 와 hostnames 변경

```
1.mmchnsd 명령으로 primary or backup NSD 서버 없이 directly-attached NSD로 변경
        [root@blade1  /]# mmchnsd "gpfs1nsd;gpfs2nsd;gpfs3nsd;gpfs4nsd"

2.ip addresses 및 hostnames 변경 후 mmchnsd변경으로 복구
        [root@gpfs1  /]# mmchnsd "gpfs1nsd:blade1:blade2:::"

3.mmlsnsd 명령으로 primary or backup NSD 서버 확인
 [root@gpfs1  /]# mmlsnsd -d "gpfs1nsd"

  File system   Disk name     Primary node              Backup node
 ---------------------------------------------------------------------------
  gpfs1         gpfs1nsd      blade1                    blade2
```

4.7.2 GPFS cluster configuration data 파일 복구

GPFS cluster configuration data 파일은 /var/mmfs/gen/mmsdrfs 파일에 저장된다. 시스템 장애로 인한 서버 재 설치나, 사용자 실수로 인하여 mmsdrfs파일이 삭제되는 경우 다른 서버에 있는 mmsdrfs파일로 구성정보를 복구 할 수 있다. 하지만 최신 level로 파일을 동기화 하기 위해서 primary or secondary 파일로 복구 할 것을 권장 한다. GPFS cluster 구성 후 모든 nodes mmsdrfs 파일은 백업 받는 것을 권장 한다.

```
1.mmshutdown -a 명령으로 모든 nodes에서 gpfs 데몬 중지
2.mmsdrfs 파일이 삭제된 node에서 mmsdrrestore 명령 실행
        [root@blade2  /]# mmsdrrestore -p blade1 -F /var/mmfs/gen/mmsdrfs -R /usr/bin/scp
        Usage:
        mmsdrrestore -p remoteNode -F remoteFile -R RemoteFileCopyCommand
3.mmchcluster -p LATEST명령으로 최신 level로 동기화
        [root@blade2  /]# mmchcluster  -p LATEST
4.mmstartup -a 명령으로 장애 서버가 정상적으로 동작하는지 확인
```

5. XEN

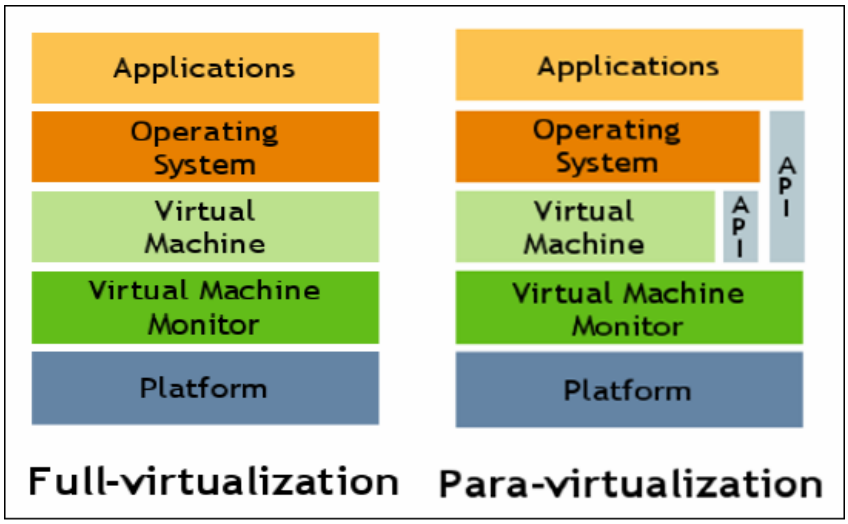
5.1 Xen

서버의 자원을 가상화 하여, 하나의 물리적인 서버에 여러 가지 OS를 설치할 수 있는 가상머신의 환경을 만들어 주는 솔루션으로, Intel Server 에 사용하는 Vmware 와, IBM System p 에 사용하는 LPAR 와 유사하다.

5.2 Xen 에서 지원하는 가상화 형태

A) Para Virtualization

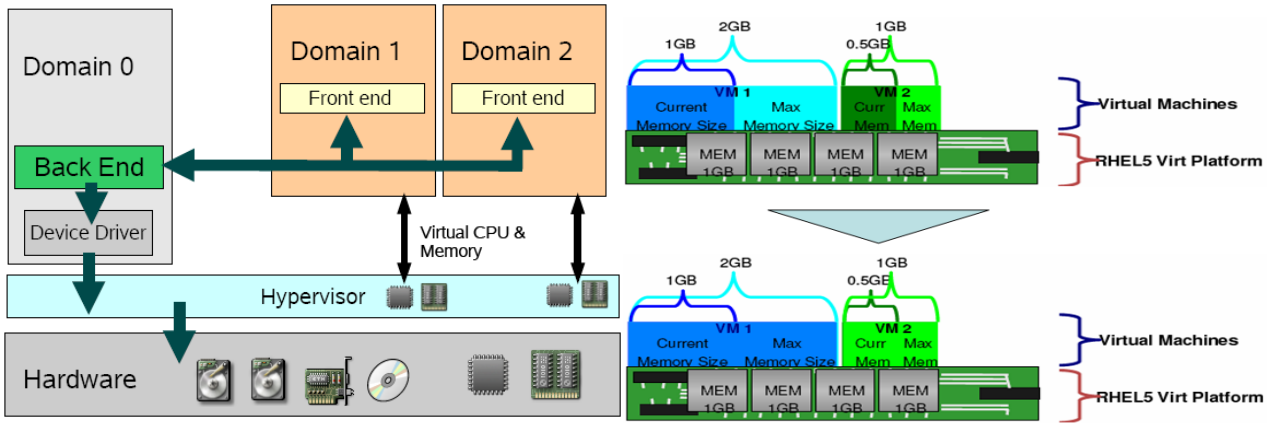
System의 모든 자원을 가상화 하는 것이 아니라, 설치되는 가상머신이 Xen Kernel 과 Communication하기 위한 API가 있어야 하는데 이 경우에는 Virtual Machine 에 설치되는 OS 의 Kernel 이 수정되어야 한다. 그러나 Full Virtualization 에 비해 성능을 좋은 편이며, Memory Ballooning 을 지원하여 시스템의 Memory를 좀더 효율적으로 관리 가능하다.



B) Full Virtualization

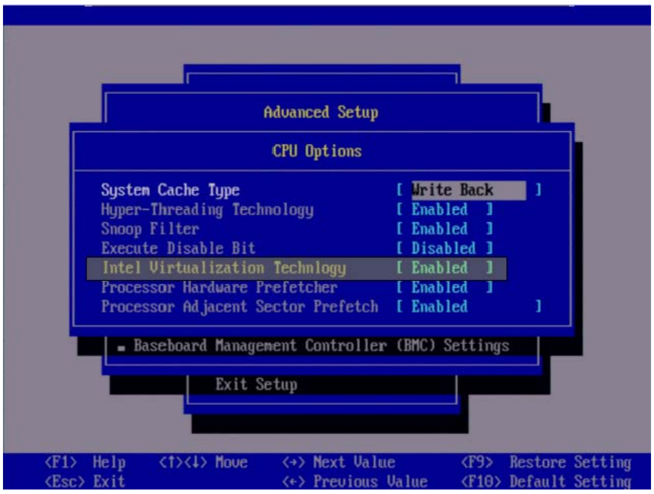
Intel 및 AMD 에서 제공하는 Virtualization 기능이 있고, BIOS 에서 본 기능을 Enable해야만 사용 가능하다.

5.3 Xen 의 기본 구조 및 Memory Ballooning 이란?



위의 내용과 같이 Xen 의 기본 구조를 살펴 보면, HW 위에 Hypervisor 가 올라가 있으며, 실제 Virtual Domain 을 관리 하기 위한 Domain0 가 있고, Domain 1,2,… 이런 가상 머신은 Hypervisor 를 통하여 실제 하드웨어를 Access 하게 되어 있습니다. 또한 Full Virtualization 에서는 가상 머신이 Operation 중에 시스템 Memory Size를 확장 할 수 없지만, Para-Virtualization 환경에서는 위의 Memory Ballooning 의 그림처럼, 시스템 OS 를 Operation 중에 시스템 메모리의 영역의 확장이 가능하다.

5.4 Xen 구성 시 확인 사항



Para-virtualized guest install하기 위해서는 CPU가 PAE를 지원해야 함. x86_64, ia64는 PAE를 지원하지만 i386의 경우는 아래와 같이 PAE를 지원하는지 확인해야 한다.

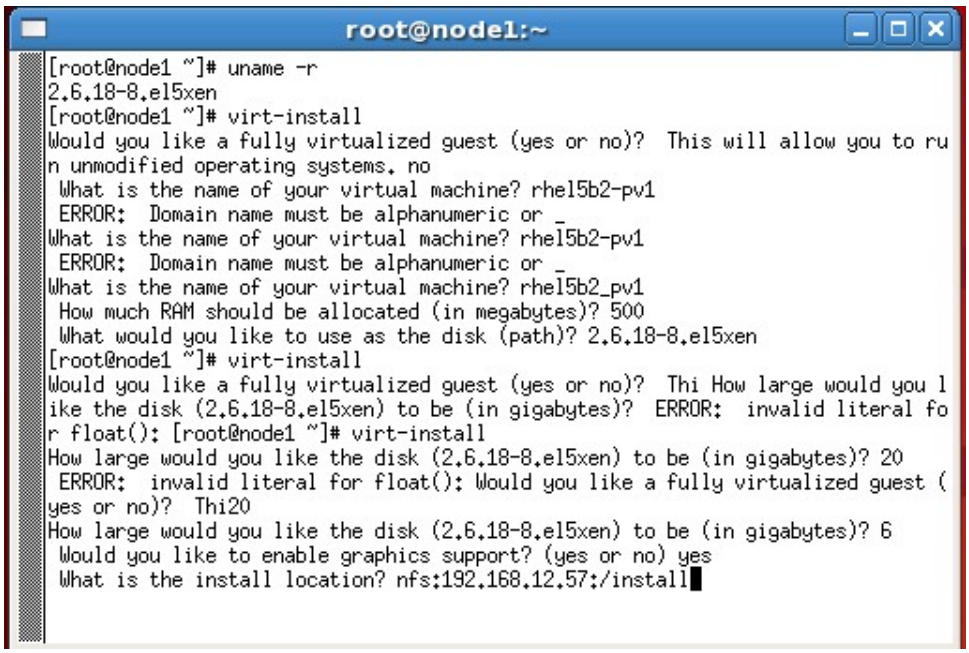
```
# grep pae /proc/cpuinfo
flags           : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts acpi mmx fxsr
sse sse2 ss ht tm syscall nx lm constant_tsc pni monitor ds_cpl vmx est tm2 cx16 xtpr lahf_lm
flags           : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts acpi mmx fxsr
sse sse2 ss ht tm syscall nx lm constant_tsc pni monitor ds_cpl vmx est tm2 cx16 xtpr lahf_lm
flags           : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts acpi mmx fxsr
sse sse2 ss ht tm syscall nx lm constant_tsc pni monitor ds_cpl vmx est tm2 cx16 xtpr lahf_lm
flags           : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts acpi mmx fxsr
sse sse2 ss ht tm syscall nx lm constant_tsc pni monitor ds_cpl vmx est tm2 cx16 xtpr lahf_lm
```

Full-virtualized guest install하기 위해서는 CPU가 Intel-VT 또는 AMD-V를 지원해야 한다.
확인하는 방법은 아래와 같다.

```
# egrep -e 'vmx|svm' /proc/cpuinfo
flags : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe constant_tsc pni monitor vmx est tm2 xtpr
```

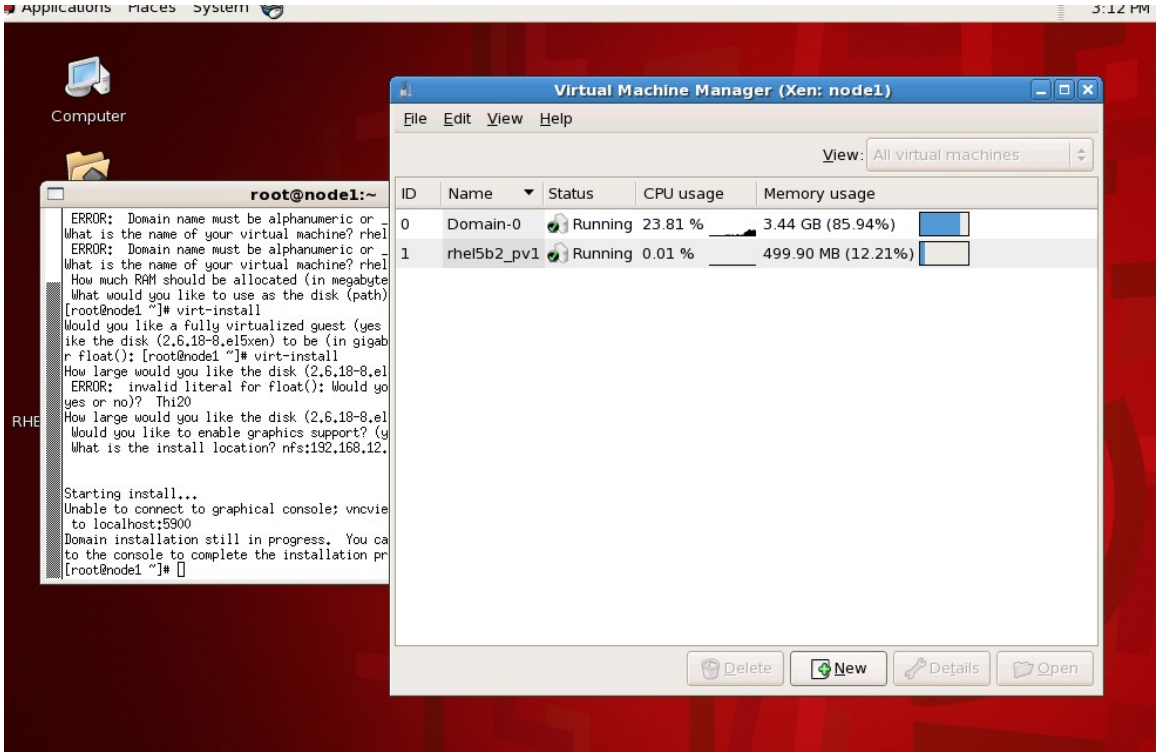
5.5 virt-install 명령으로 Xen para-virtualized guest install

1. virt-install 명령 실행후에 full virtualized guest로 인스톨할 것인가를 물으면 no를 입력. 만약 Full-virtualized guest로 install 하려면 yes를 입력
2. virtual machine name를 입력(이름은 임의로) : rhel5b_pv1
3. virtual machine이 사용할 메모리양을 입력(megabyte) : 500
4. guest가 파일시스템으로 사용할 guest image 이름을 입력(절대경로 포함) : /xen/rhel5b_pv1.img
5. guest가 사용할 디스크 용량을 입력(gigabyte): 6
6. guest OS install source를 입력 : nfs:192.168.12.57:/install 또는 http://192.168.12.57/install
7. Para-virtualization은 network intall만 지원함. 따라서 NFS나 http로 OS source를 미리 export해 놓아야 함.
8. installation이 진행됨.

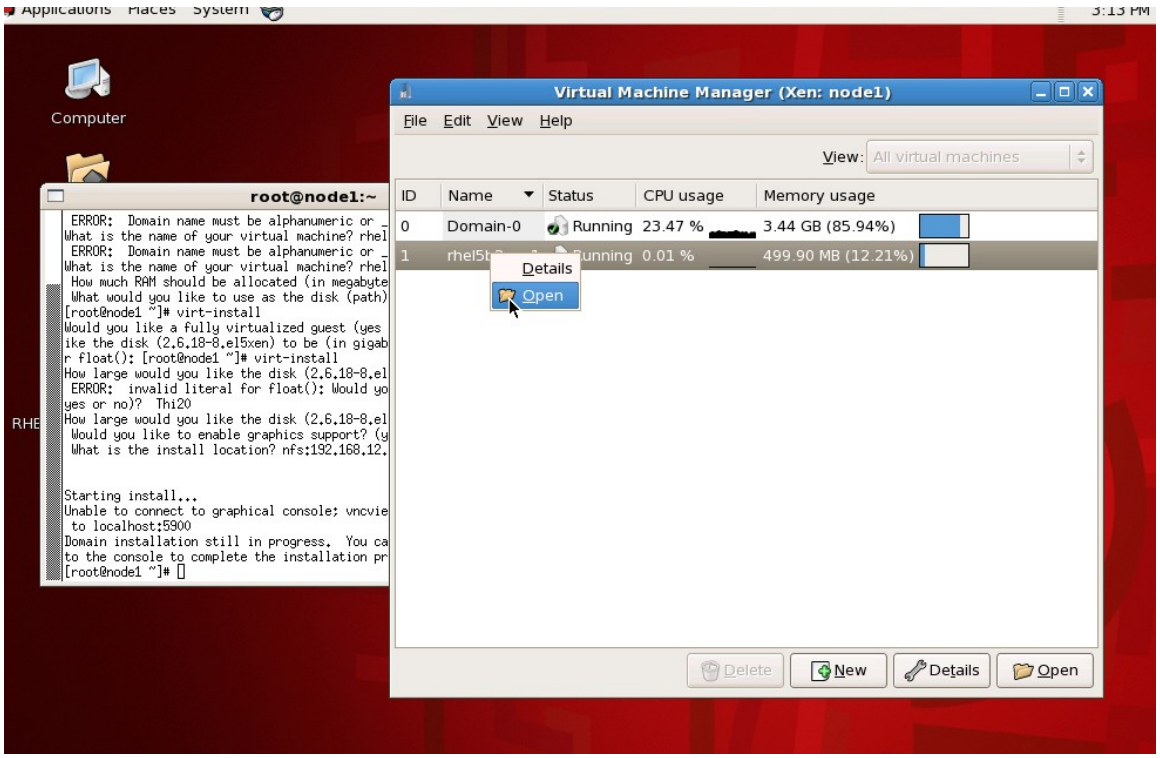


- xwindow상의 virt manager를 통해 install을 계속 진행할 수 있음.(graphics support를 yes로 할 경우, no로 했을 시는 test mod로 install이 진행됨.)

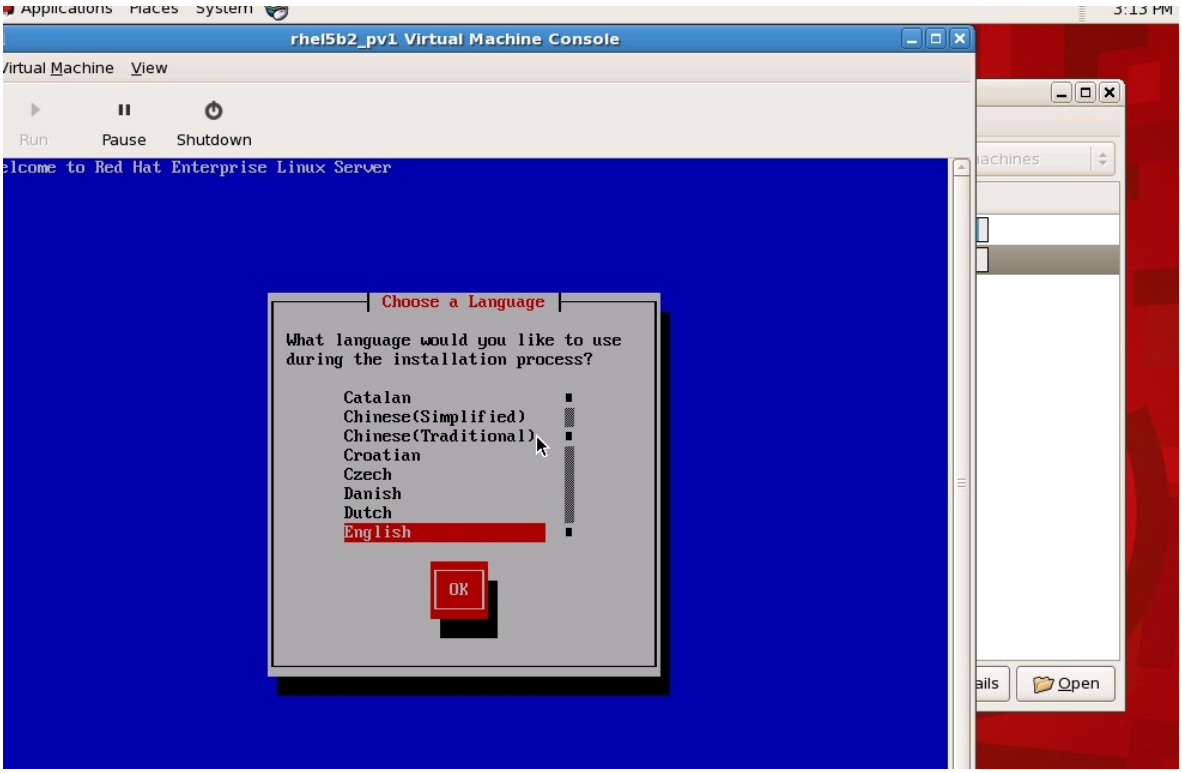
Virtual machine 실행 guest domain running 확인



Guest domain을 선택하여 open을 하면 해당 guest domain의 콘솔창이 뜸.

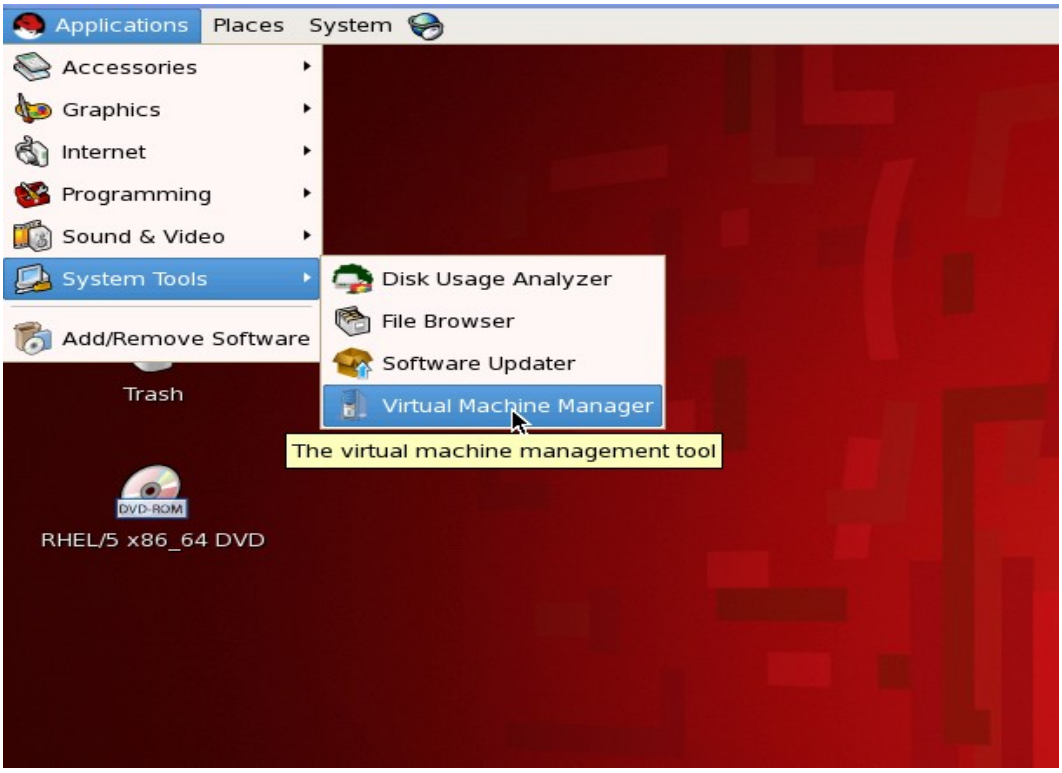


계속 설치 진행

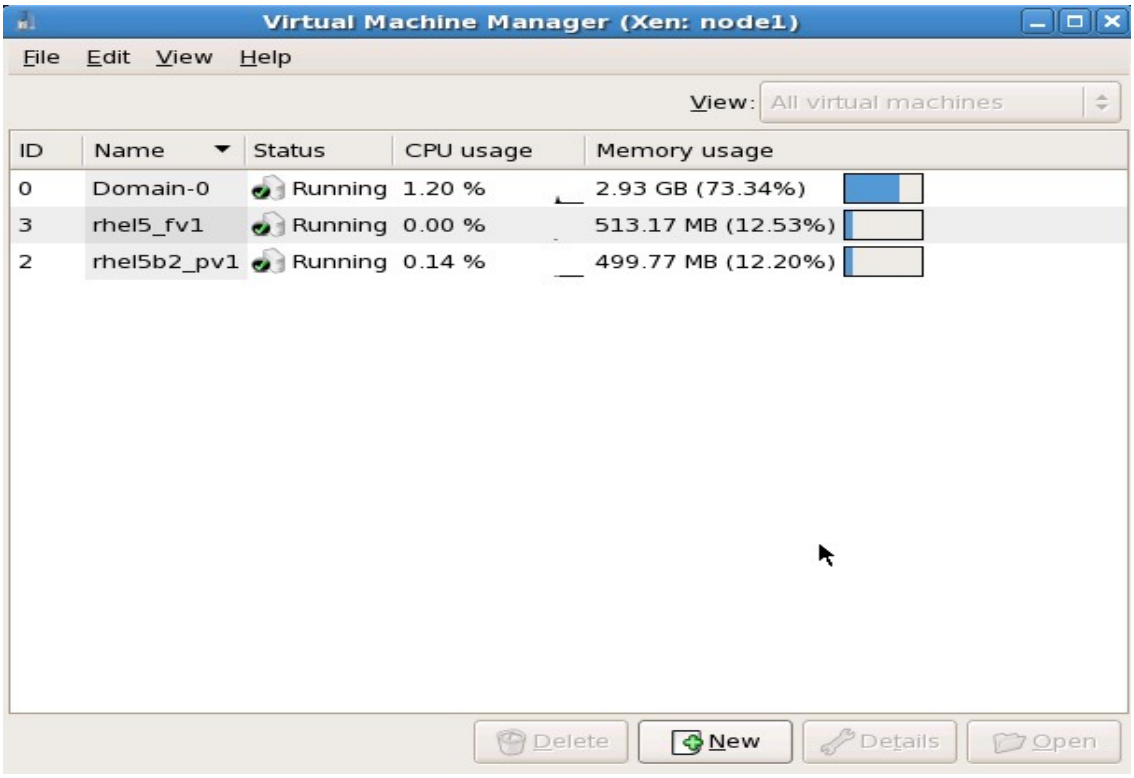


5.6 virt-manager명령으로 Xen para-virtualized guest install

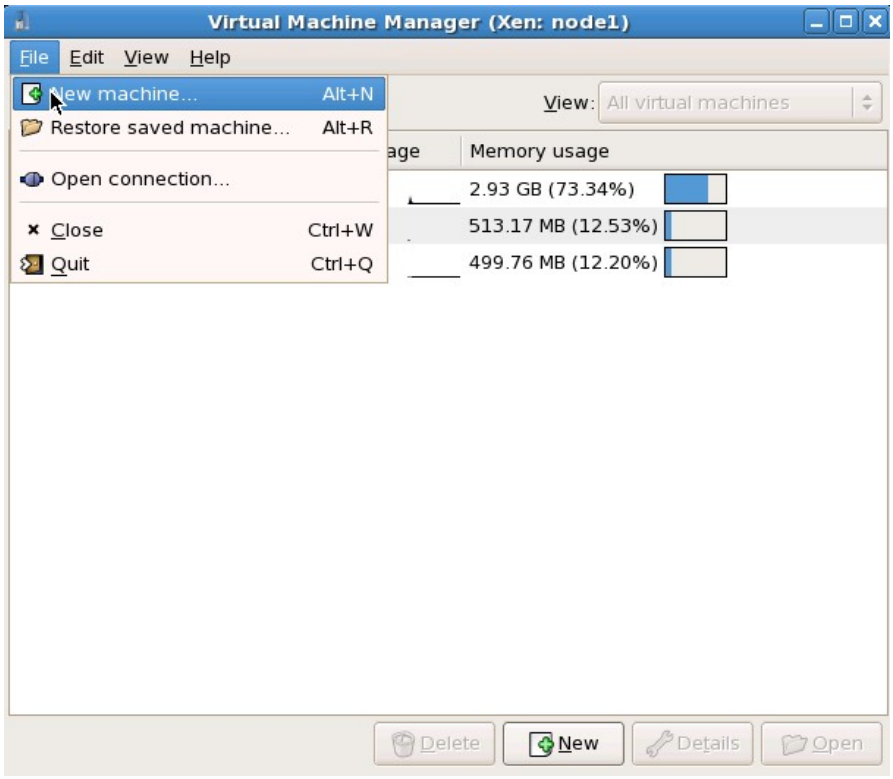
A) xwindows 콘솔창에서 virt-manager 명령을 입력하거나 아래와 같이 X윈도우 application 메뉴에서 선택



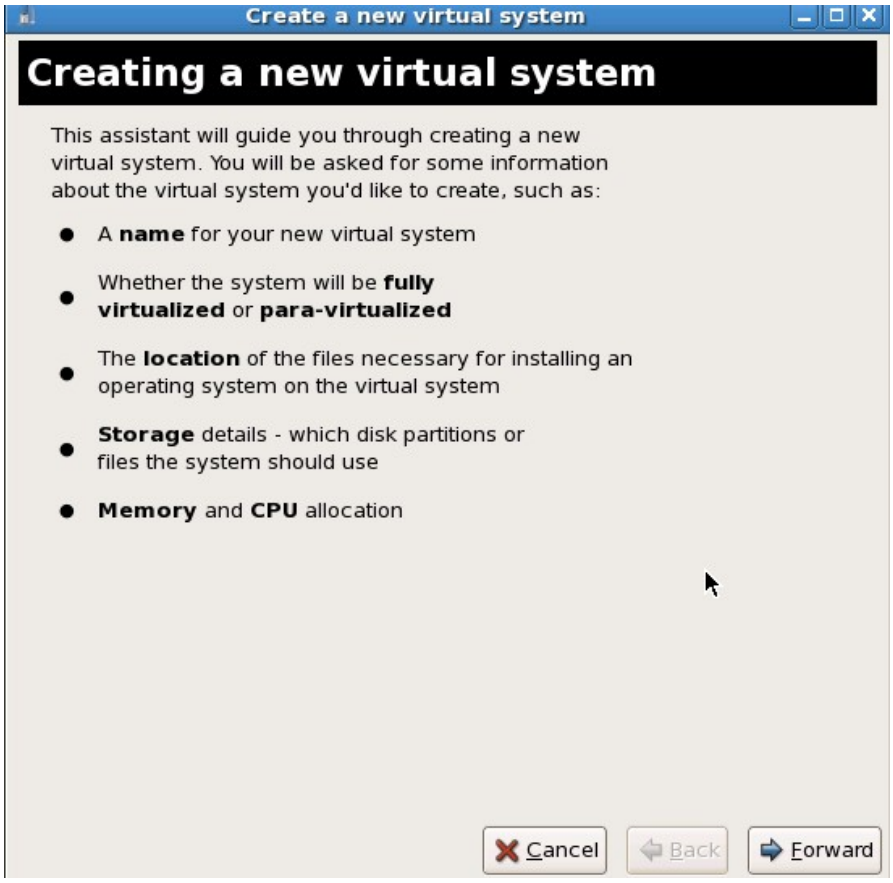
B) Virtual Machine Manager 기본창이 팝업된다.



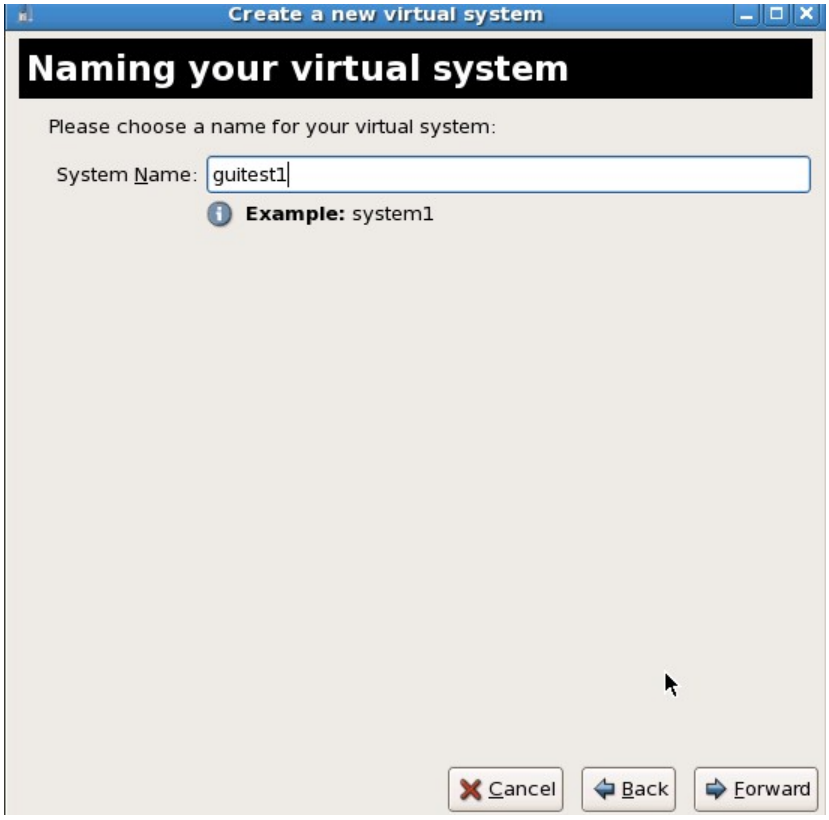
C) 상단 메뉴에서 아래와 같이 new machine을 선택



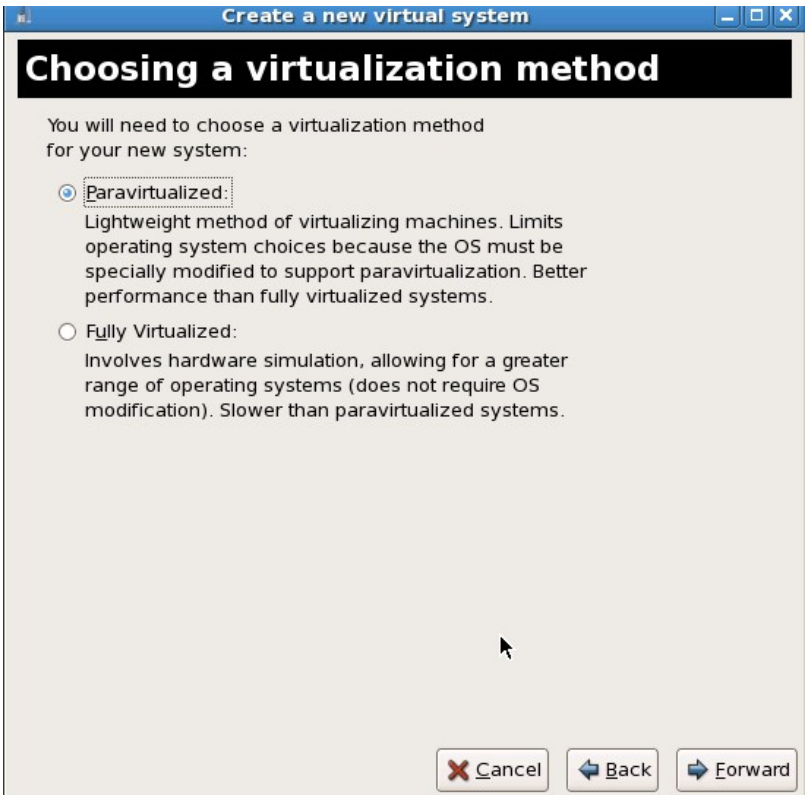
D) 아래와 같은 창이 팝업되면 Forward 선택



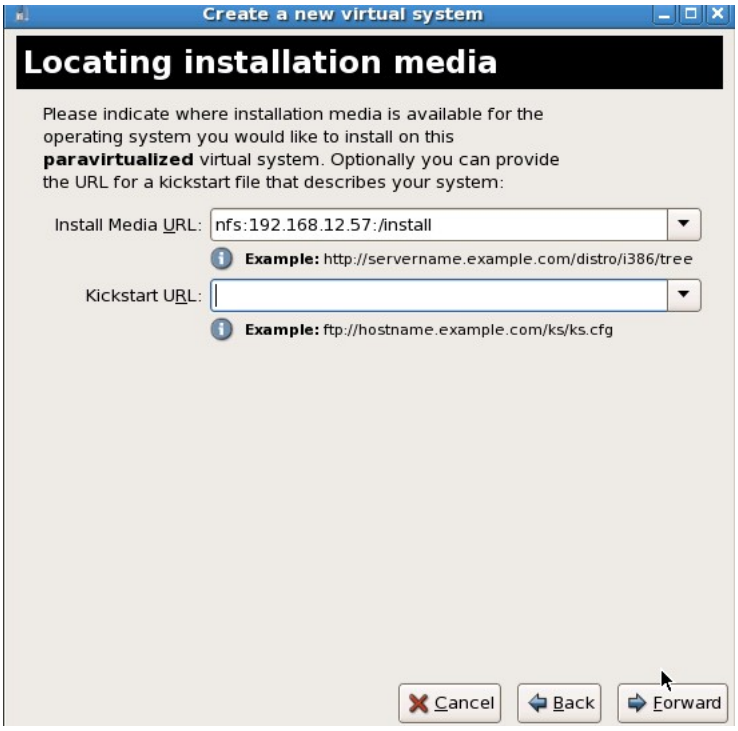
E) virtual system 이름을 정하여 입력



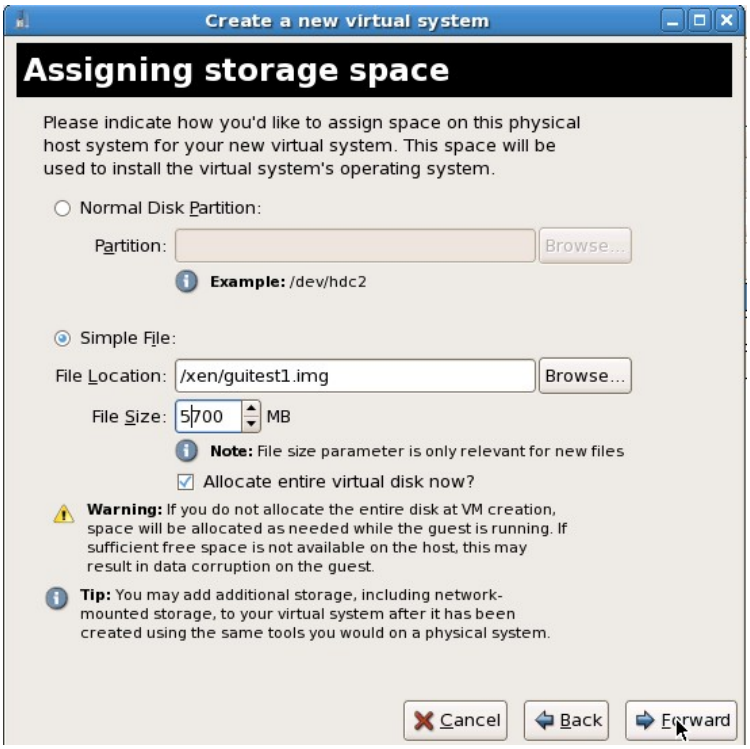
F) paravirtualized를 할 지 Fully virtualized를 할지 선택 후 Forward 선택



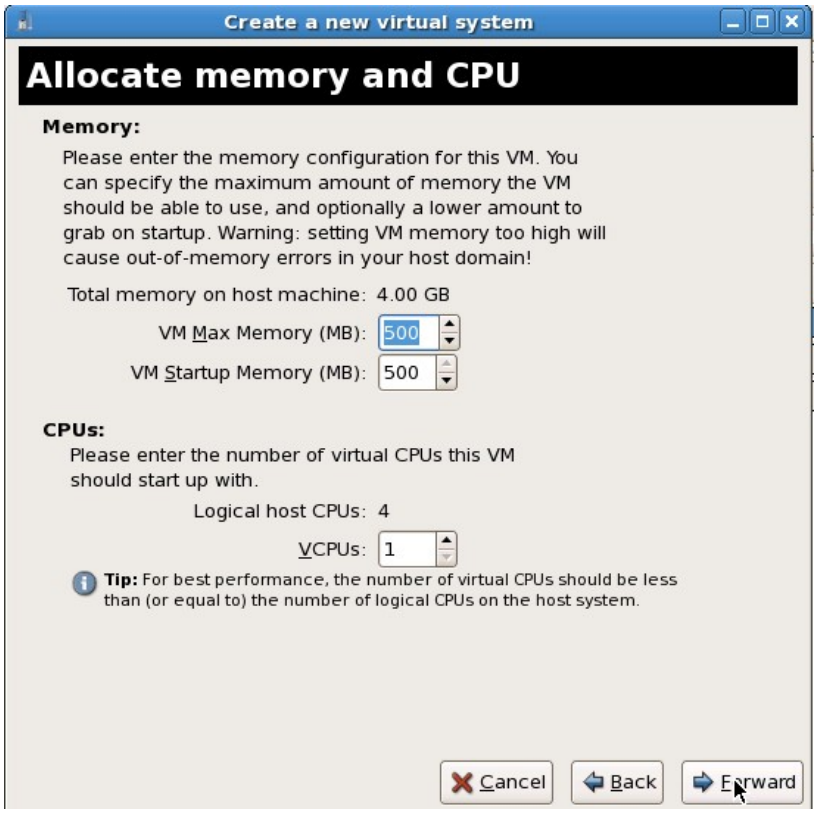
G) paravirtualized를 선택했을 경우는 아래와 같이 network install source 주소를 입력하는 메뉴로 넘어간다. 미리 구성한 NFS, HTTP 서버 주소를 입력하고 kickstart 파일이 준비되어 있으면 그 주소를 입력한다.



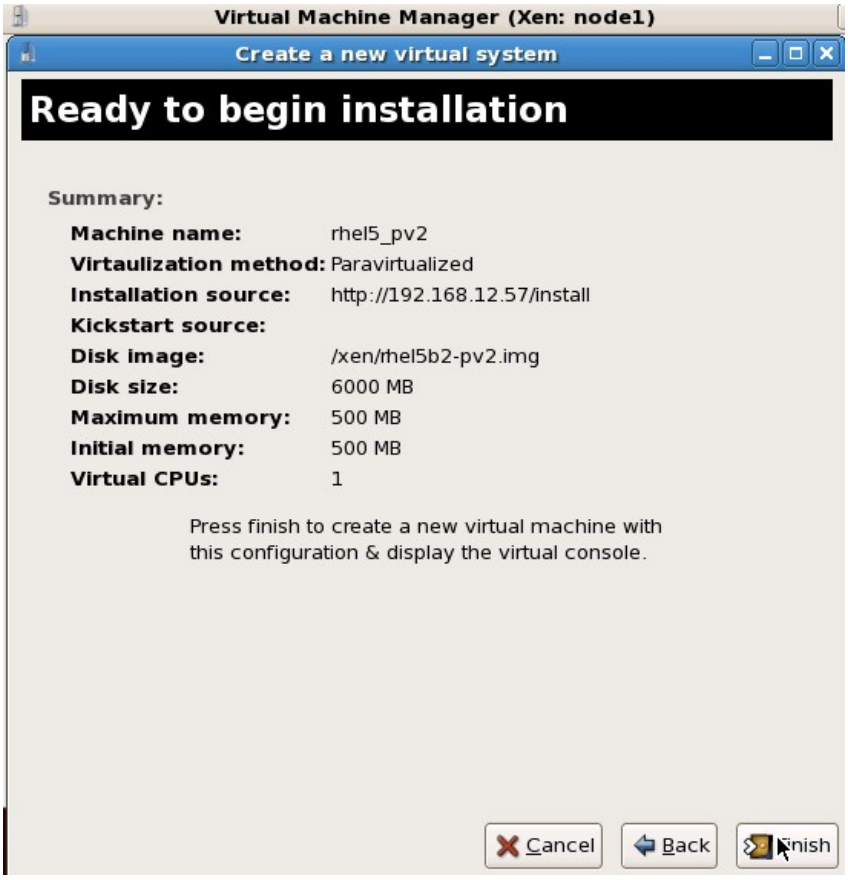
H) virtual system을 설치할 스토리지 영역을 설정한다. 물리적 파티션이 준비되어 있으면 첫번째 Normal Disk Partition에 해당 파티션의 디바이스명을 입력하고 이미지파일에 설치를 하려면 Simple file을 절대경로/name.img 형식으로 입력을 해준다.



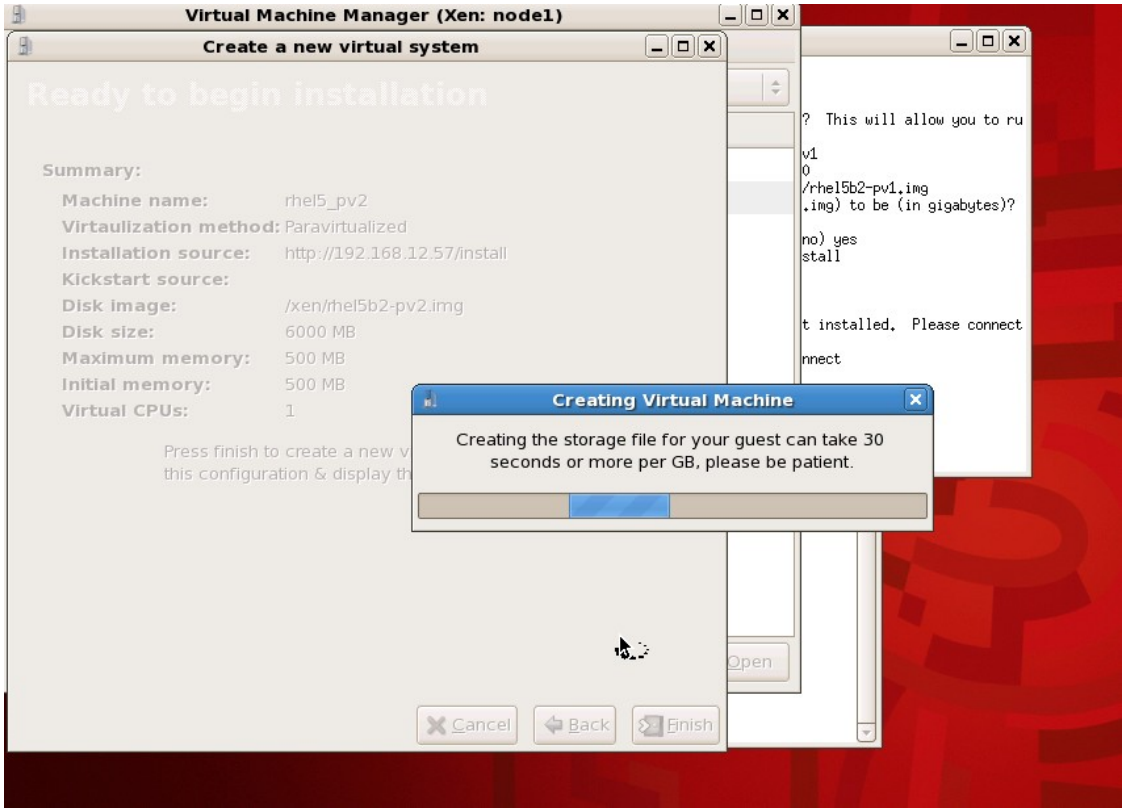
I) 다음 화면에서 vitual system에 할당할 메모리와 CPU 개수와 최대 허용량을 선택



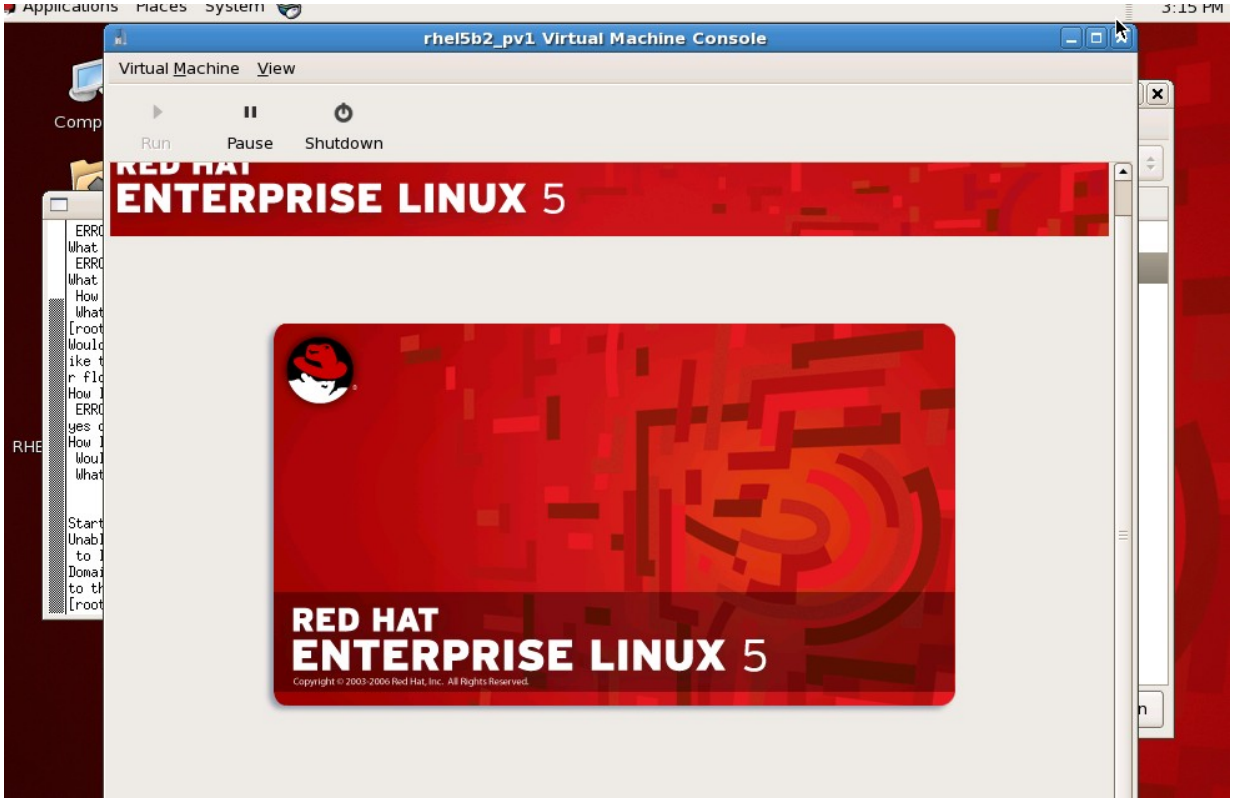
J) 다음 화면에서 최종 설정 정보들을 확인한 후에 Finish를 선택



K) 설정작업을 끝내면 virtual machine 생성작업이 실행된다

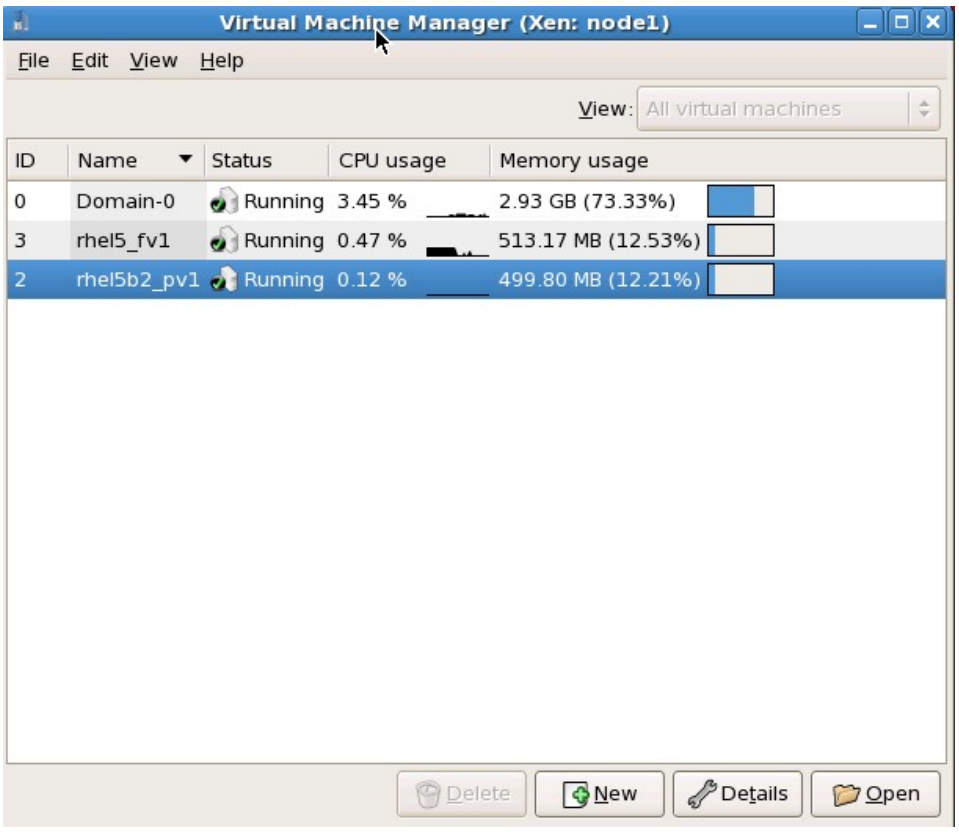


L) OS 설치화면이 팝업되면 일반 OS와 동일하게 install을 진행한다

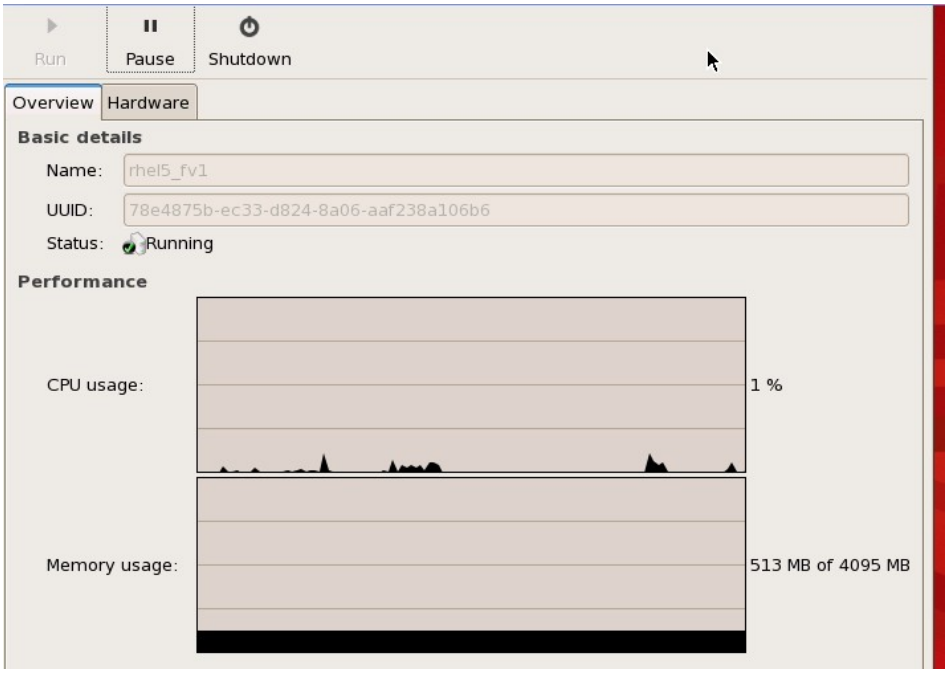


5.7 Virtual system Management

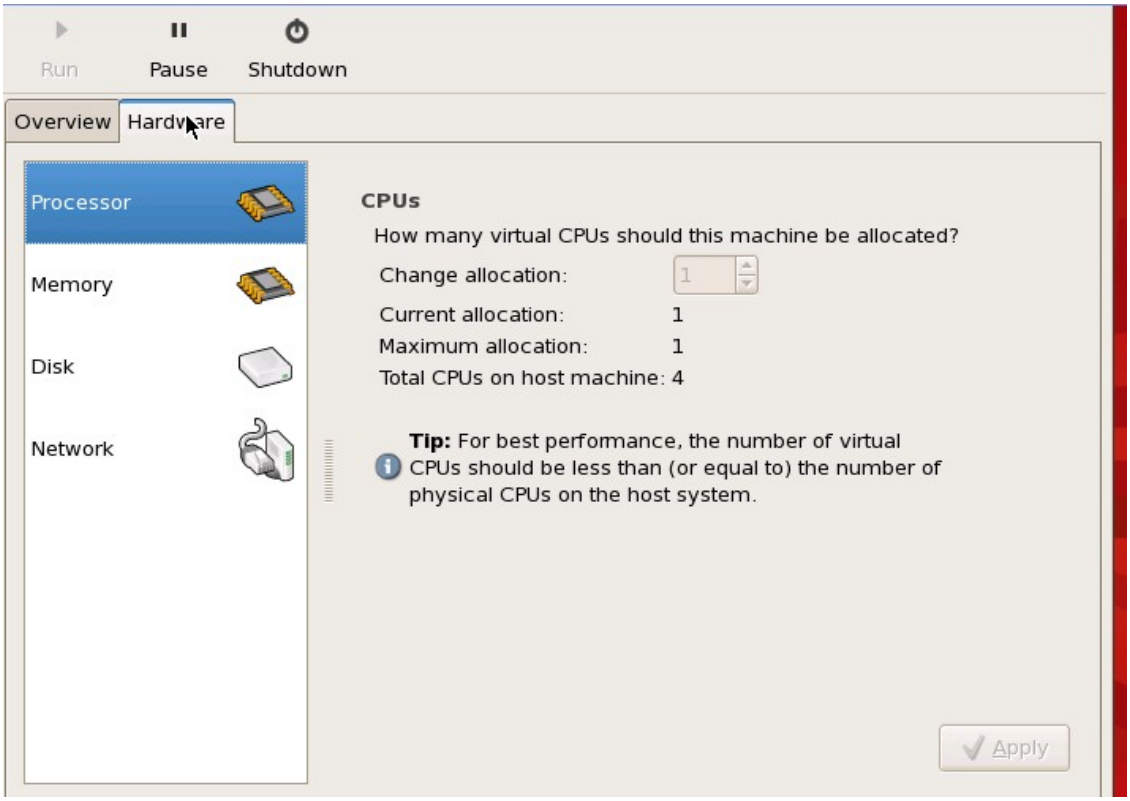
A) Virtual Machine Manager 기본 화면



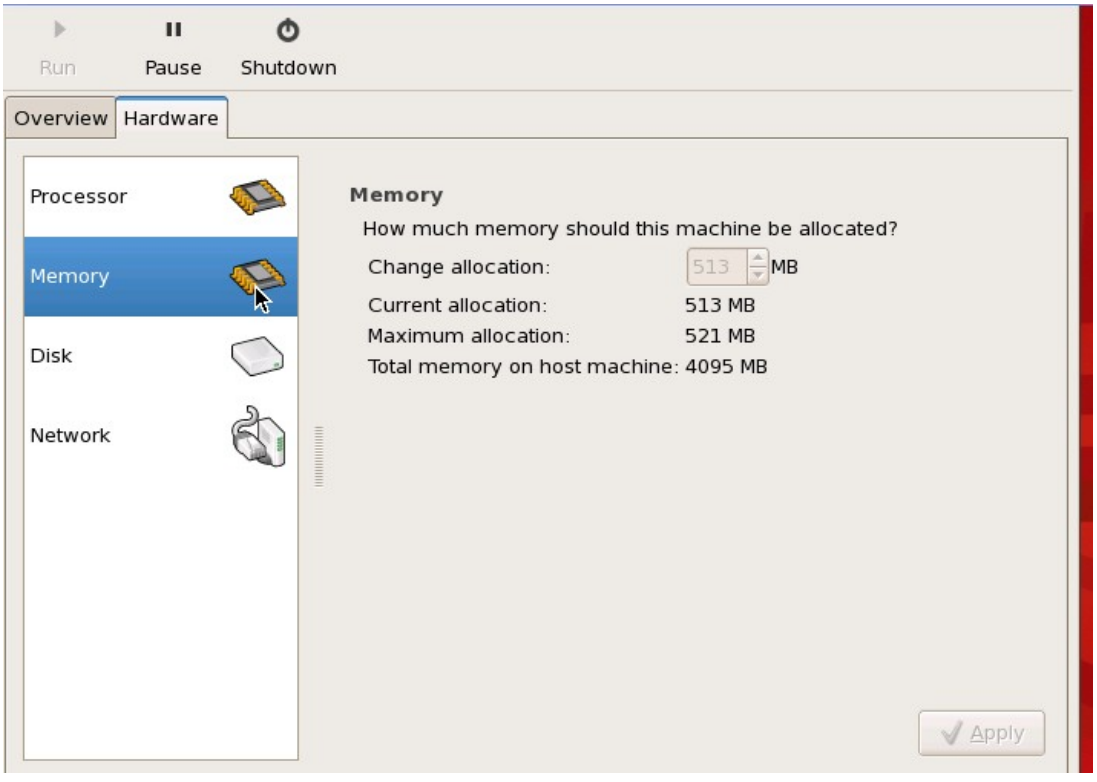
B) Virtual Machine Manager 기본 화면에서 해당 virtual system을 선택한 후에 아래 Details를 클릭하면 아래와 같은 관리창이 팝업된다. 상단 탭에서는 virtual system을 shutdown/Pause 할 수 있는 탭이 있다.



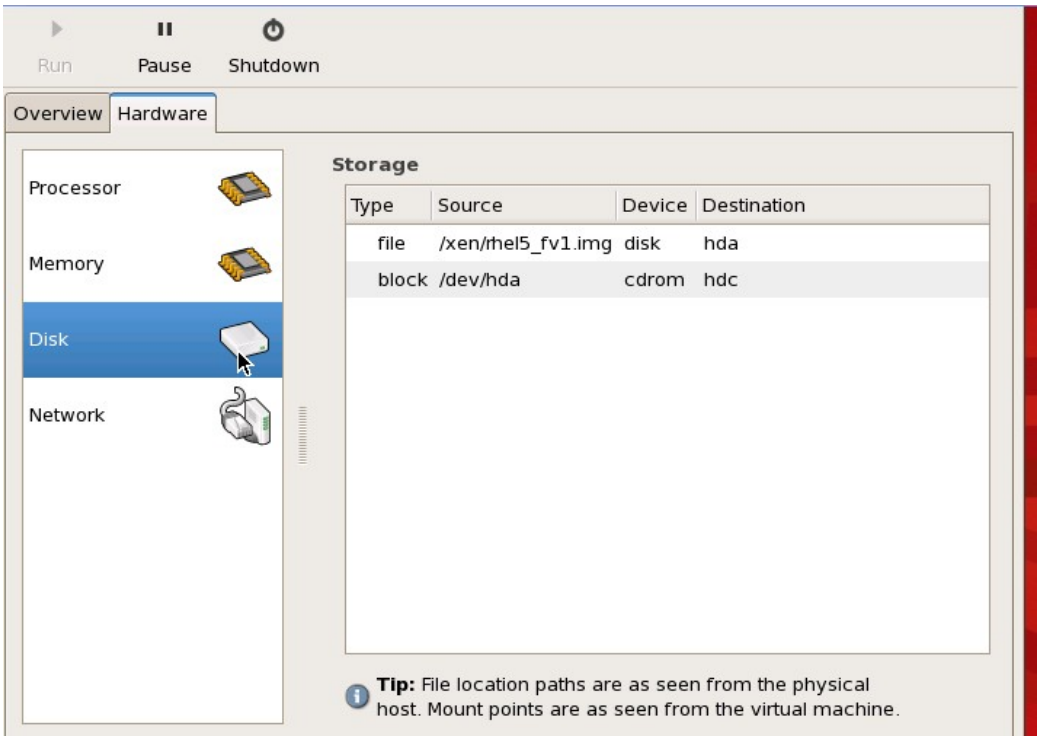
C) Hardware 탭을 선택하면 virtual system에 할당된 CPU, memory, disk, network 정보 확인 및 설정 변경을 할 수 있다.



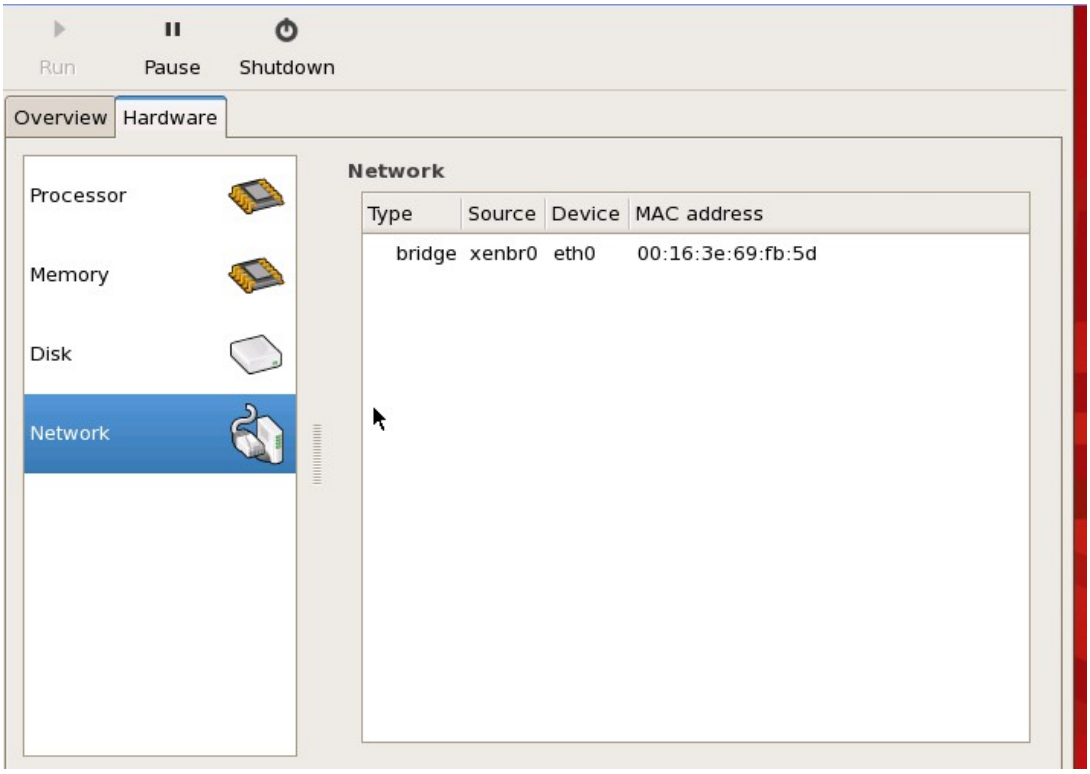
D) 메모리 관리 탭



E) 디스크 관리 탭



F) Network 관리탭



5.8 Command 기반 Management

A. xm create [virtual system name] : 이미 생성이 되어 있거나 생성할 virtual system을 시작한다.

B. xm console [virtual system name] : virtual system의 콘솔을 띄운다.(para-virtualized의 경우 유용함)

C. xm destory [virtual system name] : virtual system을 제거한다.

D. xm shutdown [virtual system name] : virtual system을 shutdown(full-virtualized의 경우는 hard poweroff를 한다)

E. xm top [virtual system name] : top과 비슷한 시스템 모니터링을 할 수 있다.

F. xm mem-set [virtual system name] [value] : para-virtualized guest의 메모리 사이즈를 조절한다.

G. xm vcpu-set [virtual system name] [value] : : para-virtualized guest의 CPU 개수를 조절한다.