

머리말

본 보고서는 LDG (LIGO Data Grid) 서버 구축을 위한 가이드이다. 본 보고서는 LIGO 실험에 대한 소개와 LDG 연구 환경을 구축하기 위한 LSC (LIGO Science Collaboration)에서 제공되는 LDG 구축과 관련되는 실제 어플리케이션을 설치하는 방법을 포함하여 LSC member로서 참여하는 방법 등을 기술하였다.

2010년 11월 10일

작 성 자: 유 진 승 (KISTI)
장 행 진 (KISTI)

<제 목 차 례>

1. LIGO 실험 설명	1
가. 개요	1
나. 중력파 물리학과 중력파의 시대 (2014년~)	2
다. LIGO (Laser Interferometer Gravitational-wave Observatory)	2
라. LISA (Laser Interferometer Space Antenna)	3
마. 중력파 검출의 파급 효과	4
바. 국내의 LIGO 실험 데이터 분석 인프라 구축과 연구 현황	6
2. LDG (LIGO Data Grid)	7
가. LSG Data Grid 적용 환경	7
나. LDG Package	8
3. LDG Server 구축하기	10
가. 관련 Packages	10
나. Installation of LDG Server 5.0	12
4. Configuring and Deploying Condor	19
가. YUM을 통한 Condor 기본 설치	19
나. Tarball을 통한 Condor 설치	20
5. LIGO Tools S/W 설치하기	28
가. LIGOTOOLS Initial Setup	29
나. 추가 Software 패키지 설치하기	32
6. OCTAVE Software 설치하기	34
7. LAL and LALApps Software 설치하기	35
가. 설치 준비 및 사전 요구사항	35

8. LIGO User 환경	37
가. LIGO VO 계정 신청하기	37
나. Digital Certificate 신청하기	38
다. LIGO VO Computer Resource 사용 권한 얻기	39

<그림 차례>

[그림 1] 지구상에서 건설되어 작동하고 있는 간접형 중력과 관측소의 분포 ..	1
[그림 2] LIGO at Livingston, Louisiana	3
[그림 3] 중력과 검출기의 Timeline	4
[그림 4] LISA 프로젝트의 도식 [출처: www.lisa.org]	4
[그림 5] 우주의 전 역사에서 전자기파를 통해 관측 가능한 시기는 매우 제한적임 (Big-bang nucleosynthesis 이후부터 today까지)	5
[그림 6] Host Certificate를 설치하기위한 과정	17
[그림 7] Condor 설치 과정 모습	21
[그림 8] Condor가 설치된 디렉토리 내용	23
[그림 9] LIGO 계정 신청서 내용 화면	37
[그림 10] LIGO VO Computer Resource Request 양식	40
[그림 11] LIGO VO Wiki page 접속시의 인증창	41
[그림 12] GRID VO를 통한 접근 가능한 LDG Computing자원 현황	41

<표 차례>

[표 1] LIGO Tools Software Packages 목록	28
[표 2]수동으로 추가해야할 LIGO Tools Software Packages 목록	32
[표 3]수동으로 추가해야할 LIGO Tools Software Packages URL 목록	33

LDG (LIGO Data Grid) 서버 구축 가이드

1. LIGO 실험 설명

가. 개요

미국 루이지애나 주와 워싱턴 주에 건설된 LIGO (Laser Interferometer Gravitational Wave Observatory)를 사용해 중성자별 병합, 블랙홀 쌍성 병합, 초신성 폭발 등의 천문 현상에서 발생하는 중력파를 검출하는 실험이다. 1997년 결성된 LSC (LIGO Scientific Collaboration)는 현재 12개 국가, 60개 기관, 약 700여명이 참여하고 있다. 2002년 첫 데이터 검출(S1) 이래로 총 5회의 검출이 있었고, 1일 평균 100GB 이상의 데이터가 생성되며, 2009년 7월 S6가 시작됐다. LIGO는 4Km의 중력 파동 검출기를 가진 실험이다. (<http://www.ligo.org/>)

중력파 검출이 성공하면 중력파 천문학이 성립되며 전파천문학과 더불어 우주를 관찰하는 또 다른 창을 갖게 된다. 중력파 천문학이 중요한 이유는 초기우주의 상태, 초신성 내부의 역학 등 전파신호로는 볼 수 없는 수많은 천문학 현상을 중력파 신호로 관측할 수 있어 천문 관측에 비약적인 발전을 줄 수 있는 새로운 관측 장비를 갖게 되기 때문이다. 국내에서는 2008년 5개 대학, 2개 출연연 소속의 연구자들이 중력파 데이터 분석 국제공동연구를 위한 컨소시엄을 구성했고, 2009년 9월 LSC 가입 신청이 승인되어 이러한 새로운 천문학의 시대를 대비하고 있다.



[그림 1] 지구상에서 건설되어 작동하고 있는 간섭형 중력파 관측소의 분포

나. 중력파 물리학과 중력파의 시대 (2014년~)

○ 중력파의 이론 및 역사적 배경

- 1915년 아인슈타인의 일반 상대성이론에서 이론적 예측
- 질량을 가진 물질이 진동을 할 때 중력파가 방출
- 중력은 자연계에서 가장 약한 힘이므로 지상에 도달하는 대부분의 천문학적 중력파는 세기가 매우 약하여 검출이 매우 어려움 (전자기파가 이론 출현 이후 검출에 15년이 걸린 반면, 중력파는 2009년 현재 이론 출현 이후 94년간 직접적 검출이 없었음)
- 1950년경부터 Joseph Weber는 지속적으로 중력파 검출 실험 제안 (막대형 검출기를 고안)
- 1975년 이중성-펄사(Binary Pulsar)가 발견되면서 이 펄사의 주기가 중력파를 방출하면서 감소하는 것이 일반상대성이론의 예측과 정확히 일치하면서 Taylor와 Hulse가 노벨상을 수상 (중력파의 간접적 검출)
- 1970년대 마이컬슨의 간섭계를 응용한 레이저 간섭계 형 중력파 검출기 건설이 제안됨 (K. Thorne)
- 미국에서 LIGO (Laser Interferometer Gravitational-wave Observatory)라 불리는 2개의 중력파 검출기 1999년 건설 완료됨. 현재 Hanford와 Livingston에 가동 중
- 각국의 중력파 검출기의 건설 - GEO, Virgo, TAMA, AIGO, LISA
- 중력파 검출기의 건설은 중력파 검출 프로젝트가 전 지구적 프로젝트로 진행됨을 의미. 중력파 검출 후 중력파 천문대로 활용 - LIGO의 경우 12개국 700여명의 연구진이 참여

다. LIGO (Laser Interferometer Gravitational-wave Observatory)

- 2007년 12월 S5(5차 Science Run) 마친 상태
- 2009년~2011년 E-LIGO(Enhanced LIGO)가 가동
- 현재 S5까지의 데이터로는 중력파 신호가 검출되지 않았으나 이는 현재의 검출기정밀도로는 약 100년에 1개 정도의 Event rate이므로 예상되는 결과임

○ Advanced LIGO

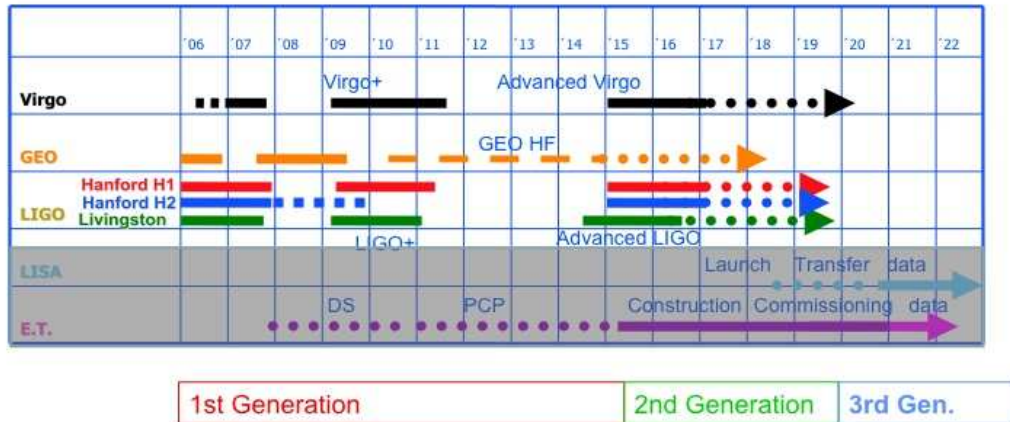
- 새로운 프로젝트인 Advanced LIGO 프로젝트의 예산 승인 확정(2014년까지 Installation 완료 예정)
- 검출기의 정밀도가 현재의 10배 향상
- 1년에 6000여개 정도의 Event Rate
- 2-3일에 1개꼴로 중력과 신호 검출 예상



[그림 2] LIGO at Livingston, Louisiana

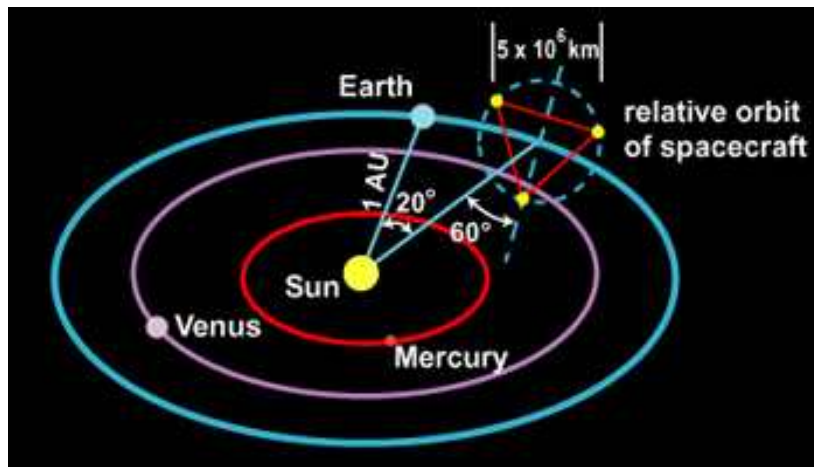
라. LISA (Laser Interferometer Space Antenna)

- 지상 검출기가 지구상의 환경 노이즈에 노출되어 있다는 단점을 보완
- 우주 공간(지구 주변)에 3개의 위성 안테나를 띄워 중력과 검출에 사용
- 장파장의 중력과 검출 가능: 우주배경 복사처럼 Stochastic signal 검출



G. Gonzalez, LSU, Pohang Summer School

[그림 3] 중력과 검출기의 Timeline



[그림 4] LISA 프로젝트의 도식 [출처: www.lisa.org]

마. 중력과 검출의 파급 효과

○ 직접적인 파급효과

- 현재 전파망원경의 최대 가시거리는 이론적으로 우주의 역사를 비추어 보아 원자핵이 합성되기 시작한 이후 (전자기파의 출현 이후: 핵 합성기 (Nucleosynthesis)이후)가 최대 가시거리임
- 빅뱅 초기의 우주부터 핵 합성기 이전까지는 오로지 우주에 중력파만이 존재함
- 중력파는 전자기파와 달리 다른 물질들과 상호작용을 하지 않기 때문에 우

주 초기의 정보를 고스란히 담고 있음

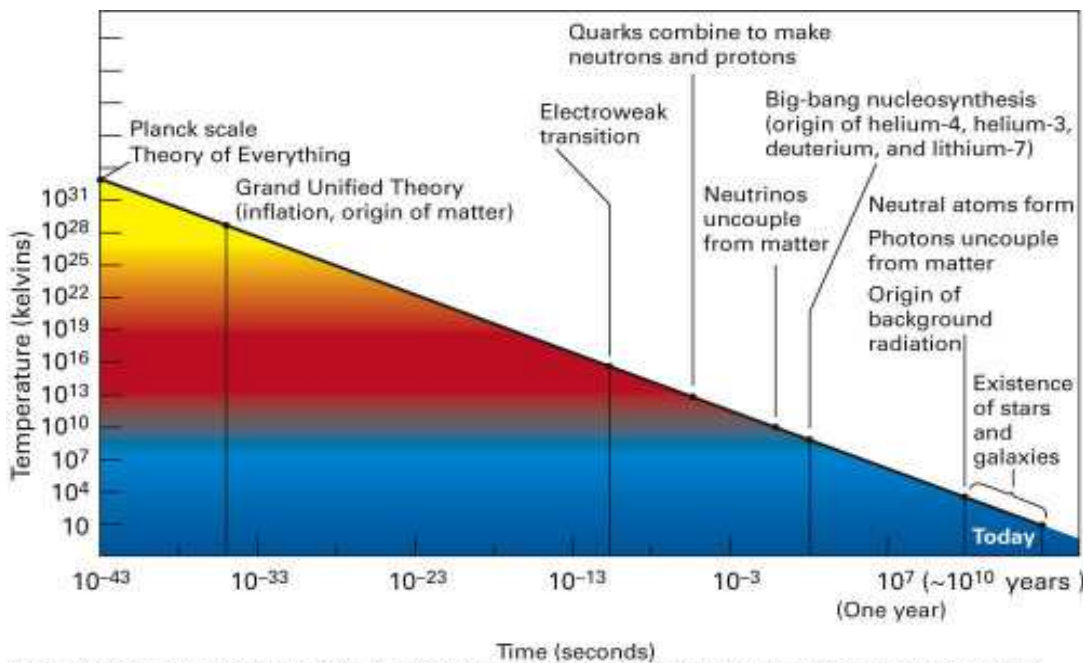
- 따라서 중력파 검출의 의미는 우주초기 빅뱅이후의 우주의 모습을 알 수 있는 유일한 통로임

- 중력파는 초신성 폭발과 같은 천체 현상에서 전파망원경이 볼 수 없는 폭발 내부의 정보를 담고 있어 중력파를 통해 폭발 내부를 알 수 있음

- 이 중력파가 검출되면 현재 건설되어 있거나 앞으로 건설 중인 모든 지상의 검출기 및 우주의 검출 안테나 등은 중력파 망원경의 역할을 하게 되며, 이는 새로운 학문의 분야인 "중력파 천문학"의 시작을 의미함

※ 광학 천문학 → 전파 천문학 → 중력파 천문학

- 중력파의 검출 및 검출과 관련된 모든 응용분야는 노벨상수상 후보로의 업적으로 평가됨



[그림 5] 우주의 전 역사에서 전자기파를 통해 관측 가능한 시기는 매우 제한적임 (Big-bang nucleosynthesis 이후부터 today까지)

○ 간접적 파급효과

- 중력파 검출 실험은 물리학, 천문학, 천체물리학, 수학, 컴퓨터 과학, 공학 등이 총체적으로 집약되어 진행되는 거대 프로젝트임

- 따라서 인접, 혹은 다양한 학문의 종합적 지식을 결집하여야 하므로 융/복합

의 다 체계적인 연구 역량이 필요한 학문

- 전 세계의 산재한 중력과 검출기 간의 상호 교류를 통하여야 이 검출 실험의 정밀도와 정확도를 보장할 수 있기에 거대 국제 공동 프로젝트로서 추진되어야 함
- 중력과 검출기의 건설은 다양한 학문 분야의 시설 인프라를 통해 연구 인력을 결집시키고, 그 성과를 창출한다는 점과 국제적인 거대 프로젝트에 참여한다는 점에서 국가위상 향상에 기여함

바. 국내의 LIGO 실험 데이터 분석 인프라 구축과 연구 현황

- 국내에서는 2008년 5개 대학, 2개 출연연 소속의 교수, 연구원, 학생 약 15~20명이 중력과 데이터 분석 연구를 위한 컨소시엄인 『한국중력파연구단』(KGWG, Korean Gravitational-Wave Group)을 구성
- 국내에는 LIGO와 같은 중력과 검출 관측소가 없기 때문에 중력과 데이터 분석 연구를 위해서는 LIGO와 같은 해외 관측소의 데이터를 활용해야 함
- 이는 LSC의 가입을 통해 가능
- 약 1년여의 준비 끝에 2009년 9월 23일 LSC 가입 신청이 승인되어 국제공동 연구 거대 실험 참여의 길이 열림
- 국내연구진은 CBC(Compact Binary Coalescence) 중력파원에 대한 분석 연구를 하기로 LSC 측과 협의됨
- LIGO는 1일 평균 약 100GB의 대용량 데이터 생성
- 중력과 데이터 분석을 위한 컴퓨팅 자원은 KISTI NSDC 활용
- 2009년 12월 중력과 데이터 분석 프로그램(LIGO Grid) 설치 예정
- Detector characterization 및 데이터 분석 기술 습득을 위해 LSU(Louisiana State University), Caltech, UWM(University of Wisconsin Milwaukee) 등에 국내 연구진 파견 예정

2. LDG¹⁾ (LIGO Data Grid)

LSC DataGrid는 일정하고 동일한 형태의 LSC data 분석 환경을 만들기 위한 “Grid Computing Middleware”와 함께 LSC 계산 자원 및 데이터 저장장치 자원의 조합이다. 보다 구체적으로 표현한다면, LSC DataGrid는 다음 자원들의 조합이라 할 수 있다.

- ◆ Linux clusters at the LIGO Hanford and Livingston sites
- ◆ Linux clusters at the Tier-1 site Caltech
- ◆ Linux clusters at Tier-2 sites MIT, UWM and PSU
- ◆ Linux clusters at GEO sites Birmingham, Cardiff and the Albert Einstein Institute (AEI)
- ◆ Linux clusters at Caltech, MIT, LHO and LLO
- ◆ Data storage at each of the sites

이와 함께, Globus, GridFTP, Condor와 같은 Grid computing software를 포함하여, LSC 과학자들이 일정하고 동일한 형태로 자원을 접근하고 데이터를 분석할 수 있도록 하여준다. 이러한 Software tool들을 가지고 있는 application으로 Grid middleware distribution인 VDT²⁾ (Virtual Data Toolkit)이다. VDT는 OSG³⁾ (Open Science Grid)의 product로 NSF 와 DoE에서 자금 지원을 받아서 개발되고 있다. OSG가 자금 지원을 받기 전에는 VDT는 GriPhyN, iVDGL, PPDG 협력체의 product 였다.

가. LSG Data Grid 적용 환경

LSC Computing Committee는 LDG가 적용되어 운용될 수 있는 연구 환경에 대해 매년 반영하는 방식으로 OS를 결정한다. GRID Middleware가 이기종의 computing 환경에서 동작하는 방식이지만, 현재의 Computing 시장에서는 매우 변화무쌍하고 빠르게 변화하고 있기에 모든 OS에 대한 환경을 지원하는 것은 너무 무리가 가는 상황인 것은 당연하다. 이에 LDG가 제대로 동작할 수 있는 환경을 특정하여 지속적으로 안정적으로 개발하여 제공하기 위하여 특정 OS 몇 개에 대해서만 안정적인 서비스를 제공하고 있다. 현재 LDG가 동작 가능한 OS는 다음과 같다.

- ◆ CentOS 5.3 (RHEL5) i386, x86_64
- ◆ Debian 5 Lenny i386, amd64
- ◆ Fedora Core 3,4,7,8,10,11

1) LIGO Data Grid: <https://www.lsc-group.phys.uwm.edu/lscdatagrid/>

2) VDT Website: <http://vdt.cs.wisc.edu/components/vdt.html>

3) OSG Website: <http://www.opensciencegrid.org>

위의 OS는 Server, Client로 모두 동작이 가능하다.

나. LDG Package

2010년 7월 기준으로 LDG Package (LDG-5.0)는 VDT-2.0.0p15와 Condor-7.4.2⁴⁾를 기반으로 하여 구성되어 있다.

VDT package는 LDG-4.8에서부터 사용되어 왔으며, RPM, DEB, Mac bundle installer를 통하여 설치가 진행되며, 이미 기본적인 환경 설정 부분이 완료되어 포함하고 있다. 즉, 일단 설치되면 서비스를 즉시 시작할 수 있으며 추가적인 환경 설정이 필요하지 않는다. VDT package에 포함되어 있는 세부 구성요소로는 Globus-4.2.1, MyProxy-5.0.0, GSI-OpenSSH-5.3.1, UberFTP-2.4.0, OSG-CA-1.13가 있다.

하지만, Condor-7.4.2는 VDT의 구성 요소로 포함되어 있지 않고 별도로 설치가 필요하다.

CentOS 5.3기준으로 설치해야할 패키지 크기는 LDG-Server의 경우는 대략 12MB 정도이며, LDG-Client의 경우는 5.8 MB 정도이다.

다. LIGO 데이터 분석 툴

LIGO 데이터 분석 연구를 위해서 필요한 컴퓨팅 툴이 필요하다. LIGO Tool⁵⁾이라는 이름으로 필요한 툴들이 정리되어 있다. 또한, LIGO 분석 연구를 위해서는 Matlab이 기본적으로 필요한데, Matlab은 상업용 버전으로 상당히 고가에 속하여 이를 흉내낸 GNU 무료 프로그램이 있는데, 이것이 Octave⁶⁾이다. 이의 분석 툴이 LDG 환경에서의 연구 환경에 반드시 필요한 항목이다.

라. 필요한 디스크 공간

- LSC Software: 설치에 필요한 크기는 /opt에는 약 420MB가 필요하며, /usr에는 약 1.1GB가 필요하다.
- Condor Software: 설치파일 400MB, 실행 파일 1.1GB이며, 동작중의 Log

4) Condor 공식 Web site: <http://www.cs.wisc.edu/condor/>

5) LIGO Tool Homepage <http://www.ldas-sw.ligo.caltech.edu:80/ligotools/>

6) Octave, <http://www.gnu.org/software/octave/>

저장 등에 추가로 공간이 필요하다.

- Ligo Tools: LIGOtools Software package를 Linux 버전으로 설치 용량은 약 480MB에 이른다.
- Octave:
- LAL, LALApps: LIGO Data 분석 pipeline을 만들기 위한 필수 소프트웨어로, 기본적으로 LDG 서버에 포함되어 있지만, 각 연구자 개인이 필요한 부분을 최적화하고 별도로 개발하여 사용하여야 하므로 각 개인별로 별도로 추가 하여 설치하여야 한다. 이를 위해 "Development Tools", "Development Library" Linux Package를 설치하여야 한다. 이때, 추가적으로 약 250M~300M 정도 필요하다.

3. LDG Server 구축하기

LDG System을 구축하기 위한 과정은 NSDC Twiki Page (<http://134.75.123.21/twiki/bin/view/Main/LDGConstructionLog>) 에 기록되어 있다. LDG System 구축에 대한 기본 내용은 LIGO의 웹 사이트에 들어 있으며, 이중 NSDC 팀에서 구축중인 시스템은 CentOS 계열의 SL (Scientific Linux) 5.40 x86_64이므로 이에 맞는 내용을 Twiki에 저장하도록 한다.

2010년 7월 현재 안정된 VDT 버전은 VDT-2.0.0p15이다. 이를 설치하기 위한 안내 페이지는 <https://www.lsc-group.phys.uwm.edu/lscdatagrid/doc/installserver.html>이다.

LDG Server를 설치하기 위해서는 먼저 client에서도 필요한 LSC Software package를 설치해야 한다. 설치해야할 패키지는 2가지 종류로 LIGO package와 Non-LIGO package이다. 이에 대한 패키지 항목은 아래에 기술되어 있다.

가. 관련 Packages

1) 필요한 LIGO Package⁷⁾

CentOS5에 설치해야 하는 package는 다음과 같다.

lscsoft-all, lscsoft-internal, lscsoft-external, lscsoft-external-abc, lscsoft-external-periodic, lscsoft-frame, lscsoft-metaio, lscsoft-lal, lscsoft-lalapps, lscsoft-glue, lscsoft-ldas, lscsoft-root, lscsoft-dol, lscsoft-external-omega, lscsoft-gds, lscsoft-auth, lscsoft-external-octave

가) LSCSOFT-ALL

모든 rpm package 들을 포함한다.

나) LSCSOFT-INTERNAL

includes all the software built by LIGO/GEO/VIRGO Collaboration and all external packages that were tweaked/modified in order to fit LSCSoft requirements (exception to the rule: cfitsIO)

7) <https://www.lsc-group.phys.uwm.edu/daswg/download/repositories.html#lscsoft>

Packages	Description	Critical Package
FrameL	for data_frame manipulation http://lappweb.in2p3.fr/virgo/FrameL/	libframe, libframe-devel, libframe-utils, libframe-matlab
MetaIO	for LIGO_LW files metadata manipulation	libmetaio, libmetaio-devel, libmetaio-utils, compat-libmetaio
User Environment Scripts	LSCSOFT environmental variables definitions	lscsoft-user-env
LAL	LIGO Algorithm Library	lal, lal-devel, lalstochastic, lalstochastic-devel
LALAPPS	LAL based Applications	lalapps
GLUE	Grid LSC User Environment	glue
FrameCPP (deprecated) LDAS-TOOLS	C++ interface to access frame structures http://www.ldas.ligo-wa.caltech.edu/doc/framecpp/html/ LDAS Developers Manual LDAS Development notes	ldas-tools, ldas-tools-general, ldas-tools-general-devel, ldas-tools-genericAPI, ldas-tools-genericAPI-devel, ldas-tools-framecpp, ldas-tools-framecpp-devel, ldas-tools-diskcacheAPI, ldas-tools-diskcacheAPI-devel
DOL	Data Monitoring Tool (DMT) Off_Line http://www.ligo.caltech.edu/~jzweizig/dmt/DMTProject/ http://www.ligo-wa.caltech.edu/gds/	dol
GDS	LIGO Global Diagnostics System http://www.ligo-wa.caltech.edu/gds/	gds-core, gds-crtools, gds-devel, gds-monitors, gds-runtime, gds-web
NDS2-Client	Part of DMT offline that allow the user to down-load LIGO data from the V2 LIGO Network Data Servers. https://www.lsc-group.phys.uwm.edu/daswg/wiki/NetworkDataServer2	nds2-client, nds2-client-debuginfo, nds2-client-devel, nds2-client-headers, nds2-client-matlab-devel, nds2-client-matlab2006a, nds2-client-matlab2006b, nds2-client-matlab2007a, nds2-client-matlab2007b, nds2-client-matlab2008a, nds2-client-matlab2008b, nds2-client-matlab2009a, nds2-client-matlab2009b, nds2-client-matlab2010a, nds2-client-matlabr13, nds2-client-octave

다) LSCSOFT-EXTERNAL

모든 lscsoft-external-* group과 external package를 포함한다. lscsoft-external-* group은 설치되어 동작하는 하나의 node에서 완전히 동작하여야 하는 분석 파이프라인 (CBC, Periodic, Stochastic, Burst) 각각에 대해 “최소 요구”사항 각각을 갖는다.

라) LSCSOFT-EXTERNAL-CBC

lscsoft internal packages (libframe, metaio, lal, lalapps, glue, (pylal)) 와 external packages (numpy, scipy, python-matplotlib,

python-sqlite2, ROOT, R (+ coda, xtable), spr/spr-devel)를 포함한다.

마) LSCSOFT-EXTERNAL-PERIODIC

lscsoft internal packages (libframe, metaio, lal, lalapps, glue, (pylal)) 과 external packages (numpy, scipy, python-matplotlib, python-sqlite2, ROOT, R)를 포함한다.

바) LSCSOFT-EXTERNAL-OMEGA

ImageMagick, xorg-x11-server-Xvfb, java-1.6.0-openjdk(-devel), moreutils 에 대한 외부 패키지를 포함한다.

사) LSCSOFT-AUTH

LIGO Authentication Project에 대한 외부 패키지를 포함한다.

아) LSCSOFT-EXTERNAL-OCTAVE

Octave Project의 모든 설치에 대한 외부 패키지를 포함한다.

2) non-LIGO Package

위의 LIGO Software를 설치하기 위해서는 기타 Software Tool (gsl, fftw, python-numarray, python-matplotlib, python-pyxp)이 존재해야 한다. ROOT, (python-)SQLObject, cfitsIO, spr, ddd, ipython 등과 같은 tool은 반드시 필요한 사항은 아니다.

이러한 Software Tool은 이미 해당 OS의 저장소 (Repository)에 Extra package로 혹은 Fedora EPEL에 이미 포함되어 있다. 따라서, 해당 패키지는 Yum을 통하여 직접 설치할 수 있거나 lscsoft-* group의 종속성에 따라 자동으로 설치할 수 있다.

나. Installation of LDG Server 5.0

1) LSCSOFT와 LDG repository 설치하기

LSCSOFT와 LDG repository를 설치하는 것은 Client에서도 필요한 과정이다. lscsoft-all yum package는 184개의 패키지로 구성되어 있으며 /opt와 /usr 디

렉토리에 설치된다. 설치에 필요한 크기는 /opt에는 약 420MB가 필요하며, /usr에는 약 1.1GB가 필요하다.

가) 먼저 LSCSOFT Yum repository를 설치한다.

1. LSC data analysis software stack을 설치하기 위해서는 root로 로그인 한다.
2. /etc/yum.repos.d/lscsoft.repo 파일을 생성하여 다음의 내용을 추가한다.

```
[lscsoft]
name=LSC Data Analysis Software
baseurl=https://www.lsc-group.phys.uwm.edu/daswg/download/software/centos/5/$basearch
enabled=1
gpgcheck=0
```

3. testing repo를 설치하고 싶다면, /etc/yum.repos.d/lscsoft-testing.repo 파일을 생성하여 다음의 내용을 추가한다.

```
[lscsoft-testing]
name=LSC Data Analysis Software
baseurl=https://www.lsc-group.phys.uwm.edu/daswg/download/software/centos/testing/5/$basearch
enabled=1
gpgcheck=0
```

4. CentOS 5.2의 경우에는 python module (m2crypto for python v2.4.3)에 버그가 있으므로 이것으로 Yum 에러를 발생시킬 수 있는데, 이것은 m2crypto 모듈을 m2crypto-0.16-6.e15.2로 update시키면 해결될 수 있다. update 하는 방법은 다음과 같다.

```
yum update --disablerepo=lscsoft m2crypto
```

5. EPEL (Extra Packages for Enterprise Linux)를 update 시킬 필요가 있는데, 해당 package에 대한 설명은 위의 "관련 packages" 단락을 참고한다. update하는 방법은 다음과 같다.

```
rpm -Uvh
http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-3.noarch.rpm
```

6. 위와 같이 하는 동안 EPEL GPG key를 import하라는 요구하는 메시지가 출력될 수 있는데, 이를 해결하기 위해 직접 rpm을 사용하여 GPG-key를 import할 수 있다.

```
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL
```

7. 위에서 지정한 방법대로 모두 끝마쳤다면, LSC Data Analysis Software 모두를 install하도록 한다⁸⁾

```
yum clean all ### this is optional
yum groupinstall lscsoft-all
```

8. lscsoft-all package가 설치된 후에는 /opt/lscsoft directory가 생성되었음을 볼 수 있다.

나) Update 방법

LSD Data Analysis software를 수동으로 update 하는 방법이다.

```
yum groupupdate lscsoft-all
```

사용자는 LSC Software 환경을 사용할 수 있는 환경 설정을 다음과 같이 만들어 사용하도록 한다. bourne shell의 경우에는 .profile에 다음과 같이 지정하며,

```
if [ -f /opt/lscsoft/lscsoft-user-env.sh ] ; then
    . /opt/lscsoft/lscsoft-user-env.sh
fi
```

C shell의 경우에는 .login 파일에 다음과 같이 지정하도록 한다.

```
if ( -r /opt/lscsoft/lscsoft-user-env.csh ) then
    source /opt/lscsoft/lscsoft-user-env.csh
endif
```

2) ldg-repo-config⁹⁾ rpm 설치하기

아래와 같은 명령어를 통하여 /etc/yum.repos.d 디렉토리 아래에 ldg.repo를 생성하도록 한다.

LDG client 에서도 필요한 과정이다.

8) lscsoft-all package를 install 하기 위해서는 /usr disk partition에 최소 1.5GB 이상의 여유 공간이 존재해야 설치가 가능하다.

9) VDT Software를 현재 production level로 안정화되어 배포되고 있다. 따라서, LDG 5.0 package는 VDT production repository를 사용할 수 있도록 기본으로 production 형태로 설정되어 있다. 만약, 무슨 이유로든 testing을 위해 VDT-testing repo를 사용하려면, "/etc/yum.repos.d/ldg.repo" 파일에 있는 "vdt-testing-centos5"를 enable 시켜야 한다.

```
yum clean all
yum repolist
yum install ldg-repo-config
```

3) LDG Server package 설치하기

`yum groupinfo ldg-server` 명령어를 통하여, 어떠한 VDT package (대략 12MB 정도의 크기임)를 설치할 것인가를 알아 볼 수 있다. 이를 통하여 다음과 같은 package가 설치됨을 알 수 있다.

```
Group: LDG-5.0 SERVER
Description: VDT rpm packages for the installation of LIGO Data Grid
server (version 5.0). This is an alternative to former Pacman
installation.
Mandatory Packages:
  ldg-cert-util
  ldg-server-version
  osg-ca-certs
  vdt-globus-dev
  vdt-globus-essentials
  vdt-globus-gridftp-server
  vdt-globus-simple-ca
  vdt-gpt
  vdt-gsi-openssh-client
  vdt-gsi-openssh-server
  vdt-myproxy-client
  vdt-myproxy-essentials
  vdt-uberftp
```

```
yum groupinstall ldg-server
```

모든 Grid ssh 틀은 gsissh처럼 gsi가 앞에 붙어서 일반 ssh와 구분함으로써, OS에 설치되어 있는 기존의 ssh service와 공존할 수 있도록 한다.

osg-ca-certs-1.13-0.noarch 패키지가 이전의 배포된 VDT certificate package (vdt-ca-cert-48-1.noarch 등)와 충돌하므로 문제되는 rpm을 제거하고 다시 설치하도록 한다.

4) VDT openGSISSH server 활성화 시키기

기존의 ssh service를 중지시키고, gsisshd service를 활성화 시키도록 한다. 이때, rebooting 될 때 기존의 ssh service가 활성화 되지 않도록 설정할 필요가 있다.

```
service sshd stop
service gsisshd start
chkconfig sshd --level 235 off
chkconfig gsisshd --level 235 on
```

gsisshd는 기본으로 22번 포트에 대해 GSI Authentication 기능 (valid grid proxy만 가진 사람만 login)만 사용하도록 설정되어 있다. 따라서, passwd 인증 방식으로 login할 수 있도록 사용하도록 하거나, X11 forwarding 혹은 RSA/DAS 등의 자동 login이 가능하도록 설정해야 할 경우 다음 항목을 /etc/gsisshd_config 파일에 추가하여 주도록 한다.

```
X11Forwarding yes
PasswordAuthentication yes
```

이후 "service gsisshd restart" 명령을 통하여 gsisshd 서비스를 다시 시작시켜 주면 위에서 설정한 기능을 사용할 수 있다.

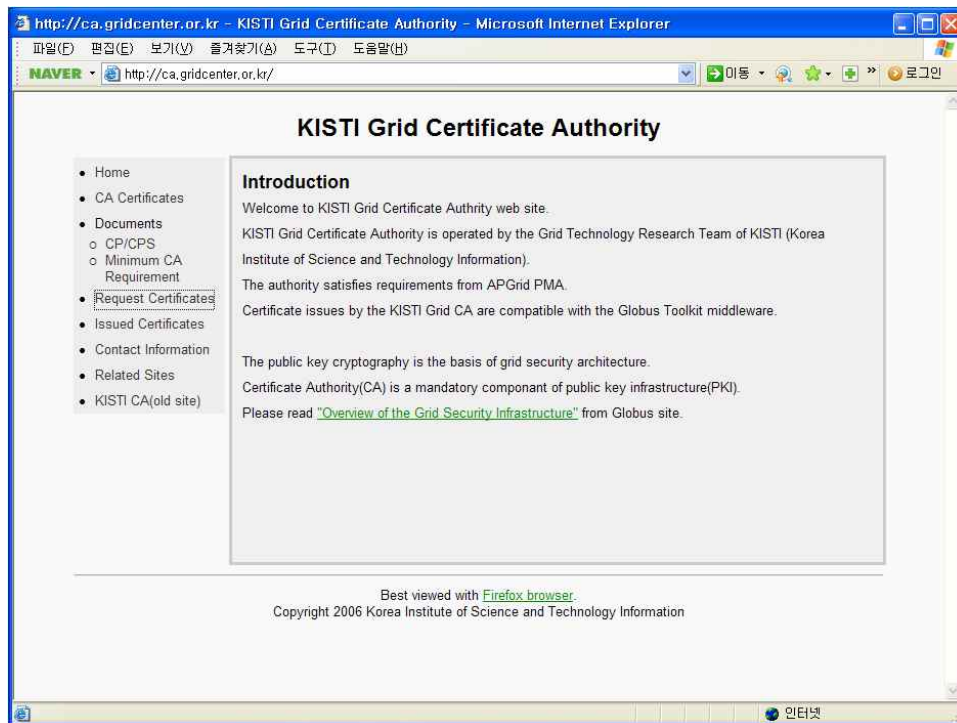
5) GridFTP server 시작하기

GridFTP 서비스는 기본으로 2811 포트를 사용하며, 다음과 같이 서비스를 활성화 시킬 수 있다.

```
service globus-gridftp-server start
```

6) Host Certificate 설치하기

host certificate를 설치하는 것은 다음의 과정을 거쳐서 진행하도록 한다. KISTI에서 운영하는 CA 발급 사이트를 통하여 host certificate를 발급받도록 한다. CA site는 아래 그림과 같다.



[그림 6] Host Certificate를 설치하기 위한 과정

host certificate를 발급받기 위해서는 본인의 Grid 인증서를 사전에 받아서 Web Client에 설치하여야 한다. 이를 통하여 Host Certificate 를 요청할 수 있다. Host Certificate를 요청하기 위해서는 해당 host의 CSR (Certificate Signing Request)값을 만들어서 upload를 해야 하는데, 해당 서버 이름이 ldg.sdfarm.kr인 경우 CSR를 만드는 방법은 다음과 같다.

```
openssl req W
  -new -sha1 -newkey rsa:1024 -nodes W
  -keyout hostkey.pem W
  -out hostcert_request.pem W
  -subj "/C=KR/O=KISTI/O=GRID/O=KISTI/CN=hostW/ldg.sdfarm.kr"
```

위의 subj 변수 값중에 앞의 KR, KISTI, GRID는 KISTI가 CA 기관으로 들어간 내용이며, 뒤의 O=KISTI/CN=hostW/ldg.sdfarm.kr는 인증서를 신청하는 host가 속해있는 기관 및 host 이름이다. 이때, "CN=hostW/ldg.sdfarm.kr"에서 hostW/ 부분은 globus 소프트웨어를 사용하는 host의 인증서임을 의미한다. 위의 명령이 성공적으로 실행될 경우 "hostkey.pem", "hostcert_request.pem" 파일이 생성되는데 위의 keyout, out parameter에 의해 각각 생성되는데, "hostkey.pem"파일은 인증서와 쌍을 이루는 host에 대한 개인키를 담고 있으며

host에 반드시 보관되어 있어야 하는 파일로 생성된 두 개의 파일은 "/etc/grid-security" 폴더에 보관하면 된다.

생성된 "hostcert_request.pem"파일을 통하여 제대로 Subject가 생성되었는지를 다음과 같은 방식으로 확인이 가능하다.

```
openssl req -in hostcert_request.pem -text -noout
```

이렇게 검사하여 Subject와 RSA public key size가 위에서 지정한 1024 bit으로 생성된 것이 확인되면 "hostcert_request.pem"파일에 들어있는 CSR 부분을 copy하여 host certificate 요청 페이지를 통하여 upload하면 된다. 이렇게 요청한 certificate는 WACC에 들어있는 e-mail로 그 결과가 통보된다.

7) 설치 상황 점검

"ldg-server-version" 명령어를 통하여 현재 설치되어 있는 LDG-server의 설치 내용을 확인할 수 있다.

```
CentOS5.3 LIGO Data Grid Server version 5.0

Globus Toolkit:                4.2.1
GSI-OpenSSH:                   5.3p1-hpn13v6
MyProxy (client):              5.0
grid-ftp:                       3.17
globus-simple-ca:              4.2.1r3
OSG CA Certificates:           1.13
```

4. Configuring and Deploying Condor

LSC DataGrid Server를 기본으로 하는 VDT server가 설치되어 질 때에 해당 Linux 장비에 Condor는 기본으로 설치되지 않는다.

Redhat Enterprise 계열의 linux 버전에서 설치를 지원하는 yum을 통한 condor 설치 방법은 배포 버전 Condor 7.4.1과 개발 버전 Condor 7.5.1부터 제공되기 시작하였다. 그전에는 Source나 OS별 설치를 지원하는 실행 파일만이 제공되고 있었다. 이러한 package 형태의 설치 방법 지원은 Debian OS에서도 지원하고 있다.

Condor를 설치하기 위해서는 OS 별로 다르지만, 대략 50MB 이상 필요하며 Condor를 통하여 job을 submit하기 위해서는 local directory에 많은 disk 공간이 필요하다.

Scientific Linux 5는 RHEL5 기준으로 CentOS 5와 호환되며, RHEL5를 중심으로 기술하도록 한다.

가. YUM을 통한 Condor 기본 설치

Condor 설치에 필요한 파일은 각종 OS 버전에 맞게 다양하게 존재하는데, 그 중 yum을 통하여 Condor를 설치하는 방법을 살펴보도록 한다. Condor yum 설치에 대해서는 <http://www.cs.wisc.edu/condor/yum>을 참고하면 보다 세부적으로 파악할 수 있다.

1) Condor YUM repository 설치하기

먼저, repository 파일은 wget을 통하여 condor의 공식 사이트를 통하여 받을 수 있다. RHEL5의 경우에는 개발 버전과 배포 버전을 모두 받을 수 있으며, RHEL4의 경우에는 배포 버전만 받을 수 있다.

```
cd /etc/yum.repos.d
wget http://www.cs.wisc.edu/condor/yum/repo.d/condor-development-rhel5.repo
wget http://www.cs.wisc.edu/condor/yum/repo.d/condor-stable-rhel5.repo
```

2) Condor User 추가하기 (Option)

Condor package는 'condor'이름의 id/group이 해당 서버에 존재하지 않는다

면, 자동으로 'condor' 이름의 id/group을 추가한다. 따라서, id/group 관리에 민감한 사이트라면 'condor' 라는 이름의 id/group을 추가하도록 한다.

나. Tarball을 통한 Condor¹⁰⁾ 설치

1) condor package 내려 받기 및 구성

condor는 [condor](http://www.cs.wisc.edu/condor/downloads-v2/download.pl) download (<http://www.cs.wisc.edu/condor/downloads-v2/download.pl>) 페이지를 통하여 해당 OS 버전에 맞는 패키지를 먼저 download 한 후에 설치를 진행하도록 한다.

OS 버전에 맞는 Condor 패키지 (condor-7.4.2-linux-x86_64-rhel5.tar.gz)를 적당한 곳에 내려 받은 후에 패키지를 푼다.

```
tar xzf condor-7.4.2-linux-x86_64-rhel5.tar.gz
```

tar로 압축되어 있는 condor binary 패키지 파일은 대략 400MB정도의 크기이며, 설치를 위해 압축을 해제한 condor 파일의 크기는 약 1.1GB에 이를 정도로 크므로 사전에 충분한 크기의 저장 공간을 확보하도록 한다.

또한, CentOS에 설치한 condor 의 전체 용량도 1.1GB 이므로 condor를 설치하는데 필요한 충분한 저장 공간을 확보하여야 하며, /log, /spool 디렉토리로 지정할 scratch 공간도 충분히 확보하여 설치를 진행하도록 한다.

2) CentOS 상의 condor 설치 과정

tar 패키지를 해제한 후 condor 디렉토리 안에서 condor_install 명령어를 통하여 condor를 설치한다. 아래 내용은 condor_install 명령어를 통하여 condor를 설치할 때 나오는 내용이다.

10) http://www.cs.wisc.edu/condor/manual/v7.4/3_2Installation.html

향후 매뉴얼 순서대로 보강하기

```
[root@ldg !VM! condor-7.4.2]# ./condor_install --prefix=/opt/ldg/condor W
--local-dir=/home/condor/ --type=manager

Installing Condor from /opt/condor/condor-7.4.2 to /opt/ldg/condor
WARNING: Multiple network interfaces detected. Condor might not work
properly until you set NETWORK_INTERFACE = <interface IP>

Condor has been installed into:
    /opt/ldg/condor

Configured condor using these configuration files:
    global: /opt/ldg/condor/etc/condor_config
    local: /home/condor//condor_config.local

In order for Condor to work properly you must set your CONDOR_CONFIG
environment variable to point to your Condor configuration file:
/opt/ldg/condor/etc/condor_config before running Condor commands/daemons.
Created scripts which can be sourced by users to setup their
Condor environment variables. These are:
    sh: /opt/ldg/condor/condor.sh
    csh: /opt/ldg/condor/condor.csh
```

[그림 7] Condor 설치 과정 모습

condor를 설치할 때의 필요한 option의 의미는 다음과 같다.

1. --install-dir=directory 혹은 --prefix=directory
condor가 설치되는 local 디렉토리에 대한 패스를 정의한다.
2. --type=manager
condor pool 안에서 해당 서버가 취해야 하는 한 개 이상의 역할을 정의한다. central manager는 submit 하는 역할을 수행하거나 job을 실행하는 역할도 가능한데 이 경우 선택할 수 있는 type option으로는 “manager”, “manager, execute”, “manager, submit”, 혹은 “manager, submit, execute” 중에 선택할 수 있다.
3. --local-dir=directory
local 디렉토리에 대한 path를 정의한다. local-dir는 condor가 동작하면서 쌓이게 되는 각종 정보에 대한 기록을 갖고 있다.

설치한 condor에 submit 기능을 추가하고자 한다면, 다음과 같이 "condor_configure" 설치 파일을 이용하여 내용을 갱신할 수 있다.

```
% condor_configure --prefix=~condor W
--local-dir=/scratch/condor --type=manager,submit
```

위의 [그림 7]의 설치 결과에 따라 condor가 설치되는 디렉토리는

“/opt/ldg/condor”이며, 여기에 설치된 내용은 다음 [그림 8]과 같다. 설치된 Condor 디렉토리에 설치된 “condor.sh”파일은 아래 내용과 같이 condor path를 지정하기 위한 환경 변수 설정 방법이 들어있다.

```
# The script should be sourced by /bin/sh or similar
CONDOR_CONFIG="/opt/ldg/condor/etc/condor_config"
export CONDOR_CONFIG
PATH="/opt/ldg/condor/bin:/opt/ldg/condor/sbin:$PATH"
export PATH
```

tar 패키지를 통하여 condor를 설치하면 5개의 기본적인 하위 디렉토리 (bin, etc, lib, sbin, libexec)가 존재한다.

1. 사용자 실행 파일 디렉토리 (/bin)

Condor 사용자가 사용하는 실행 파일을 담고 있다. 이 파일을 사용하기 위해서는 PATH 사용자 변수에 해당 디렉토리 패스가 포함되어 있어야 한다.

2. 시스템 실행 파일 디렉토리 (/sbin)

sbin 디렉토리에 있는 모든 파일은 Condor daemon 이나 agent이다. 해당 시스템 파일은 condor 관리자가 실행해야 하므로, 관리자의 PATH 사용자 변수에 해당 디렉토리 패스를 지정하도록 한다.

3. private condor binaries (/libexec)

Condor 자체에서 실행해야 하는 프로그램을 모아놓은 것으로, 절대로 수동으로 실행하면 안된다.

4. 라이브러리 디렉토리 (/lib)

The files in the lib directory are the Condor libraries that must be linked in with user jobs for all of Condor's checkpointing and migration features to be used. lib also contains scripts used by the condor_compile program to help re-link jobs with the Condor libraries. These files should be placed in a location that is world-readable, but they do not need to be placed in anyone's PATH. The condor_compile script checks the configuration file for the location of the lib directory.

5. 기본 정도 디렉토리 (/etc)

etc 디렉토리는 config_config파일과 examples 디렉토리를 갖고 있다. examples 디렉토리는 다양한 환경 설정 예제 파일과 Condor를 설치하기 위해 사용되는 파일을 가지고 있다. etc 디렉토리는 condor를 운영하기 위한 환경 설정 파일에 대한 원본을 갖고 있는 것을 권장하며, condor의 global 환경 설정 파일을 지정한 곳 중 한곳으로 부터의 softlink를 연결할 수 있다.

```

drwxrwxrwx 11 root root 4096 Jul 19 20:49 ./
drwxr-xr-x  3 root root 4096 Jul 19 20:48 ../
drwxr-xr-x  2 root root 4096 Mar 30 12:49 bin/
-rw-r--r--  1 root root  172 Jul 19 20:49 condor.csh
-rw-r--r--  1 root root  190 Jul 19 20:49 condor.sh
drwxr-xr-x  3 root root 4096 Jul 19 21:13 etc/
drwxr-xr-x  2 root root 4096 Mar 30 12:50 include/
drwxr-xr-x  6 root root 4096 Mar 30 12:50 lib/
drwxr-xr-x  3 root root 4096 Mar 30 12:50 libexec/
drwxr-xr-x  3 root root 4096 Mar 30 12:33 man/
drwxr-xr-x  2 root root 4096 Mar 30 12:50 sbin/
drwxr-xr-x  2 root root 4096 Mar 30 12:47 sql/
drwxr-xr-x  5 root root 4096 Mar 30 12:33 src/

```

[그림 8] Condor가 설치된 디렉토리 내용

3) Condor 프로그램 설명

condor_master: 이 프로그램은 계속적으로 동작하며 모든 condor 프로그램이 계속 동작할 수 있도록 한다. 만약, condor의 다른 프로그램이 죽거나 멈춰있으면 condor_master 프로그램이 다시 실행시키도록 한다.

condor_schedd: condor_schedd 프로그램이 실행되고 있는 서버에서 job을 submit할 수 있도록 한다. 즉, 해당 서버가 "submit machine"이 되는 것이다. condor_schedd 프로그램은 제출된 job을 central manager에게 알려서 제출된 job에 대한 정보를 central manager와 공유한다. 또한, 제출된 job을 실행하는 실행 장비에서 실행되는 condor_startd 프로그램과 통신한다.

condor_procd: Condor에서 실행하는 job을 추적한다.

condor_collector: condor central manager의 한 부분이다. 이 프로세스는 pool에 있는 모든 서버의 정보를 수집하며, job을 실행시키고자 하는 사용자의 정보를 수집한다.

```

% condor_config_val COLLECTOR_HOST
ldg.sdfarm.kr

```

condor_negotiator: Condor central manager의 한 부분이다. 이것은 어떠한 job이 어디에서 실행되는지를 결정한다. 이것은 collector와 같은 서버에서 동작한다.

condor_startd: 이 프로그램이 동작한다면, 제출된 job이 해당 서버에서 실행될 수 있도록 허용한다. 즉, 해당 서버는 "execute machine"으로 동작하는 것이다. condor_start 프로세스는 해당 서버에 대한 정보를 central manager에게 알려줌으로써 central manager가 관리할 수 있도록 허용한다. 그리고, 제출된 job을 실행

행시킨다.

condor_shadow: worker node로 제출된 잡 각각에 대하여 하나의 condor_shadow 프로세스가 동작한다. 마치, 원격에서 동작하는 것처럼 job을 관찰한다. 어떤 경우에는 이 프로세스는 어느 정도의 도움을 제공한다. 이 프로세스는 해당 컴퓨터의 동작 상태에 따라 보이거나 또는 그렇지 않게 된다.

4) Condor Server (Master) Node 환경 설정¹¹⁾하기

Condor Master에 설치된 /opt/ldg/condor 파일 모두를 다른 cluster node에 배포하여야 한다.

1. condor ID를 생성한 후, /etc/passwd, /etc/group에 condor가 생성되었음을 확인하고 각각의 uid, gid를 확인하도록 한다. condor ID는 별도의 shell을 필요로 하지 않으며 존재만 하면 된다.

```
# useradd condor
```

2. condor globla 환경 설정 파일인 /opt/ldg/condor/etc/condor_config 파일 내의 환경 설정을 수정한다.

```
CONDOR_HOST = ldg.sdfarm.kr
RELEASE_DIR = /opt/ldg/condor
LOCAL_DIR = /opt/ldg/condor/home
CONDOR_ADMIN = jsu@kisti.re.kr
UID_DOMAIN = sdfarm.kr
FILESYSTEM_DOMAIN = sdfarm.kr
CONDOR_IDS = 50522.50522 <- condor user의 uid, gid 값 지정
```

3. /opt/ldg/condor/etc/examples/condor.boot 파일의 환경 변수를 수정한다.

```
MASTER=/opt/ldg/condor/sbin/condor_master
```

4. 이 단계에서 Condor 폴더를 Condor Cluster node에 배포한다.
5. condor init file을 /etc/init.d/condor로 copy한 후, booting시 실행할 수 있도록 설정한다.

11) <https://www.lsc-group.phys.uwm.edu/lscdatagrid/doc/condordeploy.html>를 참고할 것.

```
# cp /opt/ldg/condor/etc/examples/condor.boot /etc/init.d/condor
# chkconfig --add condor
# chkconfig --level 235 condor on
```

6. condor의 global configuration file을 /etc/condor/ 디렉토리에 생성한다.

```
# mkdir /etc/condor
# ln -s /opt/ldg/condor/etc/condor_config /etc/condor/
```

7. Condor local 디렉토리에 대한 owner-ship을 변경한다.

```
# cd /opt/ldg/condor
# chown -R condor:condor home
```

8. condor local 환경 설정 파일인 condor_config.local 파일 내의 환경 설정을 수정한다.

```
COLLECTOR_NAME = ldg.sdfarm.kr <- Condor Server 이름
COLLECTOR = $(SBIN)/condor_collector
NEGOTIATOR = $(SBIN)/condor_negotiator
NETWORK_INTERFACE = 134.75.123.59
```

9. condor를 실행시킨다.

```
# service condor start
```

5) Cluster node에 Condor 설치하기

Condor Master에 설치한 Condor 설치 파일들을 이용하여 설치하도록 한다.

1. condor 설치 파일을 tar file로 만든 다음에 cluster node로 copy 한다. 압축된 Condor 설치 파일은 대략 350MB에 이르며, 실제 설치 파일은 1.1GB 정도 크기를 가지고 있으므로 디스크 여유 공간이 충분한 곳에서 진행하도록 한다.

```
# tar cfpz condor.tgz /opt/ldg/condor
# scp condor.tgz wn1003:/var
```

2. 각 cluster node에 condor ID를 생성한 후, /etc/passwd, /etc/group에 condor가 생성되었음을 확인하고 각각의 uid, gid를 확인하도록 한다. condor ID는 별도의 shell을 필요로 하지 않으며 존재만 하면 된다.

```
# tar xpfz condor.tgz -C /
# useradd condor
```

3. condor init file을 /etc/init.d/condor로 copy한 후, booting시 실행할 수 있

도록 설정한다.

```
# cp /opt/ldg/condor/etc/examples/condor.boot /etc/init.d/condor
# chkconfig --add condor
# chkconfig --level 235 condor on
```

4. condor의 global configuration file을 /etc/condor/ 디렉토리에 생성한다.

```
# mkdir /etc/condor
# ln -s /opt/ldg/condor/etc/condor_config /etc/condor/
```

5. Condor local 디렉토리에 대한 owner-ship을 변경한다.

```
# cd /opt/ldg/condor
# chown -R condor:condor home
```

6. local 환경 설정 파일 내용을 변경해준다.

```
# vi /opt/ldg/condor/home/condor_config.local
CONDOR_IDS = condor.condor <- 여기에 uid.gid를 넣어야 함
DAEMON_LIST = MASTER, STARTD <- STARTD 확인 필요
```

7. /opt/ldg/condor/bin/condor_status 명령어를 통하여 연동된 condor 환경 설정을 통하여 정상 동작여부를 확인할 수 있다. condor_status 명령을 통하여 condor server나 cluster node에서 모두 같은 결과를 얻을 수 있다. 다음은 condor_status 명령에 따른 그 결과물이며, CPU socket이 아니 Core 당 computing slot이 할당됨을 볼 수 있다.

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@wn1003.sdfar	LINUX	X86_64	Owner	Idle	0.000	2006	0+00:00:08
slot2@wn1003.sdfar	LINUX	X86_64	Owner	Idle	0.000	2006	0+00:00:09
slot3@wn1003.sdfar	LINUX	X86_64	Owner	Idle	0.000	2006	0+00:00:10
slot4@wn1003.sdfar	LINUX	X86_64	Owner	Idle	0.000	2006	0+00:00:11
slot5@wn1003.sdfar	LINUX	X86_64	Owner	Idle	0.000	2006	0+00:00:12
slot6@wn1003.sdfar	LINUX	X86_64	Owner	Idle	0.000	2006	0+00:00:13
slot7@wn1003.sdfar	LINUX	X86_64	Owner	Idle	0.000	2006	0+00:00:14
slot8@wn1003.sdfar	LINUX	X86_64	Owner	Idle	0.000	2006	0+00:00:07
slot1@wn1004.sdfar	LINUX	X86_64	Owner	Idle	0.000	2006	0+00:15:09
slot2@wn1004.sdfar	LINUX	X86_64	Unclaimed	Idle	0.000	2006	0+00:00:05
slot3@wn1004.sdfar	LINUX	X86_64	Owner	Idle	0.000	2006	0+00:10:11
slot4@wn1004.sdfar	LINUX	X86_64	Unclaimed	Idle	0.000	2006	0+00:00:07
slot5@wn1004.sdfar	LINUX	X86_64	Unclaimed	Idle	0.000	2006	0+00:00:08
slot6@wn1004.sdfar	LINUX	X86_64	Unclaimed	Idle	0.000	2006	0+00:00:09
slot7@wn1004.sdfar	LINUX	X86_64	Unclaimed	Idle	0.000	2006	0+00:00:10
slot8@wn1004.sdfar	LINUX	X86_64	Unclaimed	Idle	0.000	2006	0+00:00:03
Total Owner Claimed Unclaimed Matched Preempting Backfill							
X86_64/LINUX	8	8	0	0	0	0	0
Total	8	8	0	0	0	0	0

6) Condor 환경 설정 과정에서 추가로 살펴봐야할 내용

ALLOW_WRITE 변수를 모든 condor Pool node에 설정해줘야 함 (예, *.sdfarm.kr).

LOCAL_DIR(LDG의 경우 /opt/ldg/condor/home)로 지정되어 있는 패스의 owner 가 제대로 되어 있는지 확인 할 것.

condor_config.local 파일은 condor_config 광역 환경 설정 파일에서 지정된 것을 우선시 하게 된다.

condor server와 condor cluster node간에 통신에 대해 iptables에서 허용해 줘야 함.

5. 12) LIGO Tools S/W 설치하기

LIGO를 연구하는 연구 그룹의 연구 편의성을 위해서 여러 패키지로 구성되어 있는 LIGO Tool을 구성하여 사용하고 있다. LIGO Tool 소프트웨어를 설치하여 사용하는 것은 굳이 root가 진행할 필요는 없으며, LIGO 연구진 중 한 명이 정해진 곳에 해당 패키지들을 설치하고 지속적으로 새로운 버전으로 update 하는 것을 권장하고 있다. 예를 들면, Caltech에서는 peter가 LIGO Tool을 관리하며, /ligoapps/ligotools 아래에서 관리하고 있다.

실제 설치되어 있는 내용은 4개의 LDG 서버 모두 같은 구조로 구성되어 있어 /ligotools 이름을 갖는 디렉토리 안에 설치되어 있다.

설치해야 하는 LIGOTOOLS Software package는 다음과 같다.

[표 1] LIGO Tools Software Packages 목록

Package	Description
Fbe*	C library and utilities to generate frame files from simpler formats
Fr	C library and utilities in core Virgo distribution to read/write data in frame format
FrContrib	Additional utilities for working with frame files, exclusive of core Virgo distribution
Frv*	Companion to Fr package to manipulate FrVect objects
basegdsroot*	Basic container and graphics libraries for use within ROOT
dataflow	Raw data and metadata access utilities
dataviewer	Offline version of the familiar control-room GUI tool
detgeom	Matlab routines to define and manipulate detector geometry
dmtrout*	Data Monitoring Tool libraries built for use within ROOT
dt	Diagnostic Test Tools suite of graphical data-analysis programs
eventtool*	The LIGO Event Analysis Tool software, which is used within ROOT
fftw*	The FFTW package for Fast Fourier Transforms (used by LAL and Frv)
guild	Graphical User Interface to LIGO Databases
httptools	Simple utilities to retrieve files via http
ilwdread	Matlab script to read an ilwd file
lal*	The LIGO/LSC Algorithm Library
lalapps*	A collection of utilities based on LAL
ldasjob	High-level interface for running LDAS jobs from Tcl scripts
ligo-viewer*	Graphical user interface to view LIGO trend data
ltadmin	LIGOTOOLS administration utilities
medmguide	Graphical user interface to examine EPICS medm (*.adl) files
metaio	C library and utilities to read and manipulate LIGO_LW table files
mininds*	Makes data on disk available via NDS protocol (used by the dt package)
papers	Assorted scripts to help prepare papers for publication

12) <http://www.ldas-sw.ligo.caltech.edu/ligotools/>

root*	The object-oriented data analysis and display environment developed at CERN
runtools	Summarize status of interferometers during science/engineering runs
segments	Generate and manipulate lists of GPS time intervals
tclexe	Pre-built, standalone versions of Tcl/Tk shells (used by certain other LIGOTOOLS packages)
time	GPS/UTC/local time conversion utilities, plus a GPS clock
wat*	Wavelet Analysis Tool library and utilities for lossy compression of frame files

위 테이블의 package 이름의 *는 `ligotools_update`에 의해 설치되지 않으나, 수동으로 내려받기를 통한 설치를 해야 한다.

가. LIGOTOOLS Initial Setup

LIGO Tool 이 제대로 동작하도록 하기 위해서는 설치 지시를 정확히 따라야 한다. 다음의 Part 1, Part 2은 반드시 설치하여야 하며, Part 3는 설치를 권장하는 부분이다.

1) Part 1: Setup the LIGOTOOLS directory structure

`wget`을 이용하여 `ligotools` initial package를 내려 받는다. 이때, 내려 받는 위치는 별로 중요하지 않다. 내려 받은 파일 패키지를 풀어서 다음과 같이 설치를 시작한다.

```
tar xmf ligotools_init_2.4.tar
./ligotools_init
```

이와 같이 설치하면, 기본으로 다음의 디렉토리가 생성되며, 설치 중에 그 이름을 변경할 수도 있다.

```
drwxr-xr-x+ 10 install ldas 10 Jun 9 2009 ./
drwxr-xr-x 37 root root 4096 Sep 14 08:37 ../
drwxr-xr-x+ 2 install ldas 78 Jul 1 2009 bin/
drwxr-xr-x+ 3 install ldas 4 May 28 2002 config/
drwxr-xr-x+ 2 install ldas 21 Jul 1 2009 include/
drwxr-xr-x+ 2 install ldas 2 May 24 2002 java/
drwxr-xr-x+ 2 install ldas 10 Jul 1 2009 lib/
drwxr-xr-x+ 2 install ldas 27 Jul 1 2009 matlab/
drwxr-xr-x+ 24 install ldas 26 Oct 28 2005 packages/
drwxr-xr-x+ 2 install ldas 5 Jan 31 2005 root/
```

이후 모든 사용자들은 LIGOTOOL을 사용하기 위해서는 다음의 명령어를 shell에서 실행시켜야 한다. 이 명령어를 통하여 LIGOTOOLS에서 필요한 환경 변수를 설정할 수 있으며, PATH를 지정할 수 있다.

```
eval `<dir>/bin/use_ligotools`
```

여기에서 <dir>는 LIGOTOOLS를 설치한 주요 폴더로 현재는 /data/ligo/ligotools이다.

이 초기 LIGOTOOLS setup은 ltadmin package를 포함하고 있는데, 이 package는 LIGOTOOLS 디렉토리 구조를 관리하는 몇 가지 script를 제공한다. 가장 많이 사용할 것으로 보이는 명령어는 ligotools_list인데, 이것은 현재 설치되어 있는 package 리스트를 보여준다. 그리고, 많이 사용하는 명령어로 ligotools_update가 있는데 이것은 Part 2에서 설명한다.

초기 설치후 ligotools_list를 실행하였을 때의 모습은 다음과 같다.

```
LIGOTOOLS is /data/ligo/ligotools
Package      Active      Other installed versions
-----
ltadmin      2.4
```

2) Part 2: Install Packages

Part 1에서 LIGOTOOLS의 initial 패키지(ltadmin)를 설치하였으므로, 이제 나머지 다양한 LIGOTOOLS package를 설치하여야 한다. 이중, 어떤 package들은 서로 같이 동작하여야 하여야 하므로 반드시 같이 설치하여야 한다. 가급적 모든 LIGOTOOLS package를 설치하여야 하지만, 모든 것을 설치하기 어려운 상태라면 root, lal, dtt package는 제외하고 설치해도 된다. 그렇지만, 사실 모든 것을 설치하는 방법이 선택적인 설치 방법보다 훨씬 수월하다.

- 가) 먼저 LIGOTOOLS shell 환경 변수가 제대로 설정되어 있는지 확인한다. 또한, PATH 환경 설정 변수에 LIGOTOOLS bin 디렉토리가 포함되어있어야 한다.
- 나) LIGOTOOLS tclexe¹³⁾ 홈페이지에서 OS에 맞는 버전을 내려 받는다. 이때, 내려받은 파일 이름은 홈페이지에 적혀있는 파일 이름과 정확히 같은지 확인한후에 다음과 같이 설치를 진행한다. 이때, ligotools_install 명령어를 찾을 수 없다는 에러가 출력되면, 사용하는 shell이 csh, tcsh이므로 rehash 명령어를 통하여 path 지정을 새로 받아서 설치를 진행한다.

```
ligotools_install <file>
```

- 다) ligotools_update를 통하여 설치할 패키지를 자동 설치한다. 이때, 설치할

13) <http://www.ldas-sw.ligo.caltech.edu/ligotools/tclexe>

패키지들에 대한 설치 여부에 대한 질문이 나올 것이며, 활성화 할 것인가에 대한 질문이 나올 것이다. 이때, y로 확인하면 된다.

이때, 설치되는 패키지는 다음과 같다.

```
Fr_v8r08_SOURCE.tar.gz
FrContrib_v8r08_SOURCE.tar.gz
dataflow_7.2_ALL.tar.gz
detgeom_0.3_ALL.tar.gz
guild_4.0_ALL.tar.gz
httptools_6.0_ALL.tar.gz
ilwread_0.2_ALL.tar.gz
ldasjob_3.0_ALL.tar.gz
medmguide_2.1_ALL.tar.gz
metaio_7.2_SOURCE.tar.gz
papers_1.0_ALL.tar.gz
runtools_5.0_ALL.tar.gz
segments_3.1_ALL.tar.gz
time_6.0_SOURCE.tar.gz
```

ligotools_update는 필요한 정보를 얻어오기 위해서 http protocol을 사용한다. 이때, “Error getting list of packages” 메시지가 나온다면, 방화벽에 의해 필터링 되고 있음을 의미하므로, 가장 먼저 방화벽을 의심하도록 하며, 만약 proxy server를 운영하여 사용하여야 한다면, HTTPPROXY 환경 변수를 이용하여 설치할 수 있다. 이때 Proxy 설정하는 구체적인 방법은 홈페이지¹⁴⁾를 참조하도록 한다.

설치가 마무리 되면, 올바르게 설치되었는지를 확인하는 방법은 다음과 같다.

- 가) 'Fr' package에 대한 확인을 위해 “FrDump” 명령어를 사용하면 사용 방법에 대한 내용이 출력된다.
- 나) 'dataflow' package에 대한 확인을 위해 “getMeta” 명령어를 사용하면 사용 방법에 대한 내용이 출력된다.
- 다) 'dtt' package는 현재는 solaris에서만 동작하며 “dtt” 명령어를 사용하면 창이 활성화된다.
- 라) 'guild' package에 대한 확인을 위해 “guild” 명령어를 사용하면, GUI window가 팝업 된다.

설치한 패키지들에 대한 새로운 패키지들과 새로운 버전을 살펴보기 위해서는 주기적으로 ligotools_update 명령어를 통하여 확인할 수 있으며, 이것을 통하여 새로운 내용을 설치할 수 있다. 또한, "LIGOtools Administration Guide"¹⁵⁾를 통

14) <http://www.ldas-sw.ligo.caltech.edu/ligotools/faq/ljproxy.html>

15) http://www.ldas-sw.ligo.caltech.edu/ligotools/setup_info.html

하여 그 사용 방법을 익힐 수 있다.

3) Part 3: Set up environment variables at login time

LIGOTOOLS 환경 변수를 지정하는 방법으로는 shell 시작 파일 (.login, .bashrc, cshrc 등)에 "eval `<dir>/use_ligotools`" 명령어를 설정하는 것이다. 여기에서 <dir>는 use_ligotools script를 포함하는 full path로, 일반적으로 \$LIGOTOOLS/bin에 해당한다. 따라서, 환경 변수에 \$LIGOTOOLS가 LIGOTOOLS가 설치되어 있는 디렉토리를 정확히 포함하는 것이 중요하다.

나. 추가 Software 패키지 설치하기

LIGOTOOLS 자동 설치 패키지를 통하여 설치된 패키지 외에 다음의 패키지를 추가로 설치하여야 한다. 이때, 설치하는 방법은 해당 distribution file을 내려받기한 후에 "ligotools_install <file>" 명령어를 통하여 설치할 수 있다.

[표 2] 수동으로 추가해야할 LIGO Tools Software Packages 목록

Package	Description
Fbe	C library and utilities to generate frame files from simpler formats
Frv	Companion to Fr package to manipulate FrVect objects
basegdsroot	Basic container and graphics libraries for use within ROOT
dmtroot	Data Monitoring Tool libraries built for use within ROOT
dt**	Diagnostic Test Tools suite of graphical data-analysis programs
eventtool	The LIGO Event Analysis Tool software, which is used within ROOT
fftw	The FFTW package for Fast Fourier Transforms (used by LAL and Frv)
lal	The LIGO/LSC Algorithm Library
lalapps	A collection of utilities based on LAL
ligo-viewer	Graphical user interface to view LIGO trend data
mininds**	Makes data on disk available via NDS protocol (used by the dt** package)
root	The object-oriented data analysis and display environment developed at CERN
wat	Wavelet Analysis Tool library and utilities for lossy compression of frame files

위의 [표 2]의 추가 패키지 중 "dt**", "mininds"는 Solaris에서만 구현된 패키지로 Linux나 기타 버전에서는 별도로 제공되지 않는다.

다음은 추가로 설치해야할 LIGOTOOLS 패키지 Linux 버전의 내려받기 Path를 정리하여 놓은 표이다.

[표 3] 수동으로 추가해야 할 LIGO Tools Software Packages URL 목록

Package	Description
Fbe	http://www.ldas-sw.ligo.caltech.edu/ligotools/Fbe/v1r2c/Fbe_v1r2c_Linux.tar.gz
Frv	http://www.ldas-sw.ligo.caltech.edu/ligotools/Frv/v4r00/Frv_v4r00_Linux.tar.gz
basegdsroot	http://www.ldas-sw.ligo.caltech.edu/ligotools/basegdsroot/2002.4.9/basegdsroot_2002.4.9_Linux.tar.gz
dmtrout	http://www.ldas-sw.ligo.caltech.edu/ligotools/dmtrout/2002.4.9/dmtrout_2002.4.9_Linux.tar.gz
eventtool	http://www.ldas-sw.ligo.caltech.edu/ligotools/eventtool/2002.4.9/eventtool_2002.4.9_Linux.tar.gz
fftw	http://www.ldas-sw.ligo.caltech.edu/ligotools/fftw/2.1.3b/fftw_2.1.3b_Linux.tar.gz
lal	http://www.ldas-sw.ligo.caltech.edu/ligotools/lal/3.0/lal_3.0_Linux.tar.gz
lalapps	http://www.ldas-sw.ligo.caltech.edu/ligotools/lalapps/3.0/lalapps_3.0_Linux.tar.gz
ligo-viewer	http://www.ldas-sw.ligo.caltech.edu/ligotools/ligo-viewer/0.5.0b/ligo-viewer_0.5.0b_Linux.tar.gz
root	http://www.ldas-sw.ligo.caltech.edu/ligotools/root/3.02.07a/root_3.02.07a_Linux.tar.gz
wat	http://www.ldas-sw.ligo.caltech.edu/ligotools/wat/1.62/wat_1.62_Linux.tar.gz

최종적으로 Linux에서 설치한 LIGOTOOLS software package들의 목록과 설치된 버전을 살펴보면 다음과 같다.

```

LIGOTOOLS is /data/ligo/ligotools
Package      Active      Other installed versions
-----
Fbe          v1r2c
Fr           v8r08
FrContrib   v8r08
Frv         v4r00
basegdsroot 2002.4.9
dataflow    7.2
detgeom     0.3
dmtrout     2002.4.9
eventtool   2002.4.9
fftw        2.1.3b
guild       4.0
httpptools  6.0
ilwdread    0.2
lal          3.0
lalapps     3.0
ldasjob     3.0
ligo-viewer 0.5.0b
ltadmin     2.4
medmguide   2.1
metaio      7.2
papers      1.0
root        3.02.07a
runttools   5.0
segments    3.1
tclexe      8.4.7
time        6.0
wat         1.62

```

6. OCTAVE Software 설치하기

Octave를 설치 하는 방법으로는 source 파일을 받아서 컴파일하는 방법과 binary 파일을 받아서 설치하는 방법이 있다.

이때, octave를 설치하기 위해서는 먼저 다음의 software의 최신 버전이 최신 버전이 필요하다.

```
* To compile Octave, you will need a recent versions of
* the following software:
*
* GNU Make (a recent version)
*
* g++ (preferably a recent 4.x version)
*
* flex (2.5.4 or a more recent version) -- required if
* you need to recreate lex.cc from lex.l
*
* bison (1.31 or a more recent version) -- required if
* you need to recreate parse.cc from parse.y
*
* gperf (3.0.1 or a more recent version) -- required if
* you need to recreate oct-gperf.h from octave.gperf
*
* Now would be a good time to read INSTALL.OCTAVE if
* you have not done so already.
```

그 이외 설치해야할 Software는 다음과 같다. 필요한 Software를 설치하는 것은 yum을 이용하여 설치하면 쉽게 설치할 수 있다.

```
OCTAVE를 설치할 때 추가로 필요한 Software Package
- gcc-gfortran
- Readline
- Readline-devel
- ncurses
- ncurses-devel
- libtermcap-devel
```

configure 실행 파일을 통해 설치 환경을 점검할 수 있다.

7. LAL and LALApps Software 설치하기

LAL (LSC Algorithm Library)은 데이터 분석 루틴을 포함하고 있다. 알고리즘은 높은 이식성을 지원하기 위하여 ANSI C89로 구현되어 있으며 LAL을 구성하는 Routine(함수)은 LSC member가 제공하고 있다. LAL algorithm은 Data conditioning, gravitational wave simulation, correlation of data with binary insipral signals, excess power filter등에 대한 기능을 포함하고 있다.

LALApps는 LIGO 파이프라인의 일부분으로 수행되는 데이터 처리 및 분석 연구를 위해서 필요한 사용자 어플리케이션이다. LALApps는 LAL이 제공하는 기능 위에서 동작하는 stand-alone scientific data analysis 어플리케이션들로 구성되어 있는 뭉치이다. LALApps는 data procuton을 유지 관리하는 것 뿐만 아니라 inteferomter를 분석하는데 사용하는 프로그램을 포함한다. 또한, Glue (Grid LSC User Environment)와 함께 파이프라인을 구축하도록 하는 scientific application들을 추상화하기 위한 코드를 포함한다.

LSC의 LSC-Soft Package에 이미 공통의 LAL과 LALApps 패키지가 포함되어 LDG Server 구성할 때 LSC-Soft Package가 설치되어 있다. 하지만, 이는 public하게 배포되어 있는 것으로 LSC에서 표준으로 사용하고 있는 버전과 반드시 일치하는 것은 아니다. 설치된 LAL, LALApps 패키지는 데이터가 기록된 GPS 시간에서 날짜로 변경하거나 XML 파일을 다루는 등 Shell command로 손쉽게 사용할 수 있다. 하지만, LAL suite 자체에는 몇 개의 branch가 존재하는데 데이터가 수집된 시기에 따라 다르게 유지보수 되고 있다. 이러한 것들을 개별 사용자가 별도로 만들어 사용해야 하는 것이다. 그리고, LIGO 자료 분석에 참여해서 일을 하다보면 본인이 직접 코드를 수정하고 개발해야하는 경우가 있으며, 수시로 패치가 나오므로 이러한 개개의 것들이 public release에 포함되기는 어렵기 때문에 별도로 개별 연구자들이 LAL, LALApps를 설치해서 연구에 활용할 필요가 있다.

가. 설치 준비 및 사전 요구사항

Git (the fast version control system) repository에서 LAL과 LALApps를 설치하기 위한 과정은 웹페이지¹⁶⁾에 단계별로 기술되어 있다. LAL, LALApps 뭉치를 컴파일하여 설치하기 위해서는 컴파일러와 여러 라이브러리가 포함되어 있는 기본적인 설치 패키지가 필요한데, Cent OS의 경우에는 "Development Tools"와 "Development Libraries"에 포함되어 있다. Development Tools 패키지는

16) <https://www.lsc-group.phys.uwm.edu/daswg/docs/howto/lal-install.html>

automake, gcc, perl, python, debugger 등 핵심 개발 툴을 포함하고 있으며, Development Libraries 패키지의 경우에는 어플리케이션을 개발, 설치하기 위한 핵심 라이브러리들이 포함되어 있다. 해당 패키지는 설치 크기는 OS가 설치되는 시점에 포함되어 있는 설치 구성 항목별에 따라 추가 설치해야 할 크기가 달라진다. 두 개의 패키지를 설치하는 방법은 다음과 같다. 이 때, 패키지 이름에 공란이 포함되어 있기 때문에 반드시 이중 따옴표(")를 포함해야 한다.

```
yum groupinstall "Development Tools" "Development Libraries"
```

yum group 패키지에 포함되어 있는 구성 항목들을 알고자 한다면, yum을 이용하여 groupinfo 옵션을 통하여 다음과 같이 하면 된다.

```
yum groupinfo "Development Tools"  
yum groupinfo "Development Libraries"
```

8. LIGO User 환경

가. LIGO VO 계정 신청하기

<https://my.ligo.org>에서 LIGO Data Analysis나 LIGO의 다른 Data Center로 접속하기 위해서는 먼저 LIGO VO에 계정을 만들어야 한다. 다음 [그림 9]는 my-LIGO 계정을 신청하기 위한 신청서 모습이다.

[그림 9] LIGO 계정 신청서 내용 화면

LIGO VO에 계정을 신청하게 되면, KGWG PI에게 승인 신청 요청이 전달되므로 KGWG의 멤버가 아니라면 ID 발급이 되지 않게 된다. 따라서, KGWG에 해당하는 사람임을 인정받는 형태가 되는 것이다.

나. Digital Certificate 신청하기

Grid Certificate를 얻기 위해서는 한국 내에서는 KISTI Grid Certificate Authority site (<http://ca.gridcenter.or.kr>)를 통하여야 한다. 그 신청 방법은 해당 웹페이지에 자세히 설명되어 있으니 참고하기 바라며, 간략히 설명하며 다음과 같다.

1. 먼저, Grid CA 웹사이트를 방문하여 GRID CA User 신청서를 제출하여 e-mail이나 FAX를 통하여 Grid CA관리자에게 전달한다. 이때, RA (Registration Authority)를 통하여, 신분 인증을 받으면 된다. 보통 RA는 해당 VO의 PI가 담당하거나 RA 역할을 하는 사람이 별도로 존재한다. 이때, RA는 PIN 번호를 할당하게 된다.
2. off-line 형태로 작성된 신청서를 기반으로 하여 GRID CA 관리자는 신청자에게 WACC (Web Access Client Certificate)를 전달한다. WACC은 RA가 할당한 PIN 번호를 이용하여 사용자의 Web Browser에 저장할 수 있다.
3. 이렇게 저장된 WACC을 통하여 GRID Certificate를 요청하면 된다. 이때, WACC이 저장된 Web Browser를 반드시 사용하여야 하며, WACC을 통하여 CSR (Certificate Signing Request)를 CA center 웹 사이트에 올려야 Grid Certificate를 받을 수 있다. CSR을 CA center 웹 사이트에 올린 후에 ca@gridcenter.or.kr에 메일을 보내 Grid Certificate를 요청하면, 수일내에 Grid Certificate를 발급받을 수 있게 된다.
4. Grid Certificate는 <http://ca.gridcenter.or.kr/issued/index.php>를 통하여 발급된 내용을 확인할 수 있으며, 본인의 인증서도 받을 수 있다. 이때, 반드시 CSR를 upload하기 위해 사용하였던 web browser를 사용하여야 한다.
5. 저장한 Grid certificate를 Web browser를 이용하여 인증서를 browser에 등록하며, web browser의 인증서 내보내기 기능을 통하여 개인키를 포함하여 암호화된 인증서 및 개인키를 내보낸다. 이때, 개인키를 포함하는 개인 인증서 파일은 .pfx의 확장자를 갖게 된다.
6. linux에서 .pfx파일로부터 다음과 같은 방법으로 개인키와 공개키를 얻을 수 있다. 이때, .pfx 파일의 암호를 통하여 얻을 수 있다.
 - 가. 인증서 파일을 만들기 위한 방법: `openssl pkcs12 -in user.pfx -clcerts -nokeys -out usercert.pem`
 - 나. 개인키 파일을 만들기 위한 방법: `openssl pkcs12 -in user.pfx -nocerts -out userkey.pem`
 - 다. 개인키 파일을 얻을 때는 별도로 파일 암호를 설정하는 작업을 거쳐야 한다.

```

$ openssl pkcs12 -in user.pfx -clcerts -nokeys -out usercert.pem
Enter Import Password: <.pfx 파일 암호입력>
MAC verified OK

$ openssl pkcs12 -in user.pfx -nocerts -out userkey.pem
Enter Import Password: <.pfx 파일 암호입력>
MAC verified OK
Enter PEM pass phrase: <userkey.pem 파일 암호설정>
Verifying - Enter PEM pass phrase: <userkey.pem 파일 암호설정>

```

7. 위와 같이 얻은 인증서 파일(usercert.pem)과 개인키 파일(userkey.pem)은 본인의 홈디렉토리의 .globus/ 디렉토리에 옮겨놓게 되면 GRID Certificate를 통한 연동 환경을 갖추게 되는 것이다. 이때, 2개의 파일에 대한 access permission은 다음과 같아야 한다.

```

[icani@ldg !VM! ~]$ ls -alF .globus/
total 32
drwxrwxr-x 2 icani icani 4096 Jul 14 18:16 ./
drwx----- 5 icani icani 4096 Jul 22 17:32 ../
-r--r--r-- 1 icani icani 1848 Jul 14 18:15 usercert.pem
-r----- 1 icani icani 1987 Jul 14 18:16 userkey.pem

```

다. LIGO VO Computer Resource 사용 권한 얻기

LSC에 속해있는 Data Center의 컴퓨팅 자원을 사용하기 위해서 LIGO VO 계정 신청(https://voms.phys.uwm.edu/lams/account_form.shtml)을 통하여 LIGO VO 계정을 얻어서 접근할 수 있다. 이때, 반드시 필요한 사항이 VO 가입을 위한 GRID Digital Certificate로, LIGO VO를 통한 여러 LIGO Data Center로의 접근을 하기 위한 기본적인 인증 방법이다.

LIGO VO Computing Resource Account 신청서 내용은 다음 [그림 10]과 같다. 한국 LIGO 그룹은 inspiral 그룹에 속하며 Authorizing Institute 목록에 한국 그룹에 대한 내용이 없으므로 KGWG PI의 정보를 넣어서 제출하면 된다. 또한, GRID Certificate를 기반으로 하는 VO 인증을 거쳐서 해당 LIGO VO computing resource를 사용할 수 있는 권한을 얻게되면, 각각의 Data Center에 대한 login 계정을 발급 받을 수 있으므로, 이때 사용할 login-ID와 선호하는 Shell를 넣어서 제출하면 Resource Request양식에 넣은 e-mail로 계정 생성이 되었음을 알리는 메일을 받게 된다.

이렇게 하여 사용할 수 있는 LIGO Data Grid Computing Resource는 해당 Wiki(<https://www.lsc-group.phys.uwm.edu/daswg/wiki/HowtoGetStarted>)를 통해 접근 정보를 얻을 수 있다. 이때 myLIGO의 ID와 password를 사용하여야만 접근이 가능하다. 사용자 인증에 대한 창은 다음 [그림 11]과 같다. myLIGO ID/password를 통한 wiki 접속을 통하여 사용 가능한 LDG Computing

Resource 현황은 [그림 12]에 나와 있으며, 현재까지는 4개의 Data Center 인 UWM, CIT, LHO, LLO의 컴퓨팅 자원을 사용할 수 있다.

LIGO VO Computer Resource Request Form

(javascript should be enabled)

Personal Information

Full Name:

Institution:

Position:

Working Group:

Phone:

Fax(optional):

E-mail:

Site Access

Current Certificate: (Please copy/paste the output of the command 'grid-cert-info -subject')

Format: /DC=org/DC=doegrids/OU=People/CN=John Doe 123456

Note: To request an account, you must already have a valid Grid Personal Certificate.

Tick if you have an existing account and enter certificate below:

Authorizing Institution:

PI Name:

PI Institution:

PI E-mail Address:

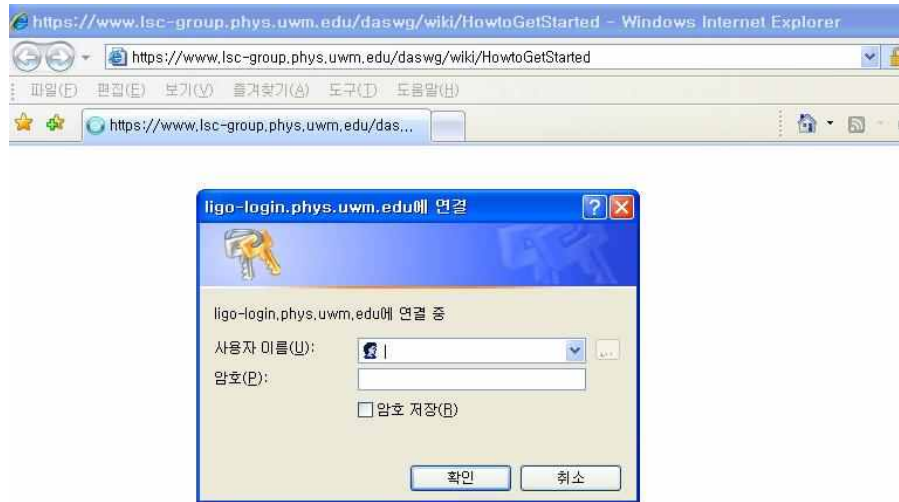
Purpose & Justification

Comments (purpose, special software needs, etc.)

Some sites allow you to choose a preferred username and shell. For those sites, please indicate your preferences here.

Preferred Username: Preferred Shell:

[그림 10] LIGO VO Computer Resource Request 양식



[그림 11] LIGO VO Wiki page 접속시의 인증창

Accessing LIGO Data Grid Computing

The LIGO Data Grid is the distributed computing system used for LIGO data analysis. It consists of autonomous clusters at Caltech, LIGO Hanford, LIGO Livingston, MIT, PSU, UWM, and several European sites. Information about the LIGO Data Grid and how to get started can be found at:

- [How to get started on the LIGO Data Grid](#)

Once you have been informed that you have an account, you should be able to log into each of the clusters. [The following details should be moved onto the LDG web site]:

- (UWM) ssh [USERNAME@hydra.phys.uwm.edu](#)
 - Home directory is /home/USERNAME
 - Scratch space (mounted across cluster, run jobs from here) is /scratch, /scratch2, /scratch3, /scratch4
 - Local scratch (e.g. compile code here) is /usr1/USERNAME
 - Frame data is in automounted directories /nfsdata/nfsdata01 /nfsdata/nfsdata24
- (CIT) ssh [USERNAME@idas-grid.ligo.caltech.edu](#)
 - Home directory (mounted across cluster, run jobs from here) is /archive/home/pbrady
 - Local scratch is /usr1/USERNAME
 - Frame data is in automounted directories /archive/frames
- (LHO) ssh [USERNAME@idas-grid.ligo-wa.caltech.edu](#)
 - Home directory (mounted across cluster, run jobs from here) is /archive/home/pbrady
 - Local scratch is /usr1/USERNAME
 - Frame data is in automounted directories /archive/frames
- (LLO) ssh [USERNAME@idas-grid.ligo-la.caltech.edu](#)
 - Home directory (mounted across cluster, run jobs from here) is /archive/home/pbrady
 - Local scratch is /usr1/USERNAME
 - Frame data is in automounted directories /archive/frames

Condor is used as the batch scheduler on these clusters. You can find some introductory information at [Intro to LIGO Data Grid Client tools](#)

[그림 12] GRID VO를 통한 접근 가능한 LDG Computing자원 현황