

ISBN 978-89-6211-032-6 93560

## 슈퍼컴퓨터 4호기 사용자 지침서

최윤근, 장지훈, 성진우

## 목 차

1. 슈퍼컴퓨터 4호기 1차 시스템 대용량 시스템 개요	1
가. 3호기와 비교한 특징	1
나. 노드 구성	2
다. 4호기 1차 시스템 구성도	3
라. POWER 5+ 프로세서 구조	4
2. 사용자 환경	5
가. 로그인	5
나. 사용자 셸 변경	5
다. 홈 디렉터리	5
라. 스크래치	5
마. 데이터 아카이빙	6
바 제공시간	6
3. 프로그램 컴파일	6
가. 순차 컴파일러 및 병렬 컴파일 스크립트	6
나. 32bit/64bit 컴파일	7
다. 32bit 프로그램의 대형 메모리(2GB) 컴파일	8
라. 수학 라이브러리 사용법	8
4. 작업 실행 준비	9
가. 작업 클래스 확인	9
나. Job command file 작성	10
(1) serial 프로그램	12
(2) OpenMP 프로그램	13
(3) MPI 프로그램	14
(4) ybrid(OpenMP+MPI) 프로그램	16
5. Loadleveler를 통한 작업 실행	18
가. 작업 submit	18
나. 작업 상태 조회	18

다. 노드별 가용 CPU 알아보기	19
라. 클래스별 상태 조회	19
마. 작업 취소	20
6. Help Desk	20
<부록> 시스템 상세 사양	21

## 슈퍼컴퓨터 4호기 대용량 시스템 사용자 가이드

본 가이드는 KISTI 슈퍼컴 4호기 1차 시스템의 안정성 테스트를 위한 정책 및 가이드 이며 정식 유료 사용자 서비스 시 변경될 수 있습니다.

### 1. 슈퍼컴퓨터 4호기 1차 시스템 대용량 시스템 개요

#### 가. 3호기와 비교한 특징

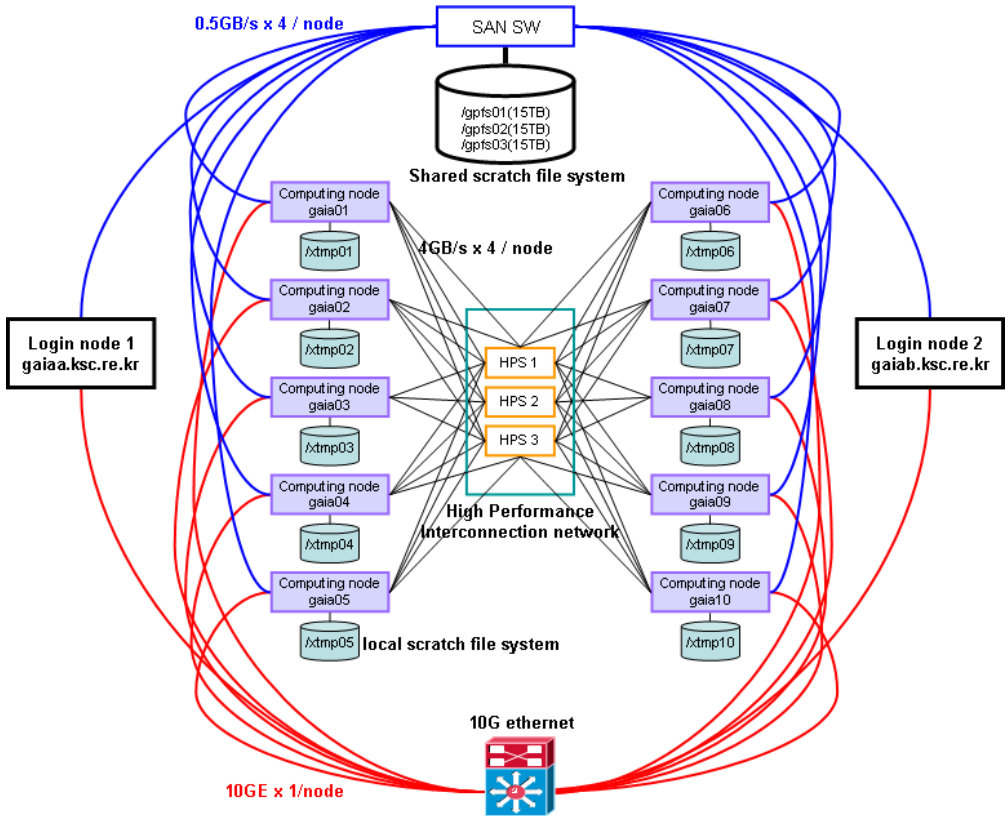
항목	3호기	4호기 1차
모델	p690	p595
노드 수	21	10
노드당 코어 수	32	64
프로세서	POWER4	POWER5+
Rpeak(GFlops)	5.2/코어	9.2/코어
유효 총 성능(TFlops)	4.3	5.9
총 메모리(TB)	4.4	2.8
디스크 타입/용량(TB)	SSA/98	FC SAN/60
Interconnection	SP	HPS

나. 노드 구성

KISTI 슈퍼컴퓨터 4호기 대용량 시스템은 총 10대의 컴퓨팅 시스템과 2대의 로그인 시스템, 그리고 1대의 디버깅 시스템으로 구성된다.

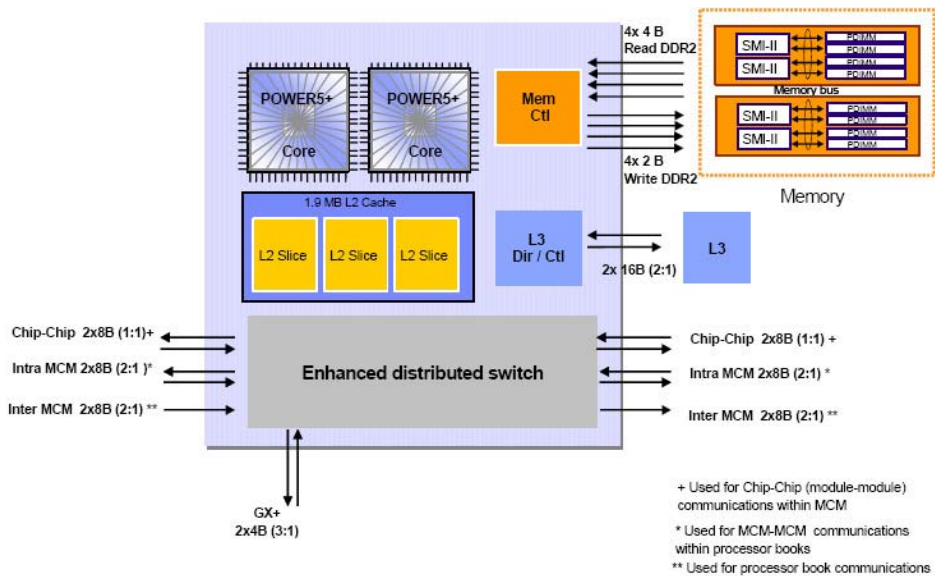
구 분		컴퓨팅 시스템	로그인 시스템	디버깅 시스템
시스템명		p5 595	p5 510Q	p575
수 량		<u>10대</u>	2대	1대
C P U	프로세서명	POWER5+ [2.3GHz]	POWER5+ [1.65GHz]	POWER5+ [1.9GHz]
	전체 이론 최고성능 (GFlops)	5,888 [588.8/노드]	8	121.6
	전체 개수(개)	<u>640</u> [64/노드]	6	16
주기억장치 용량(GB)		<u>2,816</u> [256x9, 512x1]	8	32
내장 디스크 용량(GB)		73.4GBx16 [1노드]	73.4GBx2[1노 드]	73.4GBx2
외장 디스크 용량(TB)*		<u>63</u>		

다. 4호기 1차 시스템 구성도



라. POWER5+ 프로세서 구조

구분	시스템 명	p5 595
프로세서 명		POWER5+
Clock Speed		2.3GHz
이론 최고 성능		9.2
단일 칩 당 내장 CPU 수		2개
Cache 구조	+L1 Cache	64/32KB(I/D) /core
	+L2 Cache	0.95MB/core
	+L3 Cache	18MB/core
기타		NUMA enabled



## 2. 사용자 환경

### 가 로그인

- 로그인은 로그인 노드로만 가능
- 컴퓨팅 노드로의 직접 액세스 불가능
- 액세스 인터페이스는 ssh, sftp만 허용
- 로그인 방법

```
$ ssh -i 사용자ID 150,183.146.21  
또는  
$ ssh -i 사용자ID 150,183.146.22
```

### 나. 사용자 셸 변경

- 기본 셸은 Korn shell(ksh)
- 셸 변경은 다음과 같이 한다

```
$ chsh
```

### 다. 홈 디렉터리

- 기본 쿼터는 3G
- 사용자 데이터 저장용
- 백업하지 않음

### 라. 스크래치

- /gpfs1(15TB) 병렬 파일시스템 사용 가능
- 사용량 제한 없음
- 최종 수정 후 24시간 경과 시 자동 삭제



마. 데이터 아카이빙

- 제공하지 않음
- 결과 데이터는 개별적으로 백업

바. 제공 시간

- 유저당 40,000 SRU
- $SRU = CPU\ time * SCF(4\text{-가 참조})$

### 3. 프로그램 컴파일

가. 순차 컴파일러 및 병렬 컴파일 스크립트

언어	Serial	OpenMP	MPI	OpenMP +MPI
C	xlc	xlc_r	mpcc	mpcc_r
C++	xlC	xlC_r	mpCC	mpCC_r
Fortran77	xlF	xlF_r	mpxlF	mpxlF_r
Fortran90	xlF90	xlF90_r	mpxlF90	mpxlF90_r
Fortran95	xlF95	xlF95_r	mpxlF95	mpxlF95_r

아래는 기본적으로 추천되는 컴파일 옵션이며, 이 옵션을 사용하여 Fortran 코드를 컴파일하는 예이다.

**Recommended Compilation Option : -O3 -qarch=pwr5  
-qtune=pwr5**

<Serial>

```
$ xlf -O3 -qarch=pwr5 -qtune=pwr5 -o sample_serial  
sample_serial.f
```

<OpenMP>

```
$ xlf_r -qsmp=omp -O3 -qarch=pwr5 -qtune=pwr5 -o  
sample_omp sample_omp.f
```

<MPI>

```
$ mpxlf -O3 -qarch=pwr5 -qtune=pwr5 -o sample_mpi  
sample_mpi.f
```

<OpenMP+MPI(hybrid)>

```
$ mpxlf_r -qsmp=omp -O3 -qarch=pwr5 -qtune=pwr5 -o  
sample_hybrid sample_hybrid.f
```

나. 32bit/64bit 컴파일

- 기본적으로 AIX용 컴파일러는 32bit 모드로 컴파일함.
- 원하는 bit 모드를 지정하려면 아래와 같이 bit 모드 옵션이나 환경변수를 사용.

컴파일러 옵션: -q32/-q64

OBJECT\_MODE 환경변수: 32/64

Archive(ar) 옵션: -X32/-X64/-X32\_64

- 사용자 프로그램이 2GB를 초과하는 크기의 메모리를 요구할 경우 64bit 컴파일 옵션(-q64)을 사용

```
$ xlf -q64 -O3 -qarch=pwr5 -qtune=pwr5 -c bt.f
```

다. 32bit 프로그램의 대형 메모리(2GB) 컴파일

- 256MB이상의 메모리(주로 heap 메모리)를 사용하려면 아래의 예와 같이 링크 옵션 **-bmaxdata**를 사용한다.

```
$ xlf -o large_mem_program -bmaxdata:0x80000000
large_mem_program
```

- **bmaxdata**의 단위는 바이트이며 크기는 십진수, 8진수(0으로 시작), 16진수(0x로 시작)로 나타낼 수 있는데, 위의 예는 8개의 데이터 세그먼트를 허용하여, 최대 2GB까지 heap 메모리를 사용할 수 있게 하는 것이다. 64bit 프로그램 모드에서는 이와 같은 256MB 세그먼트 한계가 없으나 **-bmaxdata**를 지정하여 그 값을 명시할 수도 있다.

라. 수학 라이브러리 사용법

- ESSL

ESSL 루틴을 포함한 프로그램의 컴파일은 다음과 같이 할 수 있으며 사용자는 필요한 옵션을 더 추가해 사용하면 된다.

< Serial 코드 >

```
$ xlf -O3 -qarch=pwr5 -qtune=pwr5 -lessl essl_exam.f
```

< SMP 코드 >

```
$ xlf_r -O3 -qarch=pwr5 -qtune=pwr5 -lesslsmp essl_exam.f
```

## - PESSL

PESSL 루틴을 포함한 프로그램의 컴파일은 다음과 같이 할 수 있으며 사용자는 필요한 옵션을 더 추가해 사용하면 된다.

```
< mpi 코드 >
$ mpixlf -O3 -qarch=pwr5 -qtune=pwr5 -lessl -lpeSSL -lblacs
essl_exam.f

< mpi+openmp 코드 >
$ mpixlf_r -O3 -qarch=pwr5 -qtune=pwr5 -lesslsmp -lpeSSLsmp
-lblacssmp essl_exam.f
```

## 4. 작업 실행 준비

**[주의]**  
작업 실행은 Loadleveler를 통해서만 실행 할 수 있으며 poe를 통한 인터랙티브 작업 수행은 허용하지 않음

### 가. 작업 클래스 확인

- \$ llclass 명령어로 확인 가능

ClassName	SCF(*)	Max Proc	Wall_clock_time	total slots	비고
normal	1.5	1	72hr	640	시리얼 작업
p_normal	1	640	72hr	640	일반 병렬 작업

(\*) Service Charging Factor

- 사용자별 작업 제한

서비스 Stanza	type	limit	비고
maxjobs	user	2	사용자당 동시 수행 가능한 작업 수
maxidle	user	2	사용자당 동시 대기 가능한 작업수
maxqueued	user	4	maxidle+maxjobs

나. Job command file 작성

LoadLeveler에서 배치 작업을 수행하기 위해서는 아래 예제와 같은 Job Command File을 작성해야 한다. 예제 파일은 /LoadL/ll\_job\_command\_file 디렉터리 참조

[주의]

Job Command File 작성 시 다음 사항에 유의하자

- Serial 및 OpenMP 프로그램인 경우, job\_type=serial 또는 생략
- 반드시 ConsumableCpus, ConsumableMemory 및 wall\_clock\_limit을 지정해야함.

- MPI 통신 네트워크 옵션:

Network.MPI=sn\_all,shared,US,,instances=2

- 노드 내 MPI 통신 옵션: MP\_SHARED\_MEMORY=yes

- OpenMP busy-wait threads 옵션:

XLSMPOPTS="spins=0:yields=0"

- MPI작업 및 Hybrid 작업을 위한 job command file 작성 시 기존 3호기와 차이점이 있으므로 유의해서 job command file을 작성

- Job command file의 변수 값을 얻기 위한 클래스별 상태 조회

○ 클래스별 최대 task 수 및 가용 task 수 알아보기

```
$ llclass
```

○ 클래스별 상세 정보

```
$ llclass -l 클래스명
```

○ 클래스별 wall\_clock\_limit 알아보기

```
$ llclass -l class_name | grep Wall
```

[주의]

테스트 기간 중 wall\_clock\_time 은 72시간입니다.

(1) Serial 프로그램

: /LoadL/ll\_job\_command\_files/ll\_serial.cmd

```
#!/bin/ksh
# @ job_type = serial
# @ class = normal
# @ error = serial.err
# @ output = serial.out
# @ notification = complete
# @ notify_user = my_email_address
# @ resources = ConsumableCpus(1)
ConsumableMemory(1500mb)
# @ wall_clock_limit=01:00:00
# @ queue
export MEMORY_AFFINITY=MCM
./my_serial_executable
```

## (2) OpenMP 프로그램

: /LoadL/ll\_job\_command\_files/ll\_openmp.cmd

ConsumableCpus(4)과 OMP\_NUM\_THREADS=4이 서로 일치해야 한다. OpenMP 쓰레드 개수는 아래의 예에서처럼 OpenMP runtime 변수인 OMP\_NUM\_THREADS로 지정하거나 IBM XL runtime 변수인 XLSMPOPTS="parthds="로 지정해 준다. XLSMPOPTS 환경변수에서 'yields=0:spins=0'은 쓰레드 수행 시 busy wait 상태를 유지하여 쓰레드가 sleep 상태로 빠지지 않게 하여 수행 성능은 좋아지지만 그만큼 전체 CPU 사용 시간은 증가한다. 프로그램 성능과 시스템 사용료 산정 기준을 고려하여 이 옵션을 적절히 사용해야 할 것이다.

```
#!/bin/ksh
# @ job_type = serial
# @ class = normal
# @ error = openmp.err
# @ output = openmp.out
# @ notification = complete
# @ notify_user = my_email_address
# @ resources = ConsumableCpus(4)
ConsumableMemory(1500mb)
# @ wall_clock_limit=01:00:00
# @ queue
export MEMORY_AFFINITY=MCM
export OMP_NUM_THREADS=4
./my_openmp_executable
```



### (3) MPI 프로그램

: /LoadL/ll\_job\_command\_files/ll\_mpi.cmd

MPI 프로그램을 수행하는 예로서, `job_type=parallel`로 지정하였다. `network.MPI` 키워드와 `node`, `tasks_per_node`와 같은 노드 할당 키워드를 지정하였다. `sn_all`은 IBM 시스템의 MPI 통신용 네트워크 종류이며, `resources` 옵션에서 `ConsumableCpus` 및 `ConsumableMemory`를 지정해야 한다.

```

#!/bin/ksh
# @ job_type = parallel
# @ class = p_normal
# @ error = mpi.err
# @ output = mpi.out
# @ notification = complete
# @ notify_user = my_email_address
# @ network.MPI = sn_all,shared,US,,instances=2
# @ rset = RSET_MCM_AFFINITY
# @ mcm_affinity_options = mcm_mem_pref mcm_sni_none
mcm_distribute
# @ resources = ConsumableCpus(1)
ConsumableMemory(1500mb)
# @ wall_clock_limit=01:00:00
# @ node = 2
# @ tasks_per_node = 32
# @ queue
export LANG=en_US
export MP_SHARED_MEMORY=yes
export MEMORY_AFFINITY=MCM
poe ./my_mpi_executable
#####
### KEYWORD for MPI TASKS
##@ total_tasks = 64
##@ blocking = unlimited
#####

```

(4) Hybrid(OpenMP+MPI) 프로그램

: /LoadL/ll\_job\_command\_files/ll\_hybrid.cmd

OMP\_NUM\_THREADS=4이므로 각각의 MPI 태스크는 4개의 쓰레드로 수행하겠다는 것이며, tasks\_per\_node=2이므로 노드당 2개의 태스크, 즉 노드당 총 8개의 쓰레드가 필요하게 된다. 또한 node=2이므로, 전체 태스크 개수는 4개가 되며, 전체 쓰레드 수는 16개가 된다.

```

#!/bin/ksh
# @ job_type = parallel
# @ class = p_normal
# @ error = hybrid.err
# @ output = hybrid.out
# @ notification = complete
# @ notify_user = my_email_address
# @ network.MPI = sn_all,shared,US,,instances=2
# @ rset = RSET_MCM_AFFINITY
# @ mcm_affinity_options = mcm_mem_pref mcm_sni_none
mcm_distribute
# @ resources = ConsumableCpus(4)
ConsumableMemory(1500mb)
# @ wall_clock_limit=01:00:00
# @ node = 2
# @ tasks_per_node = 2
# @ queue
export LANG=en_US
export MP_SHARED_MEMORY=yes
export MEMORY_AFFINITY=MCM
export OMP_NUM_THREADS=4
poe ./my_hybrid_executable
#####
### KEYWORD for MPI TASKS
##@ total_tasks = 4
##@ blocking = unlimited
#####

```

## 5. Loadleveler를 통한 작업 실행

### 가. 작업 submit

```
$ llsubmit job_command_file
```

job\_comman\_file은 보통 XXX.cmd 형태로 만들어짐

### 나. 작업 상태 조회

```
$ llq
```

보다 자세한 정보는

```
$ llq -s Job_ID
```

만일 예제에서와 같은 메시지가 나온다면 요청한 자원이 부족한 경우임

```
$ llq -s node01.184.0
=====EVALUATIONS   FOR   JOB   STEP   node01.184.0
=====
This job step has the following machine consumable resource
requirement:
Resource name = ConsumableCpus; Resource requirement per
task = 100; Minimum number of tasks = 1
The maximum values of consumable resource
"ConsumableCpus" on all machines: Total = 64; Available = 64
The amount of available resource is not adequate to satisfy the
requirement of this job step.
Not enough resources to start now.
Not enough resources for this step as top-dog.
.....
```

[주의]

- 위 예에서와 같이 **idle** 상태인 이유가 사용자가 요구한 **CPU** 자원을 제공할 수 있는 노드가 현재 없는 경우라면, 가용 **CPU** 자원이 확보될 때까지 대기하여야 한다.
- 단, **MPI** 작업인 경우에 일부 노드들의 가용 **CPU** 자원의 합이 요구 **CPU** 자원량 이상을 제공할 수 있다면, Job Submit File에 **#@ blocking = unlimited** 옵션을 추가하여 다수의 노드 간에 걸쳐서 **MPI** 작업을 수행할 수 있다. 물론 단일 노드 내에서보다는 작업 수행 성능이 저하될 수 있음

다. 노드별 가용 CPU 알아보기

```
$ llstatus -R
```

라. 클래스별 상태 조회

- 클래스별 최대 **task** 수 및 가용 **task** 수 알아보기

```
$ llclass
```

- 클래스별 상세 정보

```
$ llclass -l 클래스명
```

- 클래스별 **wall\_clock\_limit** 알아보기

```
$ llclass -l class_name | grep Wall
```

마. 작업 취소

\$ llcancel Job\_ID

## 6. Help Desk

테스트 기간 중 문제가 생기거나 의문사항이 있으시면 아래 연락처로 연락 해 주십시오.

이영주(KISTI, 슈퍼컴퓨팅센터), 042-869-0538, [yjlee@kisti.re.kr](mailto:yjlee@kisti.re.kr)  
임경빈(IBM, 시스템엔지니어), 042-869-0621, [kbim@kisti.re.kr](mailto:kbim@kisti.re.kr)

〈부록〉 시스템 상세 사양

1. 시스템 개요

컴퓨팅 노드			
구 분		1차	2차
시스템 명		p5 595	p6 H
아키텍처		SMP	SMP
프로세서 명		POWER5+	POWER6
이론 성능 (GFlops)	CPU(Core) 당	9.2	20
	노드 당	588.8	1,280
	시스템 전체	5,888	30,720
유효 총 성능(GFlops*yr)		29,440	107,520
노드 수(개)	케비넷(랙) 당	1	1
	전체	10	24
노드 당 소켓 수 (개)		32	32
케비넷[랙] 수 (개)		10	24
CPU[Core]수 (개)	소켓 당	2	2
	노드 당	64	64
	전체	640	1,536
메모리 용량 (GB)	노드 당	256 (9 노드) 512 (1 노드)	256 (18 노드) 512 (4 노드) 1024 (2



			노드)
	전체	2,816	8,704
내장디스크용량 (GB)	노드당	1,174.4 (73.4GB x 16)	1,174.4 (73.4GB x 16)
	전체	11,744	28,185.6
Interconnection 네트워크	방식	HPS	Infiniband 4X DDR
	포트 당 성능[단방향] (GB/sec)	2(4포트/서버)	2.5(4포트/서 버)
	Bisectional Bandwidth (GB/sec)	80	240
통합 네트워크 포트 수	10 GbE	10	24
	1 GbE	N/A	N/A
외장 디스크 용량(GB)		63,000	273,000
운영체제		AIX	AIX

## 2. 시스템 상세 사양

노드		
구분	1차	2차
모델명	p5 595	p6 H
프로세서 명	POWER5+	POWER6
Clock Speed(GHz)	2.3GHz	5GHz
이론최고성능(GFlops)	9.2	20.0
노드당 이론 최고성능(GFlops)	588.8	1,280.0
노드당 소켓수 (개)	32	32
소켓당 Core수 (개)	2	2
노드당 Core수 (개)	64	64
캐쉬 메모리 종류 및 크기	L1: 64/32KB(I/D) /core L2: 0.95MB / core L3: 18MB / core	L1: 64/64KB(I/D) /core L2: 8MB / core L3: 32MB / core
I/O 방식/슬롯수 (개)	PCI-X / 20 GX / 4	PCI-X / 20 GX / 4
메모리	노드 당 용량(GB)	256 (9 노드)
		256 (18 노드)

		512 (1 노드)	512 (4 노드) 1024 (2 노드)
	방식	DDR-2 533MHz	DDR-2 667MHz
	노드 당 대역폭(GB/sec)	799.5	1,200
	노드 당 용량(GB)	1,174.4 (73.4GB x 16)	1,174.4 (73.4GB x 16)
내장 디스크	RPM	10,000	10,000
	디스크 연결 방식	SCSI	SCSI
	RAID 방식	RAID-1/RAID-5	RAID-1/RAID-5
	Form Factor[WHD](mm)	785 x 2,025 x 1,326	785 x 2,025 x 1,326
<b>HMC</b>			
	구분	1차	2차
	모델명	7310-C05	7310-C06
	총 수량(개)	1	2
<b>Interconnection 네트워크 스위치</b>			
	구분	1차	2차
	방식	HPS	Infiniband 4X DDR

총 스위치 수량(대)	3	4
전체 포트 개수(개)	96	96
포트별 대역폭(GB/sec)	2(단방향)	2.5(단방향)
스위치 당 백플레인 성능(Tbps)	1	0.96
전송지연률(μsec)	3.9(MPI)	0.14
Bisectonal bandwidth (GB/sec)	80 (4 link x 10 노드 기준)	240 (4 link x 24 노드 기준)
특이사항	4 links / node	4 links / node
<b>관리 네트워크 스위치</b>		
구 분	1차	2차
방식	Gigabit ethernet	Gigabit ethernet
모델명	Cisco 3560G-48TS-S	-
총 스위치 수량(개)	2	-
스위치 당 GbE 포트 수(개)	1000BaseT: 48개 SFP: 4개	-
백플레인 성능(Gbps)	32 Gbps	-
전송지연률(μsec)	32	-

특이사항	관리 네트워크 전용, SFP는 SX type으로 총 4개 장착	별도의 스위치 증설 없 이 1차 시스템 스위치에 서 수용
<b>SAN 스위치</b>		
<b>구분</b>	<b>1차</b>	<b>2차</b>
방식	FC	좌동(1차 시스템 활용)
모델명	IBM 256B	좌동(1차 시스템 활용)
총 스위치 수량(개)	2	좌동(1차 시스템 활용)
스위치 당 포트 수(개)	160	좌동(1차 시스템 활용)
백플레인 성능(Gbps)	2.176 Tbps	좌동(1차 시스템 활용)
전송지연률(μsec)	2 이하	좌동(1차 시스템 활용)
<b>디스크 스토리지</b>		
<b>구분</b>	<b>1차</b>	<b>2차</b>
제조업체	Engenio	
모델명	DS4700 (1814-72A)	
수량(대)	3	13
물리 디스크 용량(GB)	86,400	374,400

가용 디스크 용량(GB)		63,000	273,000
RAID 구성 방식			
대당 컨트롤러 수(개)		2	RAID-5
포트 당 성능(Gbps)		4	4
호스트 인터페이스	컨트롤러 당 포트 수(개)	4	4
인터페이스 종류		FC-AL, FC-SW	
포트 당 성능(Gbps)		4	
드라이브 인터페이스	컨트롤러 당 포트 수(개)	2	2
인터페이스 종류		FC	
전체 수량(개)		15	65
대당 수량(개)		5	5
디스크 인클로저	인클로저 당 디스크 수(개)	16	16
유형		FC Disk	
디스크	단위 용량(GB)	300	

	RPM	10,000
이중화 지원 (OX)	컨트롤러	○
	캐시 메모리	○
	전원공급장치	○
	쿨링팬	○
	채널	○
핫 스왑지원		디스크, 컨트롤러, 파워, 팬, Mid-Plane, 배터리
디스크 통합관리 및 모니터링 도구 지원		SM(Storage Manager)
공유 파일 서비스 시스템		
구분		1차
파일시스템 명		GPFS
구성용량(TB)		63
서버 구성	시스템명/수량(대)	N/A
	프로세서명	N/A
	노드 당 CPU 수(개)	N/A
	노드 당 메모리 용량(GB)	N/A
		2차
		GPFS
		273
		N/A
		N/A
		N/A
		N/A

	Interconnection 방식/노드 당 포트 수(개)	N/A	N/A	
성능	최대 단일 파일 시스템 크기(TB)	이론 한계: 2 <sup>99</sup> 테스트 한계: 200		
	파일 시스템 당 최대 파일 수 (개)	2,147,483,647		
	Sustained aggregate throughput (GB/sec)	12	48	
	구간별대역폭 (GB/sec)	컴퓨팅노드 ↔ 스토리지 네트워크	20 (10노드x4링크x0.5GB/s)	48 (24노드x4링크x0.5GB/s)
		스토리지 네트워크 ↔ 디스크 스토리지	12 (3개x8링크x0.5GB/s)	52 (13개x8링크x0.5GB/s)
	파일 서버 ↔ 디스크 스토리지	N/A	N/A	



### 3. 공유 파일 서비스 시스템 구성도

