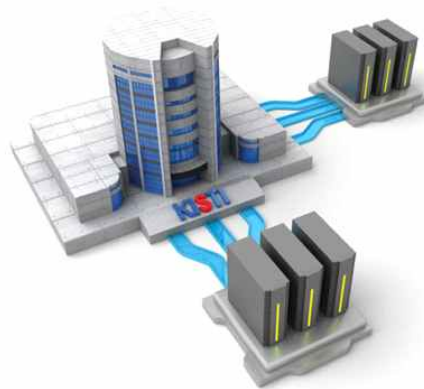




국가슈퍼컴퓨팅공동활용체제구축

PLSI 관리자 지침서

2014년 11월



- 목 차 -

1. PLSI 통합 시스템 구성	1
1.1 개요	1
1.2 연동 자원 현황	3
1.3 연동 네트워크	7
1.4 보안	8
2. PLSI Global File Sytem(PGFS)	10
2.1 GPFS 설치	10
2.2 GPFS 설정	10
2.3 GPFS Multi cluster 설정	12
2.4 GPFS 관리 명령어	13
2.5 GPFS 복구	14
2.6 GPFS 구성노드 추가	14
3. 글로벌 스케줄링(LoadLever-MC)	15
3.1 LoadLeveler 설치	15
3.2 LoadLeveler 설정	16
3.3 LoadLeveler Multicluster 설정	20
3.4 LoadLeveler 관리 명령어	20
3.5 Loadleveler account, llview 설정	21
3.6 Loadveler User 제한 설정	22
4. 통합 계정	24
4.1 통합 계정 시스템 구성	24
4.2 LDAP client 설정	26
4.3. 파트너기관 담당자의 계정 관련 업무 안내	30
5. 시간 동기화 설정	32
5.1 NTP client 설정	32
6. 통합 모니터링 시스템	34
6.1 모니터링 시스템 기능	34

- 6.2 Nagios 클라이언트 구성 36
- 7. 보안 설정 38**
 - 7.1 IP tables 설정 38
 - 7.2 TCP wrapper 설정 39

- 8. 기타 38**
 - 8.1 pdsh(ver 2.10-4) 38
 - 8.2 hostname 추가 39
 - 8.3 TCP window size 및 Jumbo Frame 설정 39
 - 8.4 CPU 사용률 정보 산출 40

1. PLSI 통합 시스템 구성

1.1 개요

PLSI는 Partnership and Leadership for the Supercomputer Infrastructure란 뜻으로 국가 과학기술 연구 개발의 핵심 도구인 슈퍼컴퓨터를 체계적인 연동과 관리를 통해 효율적으로 공동활용할 수 있는 체계를 구축하는 사업으로 국가 차원의 슈퍼컴퓨터 자원 활용의 극대화 및 슈퍼컴퓨팅 활성화와 인프라의 체계화를 목표로 2007년부터 수행하고 있다.

현재 슈퍼컴퓨팅 공동활용 및 활성화에 동참하고 있는 파트너 기관은 13개 기관이며, 해당 파트너 기관을 역할에 따라 자원센터(거점센터/연동센터) 및 응용센터로 구분하여 협력 체계를 구축하고 있으며, 주요 자원 배분 정책 및 사업 방향을 협의 결정하기 위해서 파트너 기관의 기관장을 위원으로하는 운영위원회를 구성하여 운영하고 있다.

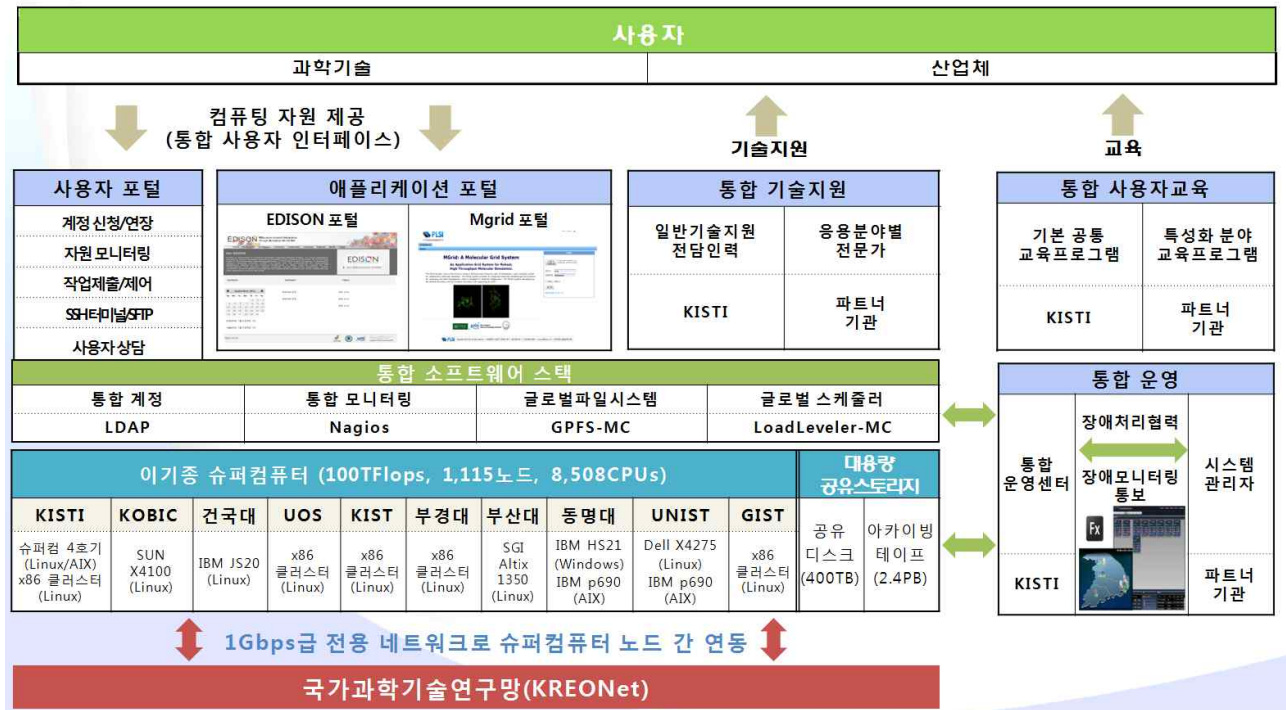
운영 위원회 (파트너 기관)	구분		역할	기관명	특성화분야-권역		협력 개시년도	KSCA (한국 슈퍼컴퓨터 협의회)
	중심센터 (Tier0)	거점센터 (Tier1)						
정책 · 사업 방향 임원 (위원장 부위원장 간사) 협의 · 의결 실무 운영팀	자원 센터	중심센터 (Tier0)	<ul style="list-style-type: none"> PLSI 사업 주관 통합 운영 및 서비스 환경 구축 통합 기술지원 체계 구축 통합 서비스 기술 개발 	KISTI	-	-	2007	18개 회원 기관 (산/학/연/관 슈퍼컴센터)
		거점센터 (Tier1)	<ul style="list-style-type: none"> 연동센터 역할 포함 사용자 커뮤니티 확대 산업체 기술혁신 지원 슈퍼컴 기술개발 보급 	부산대	기계부품/의료	부산권역	2007	
				부경대	대기/환경		2008	
				UNIST	인력양성	울산권역	2010	
				GIST	재료/뇌과학	광주.전남권역	2009	
				KOBIC	바이오	대전.충청권역	2010	
		연동센터 (Tier2)	<ul style="list-style-type: none"> 보유 슈퍼컴 자원의 PLSI 연동 연동 자원의 안정적인 운영 PLSI 사용자 기술 지원 	KIST	나노		2008	
				서울시립대	GEANT4	서울.경기권역	2010	
				건국대	분자시뮬레이션		2010	
		응용 센터	<ul style="list-style-type: none"> 사용자 커뮤니티 확대 산업체 기술혁신 지원 슈퍼컴 기술개발 보급 	KIAS	천문		2011	
				동명대	영화영상	부산권역	2007	
				전남대	분자전자	광주.전남권역	2011	
				부산TP	기계부품소재	부산권역	2007	
		기상청	기상		-	2007		

[슈퍼컴 자원 공동활용 및 활성화 협력체계]

사업 초기 3년 동안에는 국내 슈퍼컴퓨팅센터 간 파트너십 구성과 슈퍼컴퓨터 연동 서비스 인프라 구축에 주력하였고, 2012년 4월 현재 10개 기관 19개 슈퍼컴퓨터를 1Gbps 급 전용망(KREONET)으로 연결하고 통합 소프트웨어 스택을 설치하여 총 100TFlops(1,115노드, 8,508개 CPU cores)의 계산 성능을 제공하는 PLSI 통합 슈퍼컴퓨팅 서비스 체계를 구축했다.

연동된 시스템은 안정적인 서비스 제공을 위해 통합 모니터링 및 장애 대응 체계를 구축하고 있으며, 단순 자원 제공이 아닌 슈퍼컴 활용에 필요한 사용자 기술 지원 및 교육을 함께 제공하여 국내 연구자들이 슈퍼컴을 활용한 실질적인 연구 성과 창출에 기여하고 있다.

초보 사용자를 포함한 슈퍼컴퓨터 사용자들이 좀 더 쉽게 연동된 다양한 슈퍼컴퓨터를 사용하여 계산을 수행하고 자원 및 계정 정보를 파악하고 사용자 상담을 받을 수 있도록 기본 사용자 포털과 응용분야 별 애플리케이션 포털 서비스를 제공하고 있다.



[통합 슈퍼컴퓨팅 서비스 체계]

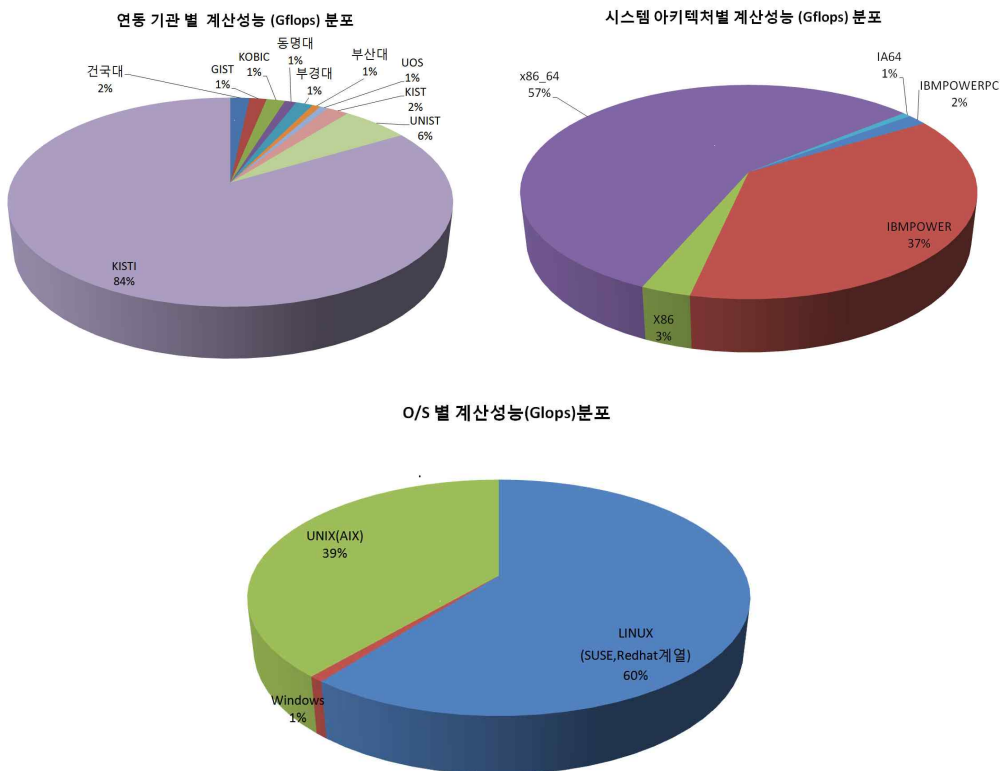
1.2 연동 자원 현황

PLSI 통합 시스템에서 연동 슈퍼컴 자원 현황을 살펴보면, 계산 성능기준으로 84% 정도를 KISTI 자원이 차지하고 있으며 시스템 아키텍처별로는 x86 계열의 시스템이 60%를 IBM POWER 프로세서 시스템이 37%의 성능분포를 보여주고 있다.

또한, O/S 별 계산성능 분포는 Linux 계열이 60%를 나머지 39%가 UNIX(AIX)를 탑재하고 있으며, Windows HPC Server를 설치한 시스템도 1% 정도 분포하고 있다.

계산 시스템 이외에도 연동 시스템 노드 간 효율적인 파일 공유 서비스 제공을 위한 약 394TB의 디스크 스토리지를 보유하고 있다.

더불어, 연동 시스템 별로 주요 특성화 분야에 적합한 응용 소프트웨어들을 보유하고 있다.



기관명	시스템명 (모델명)	프로세서	네트워크	OS	노드수	CPU수		노드 메모리 (GB)	외장 디스크 (TB)	계산 성능 (Gflops)
						전체	노드			
건국대	CI (IBM JS20)	IBM PowerPC 970 2.2GHz	Myrinet	SuSE ES9	91	182	2	2	1.6	1,601.6
GIST	KIGI (IBM x335)	Intel Xeon 2.8GHz	Myrinet	CentOS 5.3	128	256	2	3	5	1,433.6
KOBIC	KOBIC (SUN X4100)	AMD Opteron 2.1GHz	1GbE	CentOS 5.4	46	184	4	4	-	1,545.6
동명대	YONGMA (IBM BladeCenter HS21)	Intel Xeon 1.6GHz	1GbE	Windows HPCS2008	60	120	2	4	8	768
	TUSMP (IBM p690)	IBM POWER4 1.7GHz	1GbE	AIX 5.3	1	28	28	200	-	190
부경대	HAMEL (IBM x335)	Intel Xeon 2.8GHz	Myrinet	CentOS 5.3	128	256	2	3	5	1,433.6
부산대	PDAISY (SGI Altix 1350)	Intel Itanium2 1.5GHz	IB 4x SDR	RHEL AS 5.2	7	112	16	16	3.3	672
UOS	T2 (x86 클러스터)	Intel Xeon 2.13GHz	1GbE	SL linux 5.5	20	80	4	8	1	682
KIST	x86 클러스터	Intel Xeon 3.0GHz	Myrinet	CentOS 5.3	19	152	8	16	19	1,824
UNIST	UNIST_SMP (IBM p690)	IBM Power4 1.7GHz	1GbE	AIX 5.3	2	64	32	256/192	-	435
	CHEETAH (x86 클러스터)	Intel Xeon 2.53GHz	1GbE	CentOS 5.3	64	512	8	12	-	5,181
KISTI	PGAIA (IBM p595)	IBM POWER5 2.3GHz	HPS	AIX5.3	10	640	64	256/512	63	5,888
	PGAIA2 (IBM p595)	IBM POWER6 4.7GHz	IB 4x DDR	AIX6.1	24	1,536	64	256/512 /1,024	273	28,876
	PORION (IBM p575)	IBM POWER5 1.9GHz	HPS	AIX5.3	2	32	16	32	-	243
	TACHYON (SUN Blade 6048)	AMD Opteron 2.3GHz	IB 4x DDR	CentOS 4.6	192	3,072	16	32	200	28,262
	GLORY (SUN x2100)	AMD Opteron 1.8GHz	1GbE	CentOS 5.4	257	514	2	2	-	3,700.8
	BAEKDUSAN (Dell PE R410)	Intel Xeon 2.27GHz	1GbE	CentOS 5.5	14	112	8	16	-	1,017
	SOLBARAM (Intel Modular server)	Intel Xeon 2.5GHz	2GbE (bonding)	CentOS 5.5	18	144	8	24	-	1,440
	SINBARAM (Appro Green Blade)	Intel Xeon 2.6GHz (nVidia Tesla M2050 x 8ea)	IB 4x QDR	CentOS 6.0	32	512	16	64	-	14,769.6
	PGFS (글로벌공유파일 시스템-11노드)	Inter Xeon/ AMD Opteron	1/10GbE	CentOS 5.7	-	-	-	-	394	-
	SAM-QFS (SUN X4600 4노드)	AMD Opteron	10GbE	Solaris	-	-	-	-	2.4PB (Tape)	-
Total					1,115	8,508				99,962.8

[연동 시스템 리스트]

기관명	시스템명	컴파일러	병렬 라이브러리	수학 라이브러리	애플리케이션	설치경로
건국대	CI (IBM JS20)	XLF 10.1 XLC/C++ 8.0 GNU 3.3.3	MPICH-GM	ESSL 4.4 PESSL 3.3 FFTW 3.2 BLAS GotoBLAS etc.	CHARMM amber10 namd	/home01/applic/c30b1/exec/gnu/charmm /home01/applic/amber10 /home01/applic/namd/namd/2.7b1
GIST	KIGI (IBM x335)	Intel compiler 9.1 GNU 4.1.2	MPICH-GM	MKL FFTW 3.2, FFTW 2.1.5 netcdf 4.7 lapack 3.0-37	lammps(2009-07-07) lammps(2010-04-18) WRF 3.0.1 NAMD 2.8b1 GAMESS (2009-01-21 R3) ncarg VESTA java grads	/home01/applic/lammps/intel/ssh/ /home01/applic/lammps/lammps-18Apr10/intel/ssh/ /home01/applic/name/ 2.7b1/intel/ssh/ /home01/applic/gamess /home01/applic/ncarg /home01/VESTA-i686 /home01/java /home01/grads
KOBIC	KOBIC (SUN X4100)	Intel compiler 11 GNU 4.1.2	MPICH 1.2.7 openmpi	FFTW 2.1.5, FFTW 3.2.2	iperf java mpiblast namd siesta sxid	/opt/iperf /opt/java-1.6.0_22 /opt/mpiblast/1.6.0 /opt/namd/2.7b1 /opt/siesta-3.0-rc2 /opt/sxid
동명대	IBM BladeCenter HS21	Visual studio 2008	MS MPI		3D max V-ray	
	TUSMP (IBM p690)	"XCF 12.1 XLC/C++ 10.1 GNU 3.3.2"		"ESSL 4.2 PESSL 3.3"		
부경대	HAMEL (IBM x335)	Intel compiler 9.1 GNU 4.1.2	MPICH-GM	MKL FFTW 3.2, FFTW 2.1.5 netcdf 4.7 lapack 3.0-37 zlib	lammps(2009-07-07) lammps(2010-04-18) GME 2.17 WRF 3.0.1 NAMD 2.8b1 GAMESS (2009-01-21 R3) FCM grads java namd ncarg ncl nco ncview	/home01/applic/lammps/intel/ssh/ /home01/applic/lammps/lammps-18Apr10/intel/ssh/ /home01/applic/wrf/WRFV3 /home01/applic/namd/2.7b1/intel/ssh/ /home01/applic/gamess /home01/FCM /home01/grads /home01/java /home01/namd/2.7b1 /home01/ncarg /home01/ncl /home01/nco /home01/ncview
부산대	PDAISY (SGI Altix 1350)	Intel compiler 11 GNU 4.1.2	MVAPICH 1.1 openmpi	MKL	NAMD VMD 1.8.6 DL_POLY 2.20 siesta 3.0-b java VMD	/mnt/data/NAMD /mnt/data/VMD /mnt/data/dl_poly_2.20 /mnt/data/siesta-3.0-b /mnt/data/java-1.6.0_31 /mnt/data/VMD
UOS	x86 클러스터	Intel Compiler 10, 11 GNU 4.1.2	openmpi 1.4.3	FFTW 2.1.5 FFTW 3.2.2 MKL 10.2.5	NAMD siesta sxid	/opt/namd/2.7b1 /opt/siesta-3.0-rc2 /opt/sxid

[연동 시스템 별 보유 소프트웨어 현황-1]

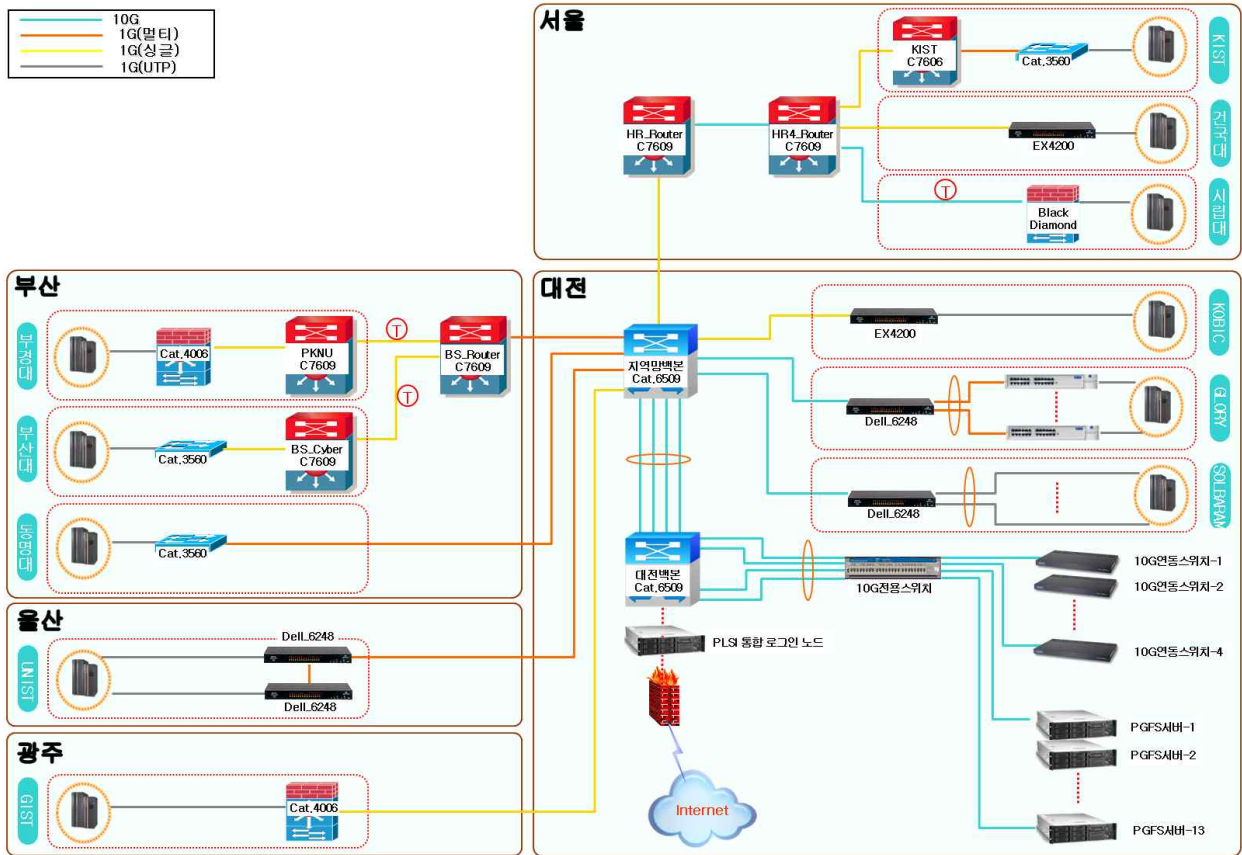
기관명	시스템명	컴파일러	병렬 라이브러리	수학 라이브러리	애플리케이션	설치경로
UNIST	UNIST_SMP (IBM p690)	XLF 12.1 XLC/C++ 10.1 GNU 3.3.2	PE 5.1.1.4	ESSL 4.4 PESSL 3.3	siesta gaussian09 gaussview	/applic/siesta-3.0-rc2 /applic/Applications/G09 /applic/Applications/gaussview
	CHEETAH (x86 클러스터)	Intel compiler 11 GNU 4.1.2	openmp i1.4.3	FFTW 2.1.5 FFTW 3.2.2 MKL 10.2.5 netcdf	NAMD 2.7b1 DL_poly siesta-3.0 lammmps iperf java OpenFOAM sxid WPS WRF	/opt/namd/2.7b1 /opt/dl_poly/intel /opt/siesta-3.0-rc2 /opt/lammmps /opt/iperf /opt/java-1.6.0_22 /opt/OpenFOAM /opt/sxid /opt/WPS /opt/WRFV3
KISTI	PGAIA (IBM p595)	XLF 10.1 XLC/C++ 8.0 GNU 4.2.0	PE 4.3.2.8	ESSL 4.2 PESSL 3.3	ABAQUS NASTRAN LS-Dyna FLUENT ANSYS AMBER STARCD CFX CHARMM Gaussian03 Gaussview Sysnoise SCTETRA PAMCRASH	abaqus : /applic/Applications/abaqus NASTRAN : /applic/Applications/msc/MD_NastranR2.1 LS-Dyna : /applic/Applications/LSDYNA Fluent : /applic/Applications/Fluent.Inc ANSYS : /applic/Applications/ANSYS AMBER : /applic/Applications/AMBER/amber10 STARCD : /applic/Applications/STARCD CFX : /applic/Applications/cfx CHARMM : /applic/Applications/charmm/c32b1 Gaussian 03 : /applic/Applications/g03 Gaussview : /applic/Applications/gaussview Sysnoise : /applic/Applications/sysnoise SCTETRA : /applic/Applications/Dcradle PAMCRASH : /applic/Applications/ESI
	PGAIA2 (IBM p595)	XLF 12.1 XLC/C++ 10.2 GNU 4.2.4	PE 4.3.2.8	ESSL 4.2 PESSL 3.3		
	PORION (IBM p575)	XLF 10.1 XLC/C++ 8.0 GNU 4.2.4	PE 4.3.2.8	ESSL 4.2 PESSL 3.3		
	TACHYON (SUN Blade 6048)	https://www.ksc.re.kr/ksc2012/02resource/02_resource.php 참조				
	GLORY (SUN x2100)	Intel Compiler 11 GNU 4.1.2	openmpi 1.4.3	FFTW 2.1.5 FFTW 3.2.2 MKL 10.2.5	java namd siesta	/opt/java-1.6.0_22 /opt/namd/2.7b1 /opt/siesta-3.0-rc2
	BAEKDUSAN (Dell PE R410)	Intel Compiler 11 GNU 4.1.2	openmpi 1.4.3	FFTW 2.1.5 FFTW 3.2.2 MKL 10.2.5	java namd siesta	/opt/java-1.6.0_22 /opt/namd/2.7b1 /opt/siesta-3.0-rc2
	SOLBARAM (Intel Modular server)	Intel Compiler 11 GNU 4.1.2	openmpi 1.4.3	FFTW 2.1.5 FFTW 3.2.2 MKL 10.2.5	MALAB java namd siesta	/phome01/applic/MATLAB/R2011b /opt/java-1.6.0_22 /opt/namd/2.7b1 /opt/siesta-3.0-rc2

[연동 시스템 별 보유 소프트웨어 현황-2]

1.3 연동 네트워크

PLSI 컴퓨팅 자원을 연결하는 연동 네트워크는 KISTI 내부 PLSI 백본 네트워크 장비를 중심으로 외부 연동 기관과는 KReonet(국가과학기술연구망)을 통해 1Gbps 수준의 전용 네트워크로 연결되어 있으며, KISTI 내부 주요 연동 시스템 및 파일 서버 등과는 10Gbps 이더넷으로 연결되어 있다.

특히, 다수의 연동 기관의 컴퓨팅 자원이 상호 연동되어 있기 때문에 외부로부터의 보안 위협을 최소화 하기 위해 연동 망 자체는 PLSI 통합 로그인 노드를 제외하고는 외부 네트워크와는 분리된 전용망 형태로 구성되어 있다.



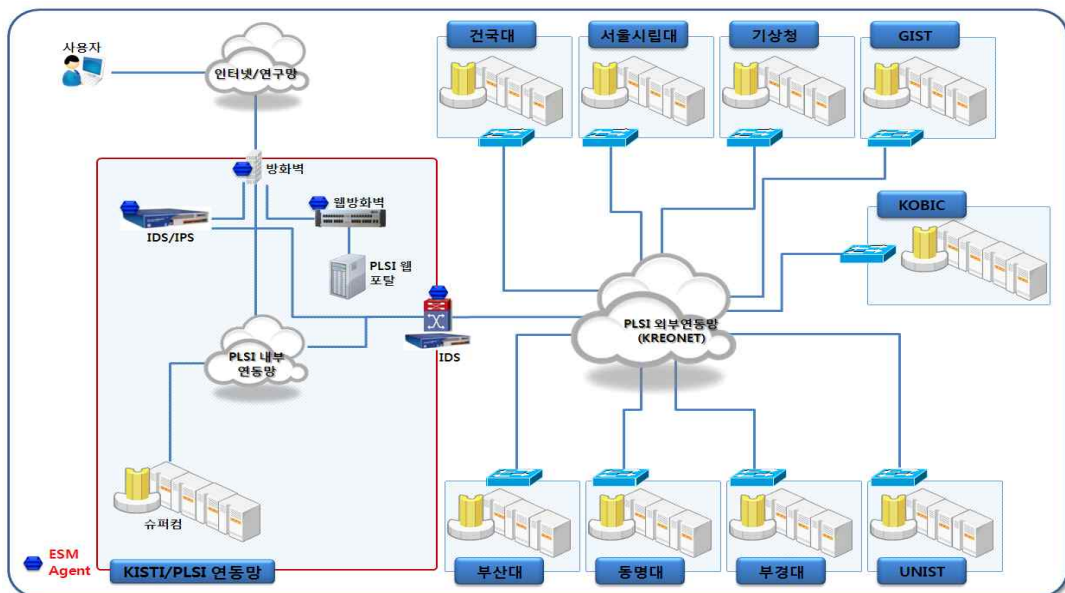
1.4 보안

1.4.1 PLSI 보안관리 수행체계

정보보안 정책/지침의 구성 체계는 모든 행정기관, 공공기관에 공통적으로 적용되는 법규 및 규정을 상위규정으로 준용하여 보안 관리를 위해 필요한 지침인 한국과학기술정보연구원 원내의 「보안 업무 규정」을 준수하도록 하고 있으며, PLSI 보안관리 수행체계도 정책/지침을 따르고 있다. 또한, PLSI 통합 서비스 환경에 대한 국정원 보안성 검토를 2009년도 하반기에 통과하였다.

1.4.2 PLSI 보안 정책

- PLSI 보안 구성도

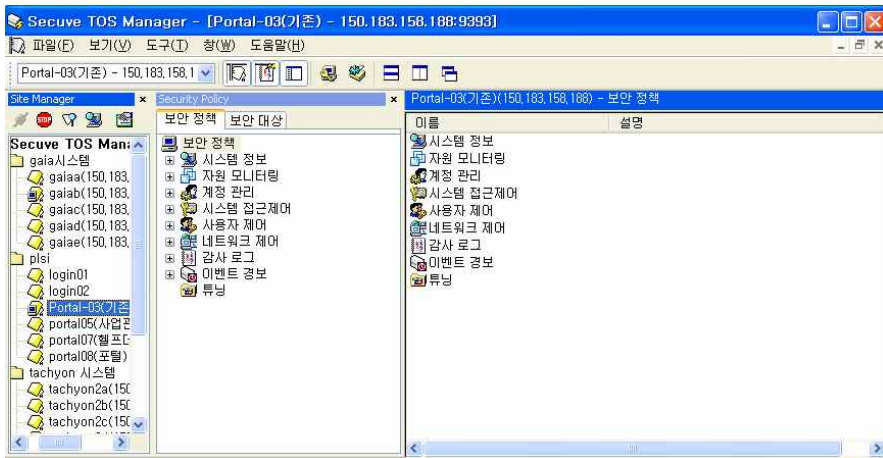


- ESM을 통해 IDS/IPS 등 장비 뿐만 아니라, 로그인서버, PLSI 포털들을 24시간 통합관리체계를 구축
- SecuveTOS를 통해 24시간 시스템 관리자 뿐 만 아니라, PLSI 사용자들이 파일시스템에 대한 불법 작업에 대한 차단 및 사용자감시를 함으로써 보안 관리 효율성 강화

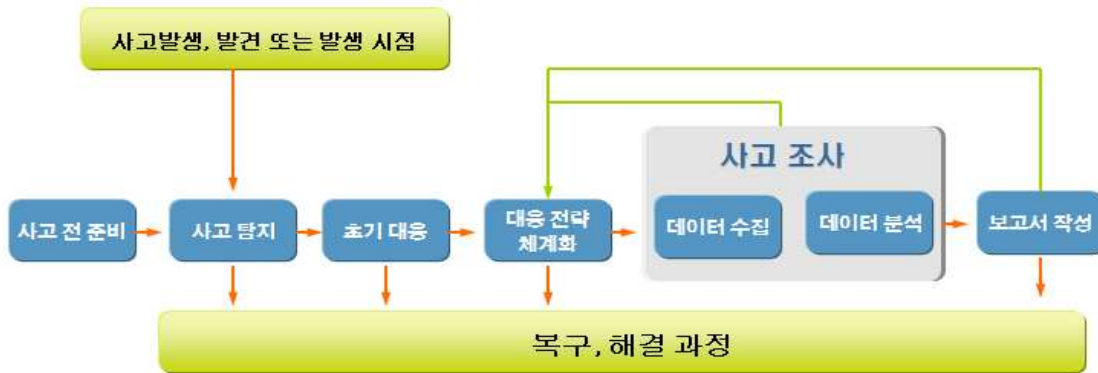
- 단계별 침해사고 대응체계 수립

- 시스템 접속 보안 체계 수립

- PLSI 통합 시스템 접근은 SSH 터미널을 사용하여 통합 로그인 노드를 통해서만 가능.
- IP레벨에서만 22번 포트 설정으로 보안성 강화.
- 취약점 점검 분기 1회 또는 긴급취약점 발견 시 수행.
- 통합 로그인 노드에 Secure TOS(서버 보안 S/W) 설치.



KiSTi 종합 관리

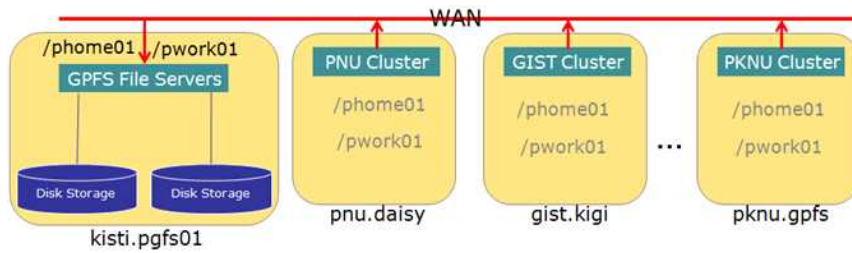


- : 전자서명기반의 사용자 인증 고려
- : 허가 받지 않은 사용자가 root 권한 획득 차단 및 관리자 경고
- : 보안 감사 로그 기록
- 일반 사용자는 연동 시스템들의 로그인/디버깅 노드를 제외한 컴퓨팅 노드등에 대한 접근 차단.
- 통합 로그인 노드에 대한 직접 root 로그인 허용 안함
- 주요 PLSI 파트너기관 보안 관리 현황 (iptables, fcheck 설치)
 - : 건국대, 서울시립대, KOBIC, UNIST, 부경대, GIST, 부산대
 - (예외)동명대(SMP), UNIST 일부분(SMP)라서 iptables 설치 못했음

- 기타 주요 보안 체계
 - 주기적인 호스트 취약점 분석 및 주요 파일 무결성 체크.
 - 사용자 홈 디렉터리 데이터 일일 백업 수행.
 - 각 연동 시스템에서 주요 로그를 통합 서버에 전송 통합 및 백업.
 - 내부 관리자 보안 교육 주기적 실시.

2. PLSI Global File System(PGFS)

IBM GPFS는 PLSI Global File System(PGFS)을 구성하는 병렬파일시스템으로써 이기종 연동 시스템 노드 간 안정적이고 효율적인 파일 데이터 공유 서비스에 활용되고 있다. 특히, GPFS-MC는 멀티클러스터 환경에서 리모트 클러스터에 구성된 GPFS 파일시스템을 상호 보안 인증을 통해 리모트 마운트할 수 있는 기능을 제공하고 있다.



2.1 GPFS 설치

2.1.1 GPFS base rpm 설치

```
#> rpm -ivh (gpfs 3.5.0-0 file).rpm
```

2.1.2 GPFS 패치 설치

- GPFS update download (ver. 3.5.0-X x86_64)

http://www-933.ibm.com/support/fixcentral/swg/selectFixes?parent=Cluster%2Bsoftware&product=ibm/power/IBM+General+Parallel+File+System&release=3.5.0&platform=Linux+64-bit,x86_64&function=all

```
#> rpm -Uvh (gpfs 3.5.0-X file).rpm
```

2.2 GPFS 설정

2.2.1 ~/.bash_profile GPFS PATH 추가

```
PATH=$PATH:/usr/lpp/mmfs/bin
MANPATH=$manpath:/usr/lpp/mmfs/messages
```

2.2.2 Portable Layer build

```
#> cd /usr/lpp/mmfs/src
#> make LINUX_DISTRIBUTION=REDHAT_AS_LINUX Autoconfig (redhat 계열 linux)
#> make World
#> make InstallImages
```

2.2.3 다수 노드로 구성된 클러스터에서는 Portable layer를 rpm으로 build

- ```
#> make rpm (Portable layer build 생성한 폴더에서 실행)
```
- 완료되면 /usr/src/redhat/RPMS/x86\_64/ 폴더에 rpm 파일이 생성된다. (32bit 는 i386폴더)

### 2.2.4 GPFS Cluster 구성

```
#> vi host_nodes.conf
```

```
host01:manager-quorum
host02:manager-quorum
host03:manager-quorum
host04:nonquorum-client
host05:nonquorum-client
host06:nonquorum-client
```

※ manager (파일시스템 관리역할), quorum (클러스터 구성정보 노드),  
client (GPFS 볼륨에 접근할수 있는 노드)

※ #> man mmchnode 참조

### 2.2.5 Cluster 생성

```
#> mmcrcluster -n host_nodes.conf -p host01 -s host02 -r /usr/bin/ssh -R /usr/bin/scp -C kisti.host
```

※ Option 설명

- n Nodes 구성파일
- p cluster 의 primary node 지정
- s cluster 의 secondary node 지정
- r remote shell 지정 (rsh or ssh 사용가능)
- R remote copy 지정 (rcp or scp 사용가능)
- C cluster 이름 지정

### 2.2.6 Cluster 생성 확인

```
#> mmlscluster
```

### 2.2.7 GPFS License 등록

- license 는 manager(manager-quorum)와 client (nonquorum-client) 를 구분하여 설정한다.

```
#> mmchlicense server --accept -N host01,host02,host03
```

```
#> mmchlicense client --accept -N host04,host05,host06
```

## 2.3 GPFS Multi cluster 설정

먼저, openssl 관련 라이브러리가 설치되어 있어야 하며, 아래 명령어들은 gpfs primary 노드에서 실행해야 한다. KISTI의 공유파일서버 GPFS 클러스터의 RSA key는 이미 생성이 되어 있으므로, 새로 구성한 GPFS cluster에서 key를 생성한다.

### 2.3.1 key 생성

```
#> mmauth genkey new
```

- GPFS 클러스터 간 상호 인증을 위해 RSA key를 생성한다.
- RSA key는 /var/mmfs/ssl 디렉토리에 저장된다.

### 2.3.2 Security configuration update

```
#> mmchconfig cipherList=AUTHONLY
```

### 2.3.3 생성된 pubic key(id\_rsa.pub)를 KISTI pgfs03 서버의 RSA public key와 교환

public key는 리모트 cluster를 구분할 수 있도록 GPFS 클러스터 이름을 붙여 저장한다.

ex) id\_rsa.pub-kisti.host or kisti.host-id\_rsa.pub

### 2.3.4 KISTI의 pgfs03 서버에서 리모트 클러스터(kisti.host) 정보 추가

```
pgfs03 #> mmauth add kisti.host -k id_rsa.pub-kisti.host
```

- kisti.host : host01에서 mmlscluster 명령으로 확인 (GPFS cluster name 항목)
- id\_rsa.pub-kisti.host : host01에서 mmauth genkey new 명령으로 생성된 key

### 2.3.5 kisti.glory(리모트) 클러스터가 글로벌 홈 및 스크래치 파일시스템을 마운트 할 수 있도록 KISTI의 pgfs03 서버에서 권한 부여

```
pgfs03 #> mmauth grant kisti.host -f /dev/phome01
```

```
pgfs03 #> mmauth grant kisti.host -f /dev/pwork01
```

```
pgfs03 #> mmauth show kisti.glory
```

### 2.3.6 kisti.glory 클러스터에서 kisti.pgfs01 cluster 추가

```
glory-mg01 #> mmremotecluster add kisti.pgfs01 -n pgfs03,pgfs04,pgfs05,pgfs06,
```

```
pgfs07,pgfs08,pgfs09,pgfs10,pgfs11,pgfs12,pgfs13 -k id_rsa.pub-kisti.pgfs01
```

- kisti.pgfs01 : kisti의 GPFS File server의 cluster 이름
- pgfs03, ~ ,pgfs10 : kisti의 File server의 host 명
- id\_rsa.pub-kisti.pgfs01 : kisti의 pgfs01에서 생성된 key

### 2.3.7 kisti.host 클러스터 노드에서 /phome01과 /pwork01을 리모트 마운트 할 수 있도록 설정

```
#> mmremotefs add /dev/phome01 -f /dev/phome01 -C kisti.pgfs01 -T /phome01
#> mmremotefs add /dev/pwork01 -f /dev/pwork01 -C kisti.pgfs01 -T /pwork01
```

### 2.3.8 kisti.host에서 /phome01, /pwork01 mount

```
#> mmstartup [-a : 전체 node]
#> mmmount /phome01 [-a : 전체 node]
#> mmmount /pwork01 [-a : 전체 node]
```

## 2.4 GPFS 관리 명령어

### 2.4.1 GPFS 데몬 시작

```
#> mmstartup -a
- 전체 node의 GPFS 데몬을 일괄적으로 시작할 때 사용한다.
#> mmstartup -N [node01,node02.....]
- 각각의 해당 node의 GPFS 데몬을 선택적으로 시작할 때 사용한다.
```

### 2.4.2 GPFS 데몬 종료

```
#> mmshutdown -a
- 전체 node의 GPFS 데몬을 일괄적으로 종료할 때 사용한다.
mmshutdown -N [node01,node02.....]
- 각각의 해당 node의 GPFS 데몬을 선택적으로 종료할 때 사용한다.
```

### 2.4.3 GPFS File system mount

```
#> mmmount /phome01 -a
- 전체 node에 /phome01 을 일괄적으로 mount 할 때 사용한다.
#> mmmount /phome01 -N [node01,node02.....]
- 각각의 해당 node에 /phome01 을 선택적으로 mount 할 때 사용한다.
```

### 2.4.4 GPFS File system umount

```
#> mmumount /phome01 -a
- 전체 node의 /phome01을 일괄적으로 umount 할때 사용한다.
#> mmumount /phome01 -N [node01,node02.....]
- 각각의 해당 node에 /phome01 을 선택적으로 umount 할 때 사용한다.
```



## 2.5 GPFS 복구

기존 GPFS 클러스터 노드가 시스템 장애로 인하여 O/S를 재설치 하였거나 혹은 사용자 실수로 GPFS가 삭제 되었을 경우 Primary/Secondary configuration server의 GPFS 구성 정보 정보로 복구할 수 있다.

### 2.5.1 GPFS 구성 정보 복구

사전에 GPFS가 파일셋이 설치되고 portable layer도 build 되어 있어야 한다.

아래 예제는 host01 노드를 host02(Primary/Secondary configuration server)의 GPFS 구성정보 (mmsdrfs)로부터 복구한다고 가정

```
host01#> mmsdrrestore -p host02 -F /var/mmfs/gen/mmsdrfs -R /usr/bin/scp
```

### 2.5.2 GPFS 구성정보 동기화

```
#> mmchcluster -p LATEST
```

### 2.5.3 GPFS 데몬 실행 및 /phome01, /pwork01 mount

```
#> mmstartup
```

```
#> mmmount /phome01
```

```
#> mmmount /pwork01
```

## 2.6 GPFS 구성 노드 추가

기존 GPFS 클러스터 가 구성된 곳에 노드를 추가하게 되어 GPFS 파일시스템을 설정하기 위해서 mmaddnode 를 이용하여 GPFS cluster list 를 추가 할수 있다.

### 2.6.1 GPFS cluster list 에 client 로 추가

```
#> mmaddnode -N host01
```

### 2.6.2 GPFS cluster list 에 quorum-manager 로 추가

- quorum-nanager node 추가시에는 GPFS 데몬을 모두 정지후 설정해야 한다.

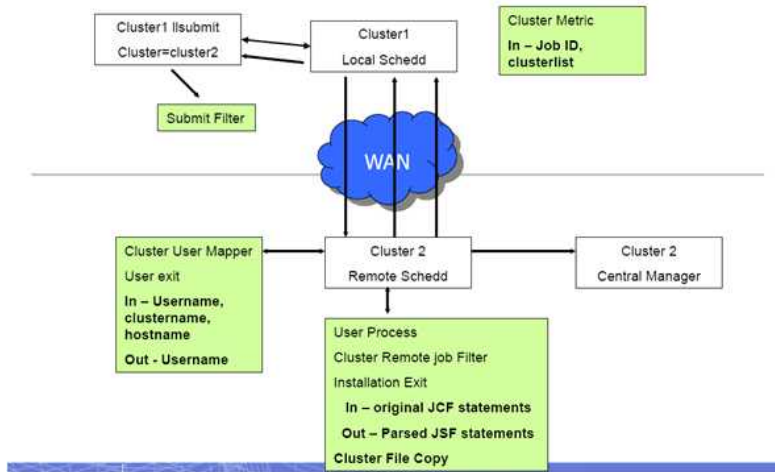
```
#> mmaddnode -N host01:quorum-nanager
```

### 2.6.3 GPFS license 추가

- "2.2.7 GPFS License 등록" 을 참조하여 설정한다.

### 3. 글로벌 스케줄링(LoadLever-MC)

IBM LoadLeveler는 AIX 기반의 IBM pSeries 시스템 및 리눅스 클러스터에서 사용되고 있는 배치 스케줄러로서, 최근 유럽의 DEISA와 같은 이기종 멀티 클러스터 환경에서 글로벌 스케줄링 서비스에 활용되고 있다. PLSI에서는 LoadLeveler-MC(멀티클러스터) 환경을 구축해서 각 Loadleveler 클러스터에 설정된 게이트웨이 노드를 통해 리모트 클러스터 간 작업 제출, 모니터링, 제어 등의 통합된 작업 스케줄링이 가능하게 되었다.



#### 3.1 LoadLeveler 설치

##### 3.1.1 Loadleveler base rpm 설치

- license 파일 설치

```
#> rpm -ivh (LoadL-full-license rpm file).rpm
```

- LoadL-full , IBMJava2 설치

```
#> cd /opt/ibmll/LoadL/sbin
```

```
#> ./install_ll -y -d /root/install_package/loadleveler/4.1.1.0 (LoadLeveler RPM 파일 위치)
```

※ CentOS 5.X 기준으로 LoadLeveler 4.1.1 설치 기준으로 작성합니다.

##### 3.1.2 update

```
http://www-933.ibm.com/support/fixcentral/swg/selectFixes?parent=Cluster%2Bsoftware&product=ibm/power/IBM+Tivoli+Workload+Scheduler+LoadLeveler&release=4.1.1&platform=Linux+64-bit,x86_64&function=all
```

- Loadleveler update download (ver. 4.1.1.14 x86\_64)

```
#> rpm -Uvh (loadleveler rpm file).rpm
```

### 3.1.3 loadl user 생성

```
#> useradd -d /LoadL loadl
```

### 3.1.4 loadl 홈디렉토리에 local 디렉토리 생성

```
#> mkdir /LoadL/local
```

### 3.1.5 local 디렉토리에 master node directory 생성 및 소유권 변경

```
#> mkdir /LoadL/local/host01
#> chown -R loadl.loadl local
```

### 3.1.6 초기화 script 실행

```
#> su -l loadl
#> cd /opt/ibmll/LoadL/full/bin
#> ./linit -local /LoadL/local/host01 -release /opt/ibmll/LoadL/full -cm host01
```

## 3.2 LoadLeveler 설정

### 3.2.1 Cluster 구성을 위한 loadl 계정공유를 위해 NFS 서버 설정

- master node (host01)에서 nfs 서버 설정을 한다.

```
#> vi /etc/exports
```

```
/LoadL host*(rw,no_root_squash,no_all_squash,no_subtree_check,async)
```

- nfs 데몬 설정

```
#> chkconfig --level 35 nfs on
#> chkconfig --level 35 portmap on
- nfs start
#> service nfs start
```

### 3.2.2 Computing node에서 host01 /LoadL 디렉토리 mount 설정

- 각 node(host01, host02, host03 ... host10) 에 /LoadL 디렉토리 생성

```
#> mkdir /LoadL
```

- 각 node /LoadL 디렉토리를 host01 의 /LoadL 디렉토리로 mount 한다.

```
#> mount -t nfs host01:/LoadL /LoadL
```

- Computing node에서 부팅할때마다 /LoadL 디렉토리를 host01:/LoadL 로 마운트 할 수 있도록 /etc/fstab 설정을 하여 운영하는 것이 편리하다.

```
#> vi /etc/fstab
host01:/LoadL /LoadL nfs rw,bg,intr,hard,rsize=32768,wsiz=32768 0 0
```

### 3.2.3 LoadL\_admin 설정

```
#> vi ~loadl/LoadL_admin (host01 ~ host10 설정)
MACHINE STANZAS:
host01: type = machine
 central_manager = true
 schedd_host = true
 resources = ConsumableCpus(4) ConsumableMemory(4gb)

host02: type = machine
 central_manager = alt
 resources = ConsumableCpus(4) ConsumableMemory(4gb)

host03: type = machine
 resources = ConsumableCpus(4) ConsumableMemory(4gb)

CLUSTER STANZAS:
kisti.host: type = cluster
 outbound_hosts = host01
 inbound_hosts = host01
 multicluster_security = SSL
 local = true

kisti.login: type = cluster
 outbound_hosts = login01
 inbound_hosts = login01
 multicluster_security = SSL

kisti.portal: type = cluster
 outbound_hosts = portal01
 inbound_hosts = portal01
 multicluster_security = SSL
```

## 3.2.4 LoadL\_config 설정

```

#> vi ~loadl/LoadL_config
line 9: LOADL_ADMIN = loadl loadladm
=> LOADL_ADMIN = loadl loadladm root

line 34: # CENTRAL_MANAGER_HEARTBEAT_INTERVAL = 300
 # CENTRAL_MANAGER_TIMEOUT = 6
=> CENTRAL_MANAGER_HEARTBEAT_INTERVAL = 100
 CENTRAL_MANAGER_TIMEOUT = 6

line 40: RELEASEDIR = $(tilde)
 LOCAL_CONFIG = $(tilde)/LoadL_config.local
 ADMIN_FILE = $(tilde)/LoadL_admin
 LOG = $(tilde)/log
 SPOOL = $(tilde)/spool
 EXECUTE = $(tilde)/execute
 HISTORY = $(SPOOL)/history

=>
RELEASEDIR = /opt/ibmll/LoadL/full
LOCAL_CONFIG = $(tilde)/local/$(host)/LoadL_config.local
ADMIN_FILE = $(tilde)/LoadL_admin
LOG = $(tilde)/local/$(host)/log
SPOOL = $(tilde)/local/$(host)/spool
EXECUTE = $(tilde)/local/$(host)/execute
HISTORY = $(SPOOL)/history

line 75: ACCT = A_OFF A_RES
=> ACCT = A_ON A_DETAIL

line 77: GLOBAL_HISTORY = $(SPOOL)
=> GLOBAL_HISTORY = /LoadL/acct/GLOBAL_HISTORY

line 219: #ENFORCE_RESOURCE_USAGE = ConsumableCpus
=> ENFORCE_RESOURCE_USAGE = ConsumableCpus

line 225: #ENFORCE_RESOURCE_SUBMISSION = true
=> ENFORCE_RESOURCE_SUBMISSION = true

line 637: LL_RSH_COMMAND = /usr/bin/ssh

```

### 3.2.5 LoadL\_config.local 수정

```
#> vi ~loadl/local/host01/LoadL_config.local
START_DAEMONS = TRUE
MAX_STARTERS = 4
CLASS = normal(4)
```

### 3.2.6 ~loadl/local/host01 복사

- 각 노드의 hostname 으로 노드 숫자만큼 복사를 해준다.

```
#> cp -r ~loadl/local/host01 ~/loadl/local/host02 (host03, host04.... host10)
```

### 3.2.7 ~loadl/local에 hostname으로 복사한 폴더의 소유권과 권한을 변경한다.

- hostname으로 지정된 폴더와 하위 폴더의 소유권을 loadl, 권한을 775로 변경한다.

```
#> chown -R loadl.loadl host02 (host02 하위 폴더까지 소유권을 loadl로 변경)
```

```
#> chmod 775 host02 (host02 폴더의 권한을 775 로 변경)
```

- 각 hostname 하위 폴더의 권한 변경 (execute : 1777 , log: 775, spool: 700)

### 3.2.8 LoadL.cfg 설정 변경 및 복사

- ~loadl/samples/LoadL.cfg 수정

```
#> vi LoadL.cfg
```

```
LoadLUserid = loadl
LoadLGroupid = loadl
LoadLConfig = /LoadL/LoadL_config (LoadL_config 위치로 변경)
```

- 수정 완료후 /etc 폴더로 복사 (전체 노드로 복사해준다)

## 3.3 LoadLeveler Multicluster 설정

### 3.3.1 llclusterauth -k 로 /var/opt/LoadL/ssl 에 key 생성

```
#> llclusterauth -k
```

- /var/opt/LoadL/ssl/id\_rsa.pub 생성

### 3.3.2 multicluster를 위한 key 교환

- KISTI의 login01서버와의 multicluster 구성

3.3.2.1 host01 의 id\_rsa.pub key를 login01 서버의 /var/opt/LoadL/ssl/authorized\_keys/id\_rsa.pub.host01 로 복사한다.

3.3.2.2 같은 방법으로 login01 서버의 key 를 host01의 /var/opt/LoadL/ssl/authorized\_keys/id\_rsa.pub.login 로 복사한다.

### 3.3.3 key를 교환하였으면 loadleveler를 시작(재시작) 한다. (login01, host01 모두)

```
#> llctl -g start (recycle)
```

### 3.3.4 multi cluster 설정 확인

- multi cluster를 확인하려면 해당하는 서버의 /etc/hosts 에 서로의 hostname 등록이 되어 있어야 한다.

```
#> llq -X kisti.login , llq -X kisti.host
```

## 3.4 LoadLeveler 관리 명령어

### 3.4.1 Loadleveler 데몬 시작

```
#> llctl -g start
```

- 전체 node의 loadleveler를 일괄적으로 시작할 때 사용한다.

```
#> llctl -h [hostname] start
```

- 각각의 해당 node의 loadleveler 데몬을 선택적으로 시작할 때 사용한다.

### 3.4.2 Loadleveler 데몬 정지

```
#> llctl -g stop
```

- 전체 node의 loadleveler를 일괄적으로 정지할 때 사용한다.

```
#> llctl -h [hostname] stop
```

- 각각의 해당 node의 loadleveler 데몬을 선택적으로 종료할 때 사용한다.

### 3.4.3 Loadleveler 데몬 재시작

```
#> llctl -g recycle
```

- 전체 node의 loadleveler를 일괄적으로 재시작 할때 사용한다.

```
#> llctl -h [hostname] recycle
```

- 각각의 해당 node의 loadleveler를 선택적으로 재시작 할때 사용한다.

## 3.5 Loadleveler account, llview 설정

### 3.5.1 ~loadl/acct 설정

- ~loadl/acct 폴더는 architecture가 맞는 기관의 acct 폴더를 수정하여 사용한다.
- /phome01/account 폴더에 해당 기관 디렉토리 생성 (예 : host01)
- /phome01/account/host01/RAW\_DATA 폴더 생성
- ~loadl/acct/bin script 수정

```
#> vi account.sh
- script 내용중 기관 이름에 현재 설치중인 시스템의 기관명으로 변경
line: 9~12
LOG=/LoadL/acct/log/cron.log
BASE=/LoadL/acct
GHIS=$BASE/GLOBAL_HISTORY
HOST=/phome01/account/host

line: 45~47
#Moving acctinfo file into the data dir and /account
cp -f $GHIS/acctinfo.$YYYYMMDD $BASE/DATA >> $LOG 2>> $LOG
mv $GHIS/acctinfo.$YYYYMMDD $HOST/acctinfo.$YYYYMMDD >> $LOG 2>>
$LOG

line: 49~53
#Moving a raw data into the data dir and /account
cp -f $GHIS/ASCLOG.$YYYYMMDD $BASE/RAW_DATA >> $LOG 2>> $LOG
mv $GHIS/ASCLOG.$YYYYMMDD $HOST/RAW_DATA >> $LOG 2>> $LOG
cp -f $GHIS/INPUT.$YYYYMMDD $BASE/RAW_DATA >> $LOG 2>> $LOG
mv $GHIS/INPUT.$YYYYMMDD $HOST/RAW_DATA >> $LOG 2>> $LOG
```

**3.5.2 llview 설정**

- llview는 architecture 맞는 다른 기관에서 복사하여 쓴다.
- 위치 : /usr/local/llview

**3.5.3 지정한 시간마다 account 데이터가 쌓일수 있도록 crontab을 설정해준다.**

- crontab 설정

```
#> crontab -e
*/5 * * * * cd /usr/local/llview/llapi;./llqxml > /phome01/llview/host01.xml
59 23 * * * /home/LoadL/acct/bin/account.sh
```

- llview, account data 확인 (/phome01/account)

**3.6 Loadveler User 제한 설정**

**3.6.1 Loadleveler 이용시 특정 사용자에게 작업 수를 제한할 때 설정 방법.**

- /LoadL/LoadL\_admin 파일에서 "USER STANZAS" 부분 설정 한다.

```
#> vi /LoadL/LoadL_admin
#####
USER STANZAS: (optional, default user stanza not optional)
```



```
These are sample user stanzas; user1, user2, user3 are sample labels
for users - revise these labels and specify attributes to each user.
#####
plsiuser: type = user
 maxjobs = 4 -> plsiuser 계정으로 한번에 run 가능한 job수
 maxidle = 5 -> plsiuser 계정으로 Idle 상태로 대기 가능한 job수
```

**3.6.2 위의 3.6.1 과 같이 설정후 LoadLeveler를 재시작 한다.**

#> llctl -g recycle

(※주의: 현재 작업이 실행중 recycle 을 하게 되면 처음부터 다시 시작 하게 된다.)

**3.6.3 위의 3.6.1 설정과 같이 plsiuser 계정으로 submit을 하면 다음과 같이 나온다.**

| Id                  | Owner    | Submitted  | ST | PRI | Class  | Running | On         |
|---------------------|----------|------------|----|-----|--------|---------|------------|
| solbaram-mg01.285.0 | plsiuser | 9/16 14:17 | R  | 50  | normal |         | solbaram01 |
| solbaram-mg01.286.0 | plsiuser | 9/16 14:17 | R  | 50  | normal |         | solbaram06 |
| solbaram-mg01.288.0 | plsiuser | 9/16 14:17 | R  | 50  | normal |         | solbaram12 |
| solbaram-mg01.287.0 | plsiuser | 9/16 14:17 | R  | 50  | normal |         | solbaram09 |
| solbaram-mg01.290.0 | plsiuser | 9/16 14:17 | I  | 50  | normal |         |            |
| solbaram-mg01.289.0 | plsiuser | 9/16 14:17 | I  | 50  | normal |         |            |
| solbaram-mg01.292.0 | plsiuser | 9/16 14:17 | I  | 50  | normal |         |            |
| solbaram-mg01.291.0 | plsiuser | 9/16 14:17 | I  | 50  | normal |         |            |
| solbaram-mg01.293.0 | plsiuser | 9/16 14:17 | I  | 50  | normal |         |            |
| solbaram-mg01.294.0 | plsiuser | 9/16 14:18 | NQ | 50  | normal |         |            |
| solbaram-mg01.295.0 | plsiuser | 9/16 14:18 | NQ | 50  | normal |         |            |
| solbaram-mg01.296.0 | plsiuser | 9/16 14:18 | NQ | 50  | normal |         |            |
| solbaram-mg01.297.0 | plsiuser | 9/16 14:18 | NQ | 50  | normal |         |            |

## 4. 통합 계정

PLSI 통합 시스템에서 전체 연동 자원에 걸쳐 통합 계정을 통한 SSO(Single Sign On) 환경을 구축하기 위해 이중화된 LDAP 서버를 통해 전체 연동 시스템에서 사용자 계정 정보를 공유하고 있으며, PLSI 계정신청은 사용자포털(portal.plsi.or.kr) 및 헬프데스크(helpdesk.plsi.or.kr)를 통해서 가능하다.

### 4.1 통합 계정 시스템 구성

통합계정시스템은 계정 정보를 저장하는 LDAP 서버와 사용자 상세 프로필 정보를 저장하는 Oracle 서버로 구성된다

#### 4.1.1 LDAP 서버

- 서버IP : xxx.xxx.xxx.xxx:389
- base : dc=kisti,dc=re,dc=kr
- userDn/password : xxx/xxx
- 내용
  - + Account ID, User ID, 비밀번호 값은 관리자 입력
  - + Account Code, Gid Number, Account No, Uid Number 값은 자동부여
  - + Home Parent Dir 값은 /phome01/(UserID)로 자동부여
- 예제

|            |        |              |      |            |         |                  |          |
|------------|--------|--------------|------|------------|---------|------------------|----------|
| Account ID | na0257 | Account Code | 200  | Gid Number | 2000257 | 초기 Shell         | /bin/ksh |
| User ID    |        | Account No   | 0257 | Uid Number | 2000257 | Home Parent Dir. | /home01  |
| 이름(영문)     |        | User 순번      | 0    | 비밀번호       |         | gecos            | na0257   |

#### 4.1.2 Oracle 서버

- 서버IP : xxx.xxx.xxx.xxx:1521
- ssid : account
- 관리자ID/PW : xxx/xxx
- 내용 : 테이블(13)
  - A. 계정 : ACCTMASTER, ADDITIONAL\_USER, HISTORY, USERID\_ISSUE, USERINFO
  - B. 자원 : CPUALLOC, PLSI\_ACCT\_DAILY\_TOTAL, PLSI\_REMAINDER\_OVER\_DATE
  - C. 기타 코드 : SANGHOCODE, ZIPCODE, APPLCODE, GROUPCODE, KINDCODE

### 4.1.3 LDAP 인증(java) : 간단한 인증처리 예시

```
import javax.naming.*;
import javax.naming.directory.*;
import java.util.Hashtable;

/** Demonstrates how to create an initial context to an LDAP server* using simple authentication.*/

class Simple { public static void main(String[] args) {

 String base = "ou=People,dc=kisti,dc=re,dc=kr";
 String ldapURL = "ldap://xxx.xxx.xxx.xxx:389";

 Hashtable authEnv = new Hashtable(11);
 String userName = "";
 String passWord = "";
 String dn = "";

 if (args.length != 2){
 System.out.println("Usage: Simpe [User_id] [Password] ");
 System.exit(1);
 }

 userName=args[0];
 passWord=args[1];

 dn = "uid=" + userName + "," + base;

 authEnv.put(Context.INITIAL_CONTEXT_FACTORY,"com.sun.jndi.ldap.LdapCtxFactory");
 authEnv.put(Context.PROVIDER_URL, ldapURL);
 authEnv.put(Context.SECURITY_AUTHENTICATION, "simple");
 authEnv.put(Context.SECURITY_PRINCIPAL, dn);
 authEnv.put(Context.SECURITY_CREDENTIALS, passWord);
 try {
 DirContext authContext = new InitialDirContext(authEnv);
 System.out.println("Authentication Success!");
 } catch (AuthenticationException authEx) {
 System.out.println("Authentication failed!");
 } catch (NamingException namEx) {
 System.out.println("Something went wrong!");
 namEx.printStackTrace();
 }
}
}
```

#### 4.1.4 계정발급 정보

- 사용자 종류
  - + 유료 : 일반(na)/학생(sa)
  - + 무료 : 내부(in)/과제(pa,pb,pc,pd)/PLSI(pe)
- 계정(acct)
  - + 유닉스 그룹 ID(문자) / 그룹 gid(숫자)
  - + 예제 : in0001
- 아이디(userid)
  - + ID : 소속구분자 + 순번 + 이름이니셜
  - + 소속(p, e, q, r 등) 구분자
  - + 순번 : 3자리
  - + 이름이니셜 : 3자리
  - + 예제 : p123kim

#### 4.1.4 계정발급 프로세스

- 4호기(일반/학생/과제) 계정
  - ①사용자 ②계약담당자 ③계정담당자
- PLSI(일반/초보) 계정
  - ①사용자 ②기관담당자 ③계약담당자 ④계정담당자
- 교육계정
  - ①사용자 ②교육담당자 ③계약담당자 ④계정담당자

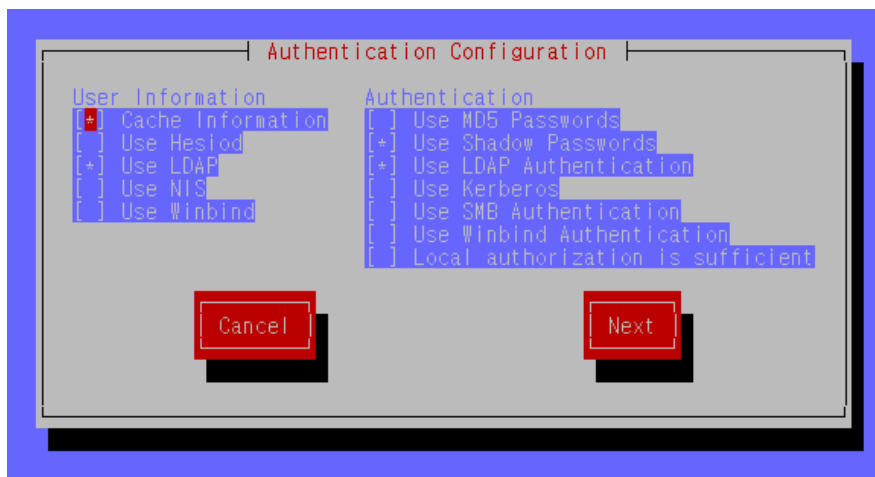
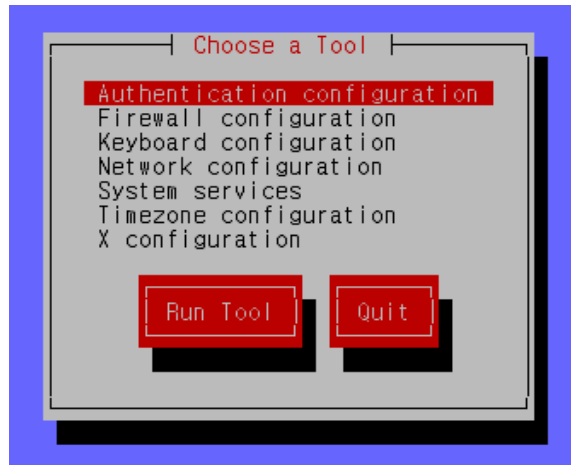
## 4.2 LDAP client 설정

추가 연동된 PLSI 시스템은 OpenLDAP 클라이언트 설정을 통해 계정 정보에 접근.인증을 수행할 수 있다.

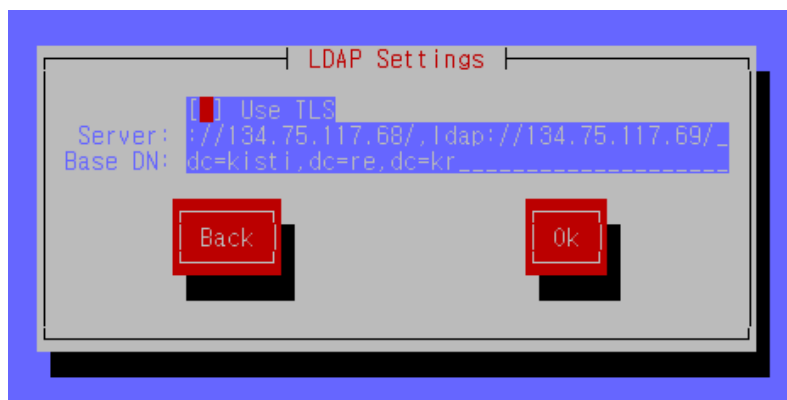
### 4.2.1 Setup을 이용한 설정방법

```
#> setup
```

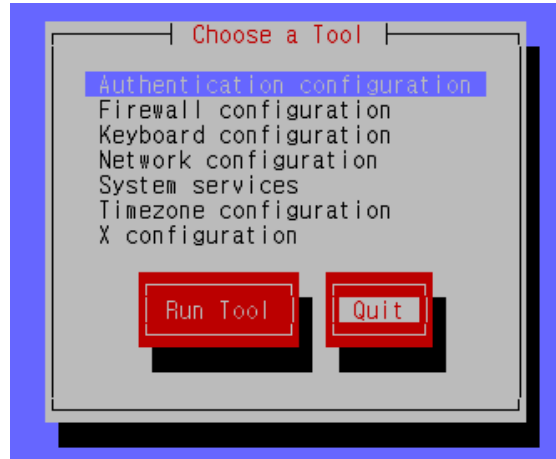
- "Authentication configuration" 선택
- User Information에서 "[\*] Cache Information" "[\*] Use LDAP" 체크
- Authentication에서 "[\*] Use Shadow Passwords" "[\*] Use LDAP Authentication" 체크



- LDAP Settings에서
- Server : ldap://134.75.117.68/,ldap://134.75.117.69/
- Base DN : dc=kisti,dc=re,dc=kr



- 설정이 완료되면 setup을 종료 한다.



#### 4.1.2 authconfig를 이용하여 설정하는 방법

```
#> authconfig --update --enablecache --enableldap --disablemd5
 --enableldapauth --ldapserver=ldap://134.75.117.68/,
 ldap://134.75.117.69/ --ldapbasedn=dc=kisti,dc=re,dc=kr
※ authconfig 옵션은 이어서 작성 한다.
```

#### 4.1.2 /etc/passwd PLSI netgroup 추가

```
#> vi /etc/passwd
+@PLSIAdm
+@PLSIUser
```

#### 4.1.3 /etc/nsswitch.conf 수정

```
#> vi /etc/nsswitch.conf
```

```
passwd: compat
passwd_compat: ldap
shadow: files ldap
group: files ldap
hosts: files dns
bootparams: files
ethers: files
netmasks: files
networks: files
```

```

protocols: files ldap
rpc: files
services: files ldap
netgroup: files ldap
publickey: files
automount: files ldap
aliases: files

```

#### 4.1.4 /etc/pam.d/system-auth-ac 수정

```
#> vi /etc/pam.d/system-auth-ac
```

```

auth required /lib/security/$ISA/pam_env.so
auth sufficient /lib/security/$ISA/pam_unix.so likeauth nullok
auth sufficient /lib/security/$ISA/pam_ldap.so use_first_pass
auth required /lib/security/$ISA/pam_deny.so

account required /lib/security/$ISA/pam_unix.so broken_shadow
account sufficient /lib/security/$ISA/pam_succeed_if.so uid < 100 quiet
account [default=bad success=ok user_unknown=ignore]
/lib/security/$ISA/pam_ldap.so
account required /lib/security/$ISA/pam_permit.so

password requisite /lib/security/$ISA/pam_cracklib.so retry=3
password sufficient /lib/security/$ISA/pam_unix.so nullok use_authok md5
shadow

password sufficient /lib/security/$ISA/pam_ldap.so use_authok
password required /lib/security/$ISA/pam_deny.so

session required /lib/security/$ISA/pam_limits.so
session required /lib/security/$ISA/pam_unix.so
session optional /lib/security/$ISA/pam_ldap.so

```

#### 4.1.5 LDAP 설정 확인

- LDAP 설정이 완료된 후 plsuser 계정을 확인한다.

```
#> id plsuser
```

```
uid=4050004(plsuser) gid=4050004(pe0004) groups=1000009(gauss),4050004(pe0004)
```

### 4.3. 파트너기관 담당자의 계정 관련 업무 안내

파트너 기관의 계정 관련 담당자는 통합 계정 관리 시스템에 접속하여 교육 계정 신청, 사용자 계정신청서 확인 및 처리, 추가구좌 심사와 같은 계정 관련 업무를 수행할 수 있다.

※ 통합 계정 관리 시스템 접속 계정 발급문의 : PLSI 사무국 : 최재민, cjmin010@kisti.re.kr

#### 4.3.1 교육 계정 신청서 작성

파트너 기관 교육에 활용될 교육 계정 발급을 파트너 기관 담당자가 발급 신청할 수 있다.

- <http://helpdesk.plsi.or.kr/plsi/jwsc/jwsc/>에서 "계정신청" 선택
- "PLSI 파트너기관 교육담당자" 신청서 작성



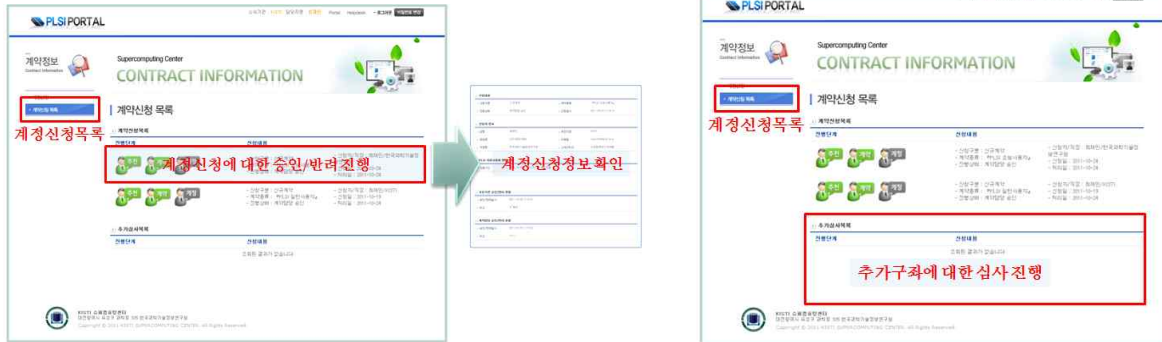
- PLSI 홈페이지 "교육&행사" → "교육일정" 에 등록할 수 있도록 PLSI 지원팀으로 연락 ( 담당자 : 박진양 , 연락처 : 042-869-1218 )





### 4.3.2. 사용자 계정신청서 처리 및 추가구좌 심사

- <http://helpdesk.plsi.or.kr/ccggadmin> 에서 "계정신청목록" 선택
- "계약신청목록"(오른쪽)에서 사용자 계정신청에 대한 목록에서 계정 신청에 대한 승인/반려 진행
- "추가심사목록"(오른쪽)에서 추가구좌에 대한 심사 목록에서 추가 구좌에 대한 심사 진행



## 5. 시간 동기화 설정

PLSI 연동 시스템 전체 노드 간 시간 동기화를 위해 외부 표준 시간 서버로부터 주기적으로 시간 동기화를 수행하는 3대의 PLSI 서비스 노드를 내부 표준 시간 서버로 지정하였다. 따라서 각 연동 시스템은 해당 시간 서버들을 포함하는 NTP 클라이언트를 설정해야 한다.

### 5.1 NTP client 설정

#### 5.1.1 /etc/ntp.conf 편집

```
Permit time synchronization with our time source, but do not
permit the source to query or modify the service on this system.
restrict default kod nomodify notrap nopeer noquery
#restrict -6 default kod nomodify notrap nopeer noquery - 주석처리

Permit all access over the loopback interface. This could
be tightened as well, but to do so would effect some of
the administrative functions.
restrict 127.0.0.1
#restrict -6 ::1 - 주석처리

Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

Use public servers from the pool.ntp.org project.
Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.centos.pool.ntp.org - 주석처리
#server 1.centos.pool.ntp.org - 주석처리
#server 2.centos.pool.ntp.org - 주석처리
server login01.plsi.or.kr
server login02.plsi.or.kr
server ims01.plsi.or.kr
```

#### 5.1.2 ntpd 데몬 시작

```
#> ntpdate login01.plsi.or.kr
#> service ntpd start
#> chkconfig --level 2345 ntpd on
```

### 5.1.3 시간 동기화 확인

- 데몬 시작 후 약 5분 정도 지나면 time sync됨

#> ntpq -p (+가 현재 서버로부터 Time sync 하고 있다는 표시)

```
remote refid st t when poll reach delay offset jitter
=====
ims01.plsi.or.k .STEP. 16 u - 64 0 0.000 0.000 0.000
+login01.plsi.or 141.223.182.106 2 u 13 64 177 0.149 0.011 0.027
*login02.plsi.or 141.223.182.106 2 u 5 64 177 0.163 0.123 0.057
```

## 6. 통합 모니터링 시스템

PLSI 통합 모니터링 시스템은 공개 모니터링 프로그램인 Nagios와 Adobe Flex로 개발된 통합 모니터링 시스템 웹 페이지를 통해 전체 연동 자원의 생사 및 서비스 별 장애 여부를 손쉽게 파악하고 빠른 시간 안에 대응할 수 있도록 구축되었다.

※ 접속 URL : <http://150.183.158.248/flex/#page=Login>

※ 모니터링 시스템 계정 발급 문의(파트너 기관 담당자) : 최재민, cjmin010@kisti.re.kr

### 6.1 모니터링 시스템 기능

- 시스템 장애 정보 수집 및 관리
- 연동자원의 성능 모니터링(CPU, MEM, FILE, JOBs, Load)
- 장애발생시 SMS / e-Mail 통지

#### 6.1.1 MAIN DASHBOARD



- Warning, Critical 이벤트에 따른 시간 별 상태변화를 실시간으로 확인
- 기관별로 이벤트들의 통계List를 확인
- 기관별로 상태 변화에 따라 실시간으로 이벤트상태를 확인
- Warning, Critical Error Logs

#### 6.1.2 기관별 DASHBOARD



- a. Group별 상태 변화에 따라 실시간으로 이벤트상태를 확인
- b. 선택된 Group의 Host Status Total, Service Status Total
- c. 선택된 Group의 시간별 Warning, Critical 발생 추이차트
- d. 선택된 Group의 최신 Error Logs

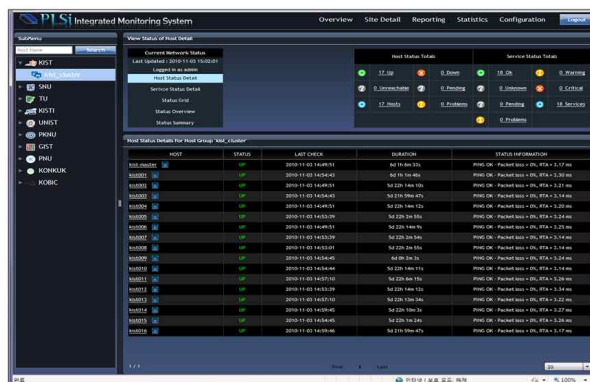
6.1.3 연동자원 성능 모니터링(ALL)



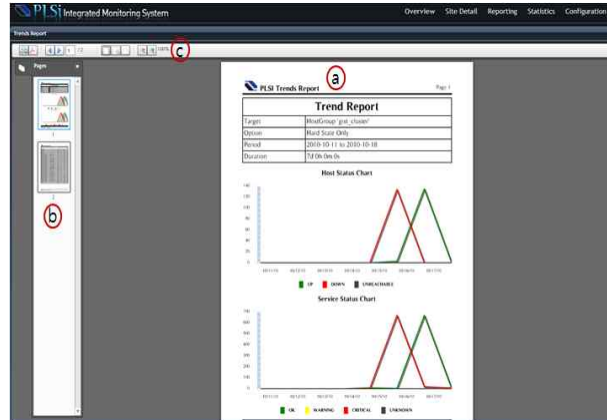
6.1.4 기관별 연동자원 성능 모니터링



6.1.5 SITE DETAIL INFORMATION



## 6.1.6 REPORTING



- Report의 타이틀, 차트, 그리드가 표현되는 화면 (A4 사이즈)
- 페이지 별 썸네일 이미지, 선택 시 페이지 이동
- 출력, pdf 파일, 확대, 축소 등의 기능 바

## 6.2 Nagios 클라이언트 구성

Nagios는 공개 프로그램으로써 PLSI 통합 시스템 환경에서 연동 시스템 장애를 모니터링하는 통합 모니터링 시스템의 핵심 구성 요소이다. 따라서, 연동 시스템 노드에서는 Nagios 클라이언트 프로그램을 설치하고 관련 서버 정보를 포함한 Nagios 클라이언트 구성을 해야한다.

### 6.2.1 설치 프로그램

- nagios-plugins 다운로드 (<http://www.nagios.org/download/plugins>)
- nrpe 다운로드 (<http://www.nagios.org/download/addons>)

### 6.2.2 nagios install

```
#> tar zxvf nagios-plugins-1.4.xx.tar.gz
#> configure --with-nagios-user=nobody --with-nagios-group=nobody
#> make
#> make install
```

### 6.2.3 nrpe install

```
#> tar zxvf nrpe-2.12.tar.gz
#> configure --prefix=/usr/local/nagios --with-nagios-user=nobody --with-
```

```

nagios-group=nobody --with-nrpe-user=nobody --with-nrpe-group=nobody
#> make all
#> make install-daemon
#> make install-daemon-config
#> make install-xinetd

```

#### 6.2.4 Nagios 설정

- /usr/local/nagios 소유권 변경
 

```
#> chown -R nobody:nobody /usr/local/nagios
```
- /nrpe 데몬 파일 수정 (nagios 서버 IP 추가)
 

```
#> /etc/xinetd.d/nrpe
only_from = 127.0.0.1 134.75.117.21
```
- /etc/services 파일에 nrpe 포트추가
 

```
#> vi /etc/services
nrpe 5666/tcp # NRPE
```

#### 6.2.5 Nagios 시작

- xinetd 데몬 재시작
 

```
#> service xinetd restart
```

#### 6.2.6 Nagios 연동 확인

- 5666 포트 확인
 

```
#> netstat -na | grep 5666
tcp 0 0.0.0.0:5666 0.0.0.0:* LISTEN
```
- nagios 서비스 체크 방법
 

서비스 체크를 하기 위해서는 ims01에서 실행한다.

```
#> /usr/local/nagios/libexec/check_nrpe -H host_name
#> /usr/local/nagios/libexec/check_nrpe -H host_name -c check_load
```

## 7. 보안설정

### 7.1 IP tables 설정

IPtables 는 특정 IP에 대한 접근을 차단하기 위한 설정이다. 설정된 IP 이외에 모든 IP를 차단하며, Class 단위, IP단위 별로 설정할수 있다.

PLSI 연동장비 에는 Master 서버에 설정 되어 있으며, 글로벌 파일서버 (PGFS), PLSI 통합 Login 서버 (login01, login02) 를 포함한 각 기관의 자체 IP대역 까지만 접근 허용하도록 설정 해 놓는다.

#### 7.1.1 IP tables 설정 파일 위치

- /etc/sysconfig/iptables

#### 7.1.2 IP tables 설정파일 수정

#> vi /etc/sysconfig/iptables (IP는 임의설정)

```

① *filter
② :INPUT ACCEPT [0:0]
③ :FORWARD ACCEPT [0:0]
④ :OUTPUT ACCEPT [18527:2968695]
⑤ :RH-Firewall-1-INPUT - [0:0]
⑥ -A INPUT -j RH-Firewall-1-INPUT
⑦ -A RH-Firewall-1-INPUT -i lo -j ACCEPT
⑧ -A RH-Firewall-1-INPUT -s 192.168.10.0/255.255.255.0 -j ACCEPT
⑨ -A RH-Firewall-1-INPUT -s 192.168.20.0/255.255.255.0 -j ACCEPT
⑩ -A RH-Firewall-1-INPUT -s 192.168.125.10 -j ACCEPT
⑪ -A RH-Firewall-1-INPUT -s 192.168.115.11 -j ACCEPT
⑫ -A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
⑬ COMMIT

```

② ~ ④ : 서버로 들어오는 기본 정책 " 모두 받아들인다" 로 세줄모두 비슷한 뜻이다.

⑤ : 이곳에서 설정하고자 하는 정책들을 적용하겠다는 뜻이다.

⑥ : INPUT에 관한처리를 RH-Firewall-1-INPUT 에 설정된 것을 사용한다는 뜻이다.

⑦ : localhost에서 생성되는 포트들에 관한 설정 사항으로 모두 ACCEPT 하게 설정되어 있다.

만일 이부분 없다면, localhost내에서 포트를 생성하여 내부에서 메시지가 교환되는 형태의 프로그램에서는 이상이 발생할수도 있다.

⑧ ~ ⑨ : 패킷의 발신지를 C class로 지정하여 접근을 허용한다.

⑩ ~ ⑪ : 패킷의 발신지를 IP 단위로 지정하여 접근을 허용한다.

⑫ : 위의 INPUT에서 허용한 부분 이외에는 모두 차단한다. (--reject-with 부분은 에러메시지 종류)



### 7.1.3 IP tables 시작 (재시작) 및 설정 확인

```
#> service iptables start (restart)
#> service iptables status , #>iptables -L
```

### 7.1.3 IP tables 설정 저장

```
#> service iptables save
```

- 저장후 **iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]** 확인,  
cat /etc/sysconfig/iptables 설정파일의 **#Completed on Mon Sep 15 14:00:56 2014**  
과 같은 날짜를 확인 한다.

## 7.2 TCP wrapper 설정

TCP wrapper는, 네트워크 서비스에서 관련한 트래픽을 제어하고 모니터링 할 수 있는 unix 기반의 방화벽 툴이다. 임의의 호스트가 서비스를 요청해 오면 실제 데몬을 구동하기 전에 접속을 허용한 시스템인지 여부를 확인하여 호스트명 및 서비스명을 로그에 남긴다음, 허가된 시스템은 서비스를 제공하고 허가되지 않은 경우에는 접속을 차단해 준다.

### 7.2.1 host.allow, hosts.deny 파일 설정

- /etc/hosts.allow , /etc/hosts.deny
- hosts.allow : 접속을 허용하는 설정파일 , 특정 서비스, IP 등을 지정할수 있다.
- hosts.deny : 접속을 차단하는 설정파일 , 특정 서비스, IP 등을 지정할수 있다.

### 7.2.2 tcp\_wrapper 설정

- hosts.allow, hosts.deny 파일을 수정하여 설정 한다.
- hosts.allow 파일에는 해당 클러스터 내의 IP 및 plsi management server IP만 설정해 둔다.
- tcp\_wrapper 는 설정후 데몬 재시작이 필요 없다.
- 설정시에는 deny 를 먼저 설정할 경우 차단될수 있으니 allow 파일을 먼저 설정한다.

## 7.2.2.1 /etc/hosts.allow 파일 수정

#> vi /etc/hosts.allow (해당 클러스터의 IP list. /etc/hosts 참조)

```
#####hosts cluster allow #####
all:192.168.0.11:ALLOW
all:192.168.0.12:ALLOW
all:192.168.0.13:ALLOW
all:192.168.0.14:ALLOW
all:192.168.0.15:ALLOW
all:192.168.0.16:ALLOW

admin node
sshd:134.75.117.40:ALLOW
sshd:134.75.117.41:ALLOW
```

- all:192.168.0.11:ALLOW : 192.168.0.11에서의 접속을 모두 허용

※ sshd:192.168.0.11:ALLOW : 192.168.0.11에서의 sshd 접속을 허용

## 7.2.2.2 /etc/hosts.deny 파일 수정

#> vi /etc/hosts.deny

```
sshd:all:DENY
```

- sshd:all:DENY : sshd 데몬으로 접속하는 모든 IP의 접속을 차단함(allow에서 허용한 IP만 가능)

## 8. 기타

### 8.1 pdsh(ver 2.10-4)

pdsh는 다수의 노드로 구성된 클러스터 환경에서 지정된 노드들에 걸쳐 병렬로 원격 명령을 실행하거나 파일을 전송하기 위해 사용되는 공개 프로그램이다.

#### 8.1.1 pdsh 설치

- pdsh 오픈소스 다운로드 (<http://sourceforge.net/projects/pdsh/>)
- pdsh base, pdsh rcmd ssh 설치 (ssh를 이용하는 경우 rcmd ssh로 설치한다.)
- #> rpm -ivh (pdsh base, rcmd file).rpm (rcmd 를 먼저 설치한다.)

#### 8.1.2 pdsh 사용법

- pdsh 를 이용하여 gpfs mount 상태 확인
- #> pdsh -w host[01-10] 'df -h | grep /phome01'
- #> pdsh -w host[01-10] 'df -h | grep /pwork01'
- host01~host10 번까지 /phome01, /pwork01 의 마운트 상태를 확인 할수있다.
  
- pdsh를 이용하여 전체 node를 일괄적으로 shutdown 할수 있다.
- #> pdsh -w host[01-10] 'shutdown -h now'
- host01~host10 번까지 일괄적으로 shutdown 시킬수 있다.
  
- pdsh 를 이용하여 전체 node에서 일괄적으로 해당 프로그램 실행 할수 있다.
- #> pdsh -w host[01-10] '/opt/test/pdsh\_test.sh'
- host01~host10 번까지 일괄적으로 /opt/test/pdsh\_test.sh script 를 실행한다.
  
- pdcp를 이용하여 전체 node에 일괄적으로 해당 파일,폴더를 복사할수 있다.
- #> pdcp -w host[01-10] /root/.bashrc /root
- master node 의 /root/.bashrc 를 host01~10 의 /root 로 복사한다.
- #> pdcp -r -w host[01-10] /root/package /root (-r option 추가)
- master node 의 /root/package 폴더를 host01~10 의 /root 로 복사한다

## 8.2 hostname 추가

추가 연동된 시스템의 호스트 이름 및 IP 주소 정보를 기타 연동 시스템의 /etc/hosts 파일에 추가한다.

## 8.3 TCP window size 및 Jumbo Frame 설정

TCP/IP 기반의 통신 환경에서 연동 시스템 간 물리적인 네트워크 대역폭을 효율적으로 활용하기 위해서는 전송지연시간 등을 감안한 TCP 윈도우즈 버퍼 사이즈 및 Jumbo Frame(MTU=9000) 설정이 반드시 필요하다.

특히, Jumbo Frame은 서버 네트워크 인터페이스 뿐만 아니라 전송 과정의 모든 네트워크 장비의 전송 포트에 대해서도 관련 설정이 동반되어야 한다.

### 8.3.1 /etc/sysctl.conf 파일에 TCP window size 설정을 추가 후 적용

```
#> vi /etc/sysctl.conf
```

```
Controls the TCP window size, appended by Joon, Woo(2008.09.05)
net.ipv4.tcp_rmem=4096 87380 1048576
net.ipv4.tcp_wmem=4096 87380 1048576
net.core.rmem_max=1048576
net.core.wmem_max=1048576
```

```
#> sysctl -p
```

## 8.4 CPU 사용자 정보 산출

연동 시스템 노드 별 CPU 사용자 정보를 산출하기 위해 sar 명령을 crontab에 등록하여 주기적으로 실행한다.

- /phome01/account/cpu\_usage/month 폴더에 연동기관 이름으로 폴더를 생성 한다.  
#> mkdir host
- /phome01/account/cpu\_usage/work 폴더에 연동기관 명으로 node list 작성한다.  
#> vi host\_node (node list 작성)
- /phome01/account/cpu\_usage script 를 작성해준다.
- /phome01/account/cpu\_usage/script 폴더에 연동기관 이름으로 script 작성한다.  
※ 기존에 작성되어 있는 script를 수정하여 사용한다.

# vi host\_sar.sh

```

※host 부분을 수정한다.
line 6~7 :
SAVE_OUT="/phome01/account/cpu_usage/month/host/usage${YEAR}${MONTH}"
SAVE_OUT_HOUR="/phome01/account/cpu_usage/month/host/usage${YEAR}${MONTH}_hour"

line 10~11 :
rm -f ${WORK}/host_cpu_1
for hostname in `cat ${WORK}/host_node`

line 13 :
ssh $hostname "LANG=C ;sar -f /var/log/sa/sa${DAY}" >> ${WORK}/host_cpu_1 2> /dev/null

line 16 :
x=`cat ${WORK}/host_cpu_1 | awk 'BEGIN { FS=" "

line 111 :
echo ${YEAR}"-${MONTH}"-${DAY} " " ${x} " host" >> $SAVE_OUT

line 113 :
cat ${WORK}/host_cpu_1 | awk -v HOST=host -v YY=${YEAR} -v MM=${MONTH} -v DD=${DAY} 'BEGIN {FS="

```

- sar 확인  
/phome01/account/cpu\_usage/script/host\_sar.sh 를 실행한다.  
/phome01/account/cpu\_usage/month, work 폴더를 확인한다.
- sar script crontab 추가

sar crontab 추가시 보안설정문제로 compute node에 설정한다. (예 : glory001)

```
#> crontab -e
```

```
05 01 * * * /phome01/account/cpu_usage/script/glory_sar.sh
```