

KISTI

GSDC TEM 분석 팜 자원 및 서비스 소개



유정록 (junglok.yu@kisti.re.kr)

한국과학기술정보연구원 (KISTI) 국가슈퍼컴퓨팅본부
대용량데이터허브센터 (GSDC)

2020. 11. 20.



목차

- GSDC (Global Science experimental Data-hub Center) 소개
- GSDC TEM 데이터 분석 환경 개요
- GSDC TEM 분석 팜 활용 가이드

연구 패러다임의 변화

데이터를 기반으로 한 연구 패러다임의 급부상

관측, 관찰을 통한 자연 현상 기술

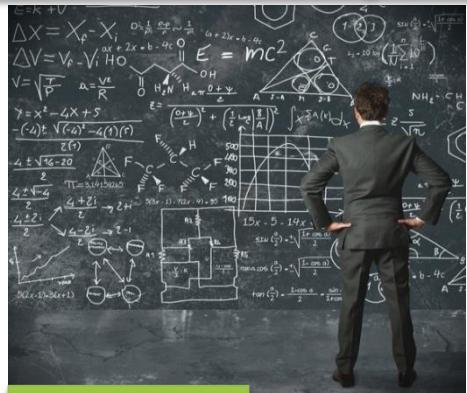
모델링과 일반화를 통한 자연 현상 기술

컴퓨터 시뮬레이션을 통해 자연 현상 기술

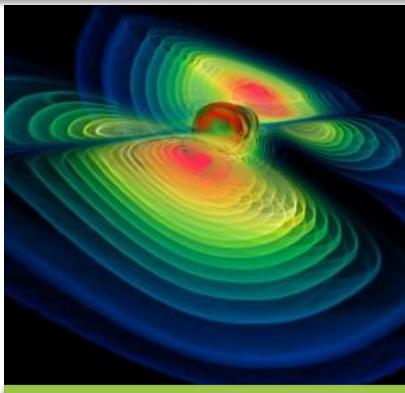
초대형 실험장비의 방대한 데이터 분석을 통한 자연 현상 기술 (데이터 집약형 연구)



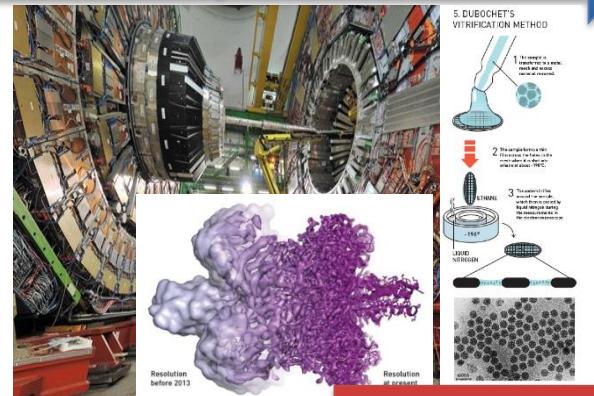
1세대: 관측



2세대: 이론



3세대: 시뮬레이션



4세대: 데이터

1610년, 갈릴레오는 최초의 망원경으로 목성, 금성, 달 관측 성공

1964년, 힉스는 물질에 질량을 부여하는 힉스 입자의 존재를 이론으로 설명

두 개의 블랙홀이 하나로 통합되면서 중력파가 생성되는 현상 컴퓨터 시뮬레이션

2013년, CERN의 CMS와 ATLAS 검출기를 통해 생성된 데이터를 분석하여 힉스입자 존재 증명 → 노벨상 수상

2017년 극저온전자현미경 (Cryo-EM) : 생체분자를 원자수준에서 3차원으로 관찰 → 노벨화학상 수상



출처: The Fourth Paradigm

GSDC 소개

데이터 집약형 기초과학 연구 지원 데이터센터

데이터 집약형 연구기반 확보를 통한 기초연구 진흥

GSDC Promoting Science

세계적 과학성과 창출을 위한 R&D 파트너

GSDC 역할

기초과학 선진화, 첨단화를 위한 기초과학분야
국가 통합데이터 센터 구축



서비스
·
개발

기술개발

분산저장구조체
고가용성스토리지
인프라통합관리

교육

연구그룹별 맞춤형 서비스

KiAF
(ALICE)
유전체
허브

LDG
(중력파)
기후/
온난화

CMS
T3
전자
현미경

방사광
가속기
뇌연구

전
략
분
야

글로벌 데이터 허브

WLCG Tier-1
서비스
(아시아허브)

국
제·국
내 협
력



기반
인프라

기초과학 데이터 통합 인프라 구축·운영 (10,000+ cores, 10+ PB, 첨단연구망)



데이터



GSDC 데이터 집약형 연구지원

국내 7개 연구 커뮤니티 공동활용
데이터집약형 다양한 연구도메인으로 확대

공동활용 34개 기관



국내의 고부가가치 대형연구시설장비 연계

- TEM : 초극저온전자현미경 CryoEM과 연계한 데이터 분석 팜 구축 (시범 : 2017~, 정식 : 2019. 09~)
- PAL : 4세대 방사광가속기 데이터분석 환경 구축 및 시범 서비스 (2018~)

4개 해외기관



해외 5,720여명 사용



이용건수 :
900만/연

SCI논문유발 :
80편 이상/연

국내 600여명 사용

GSDC TEM 서비스

- 배경

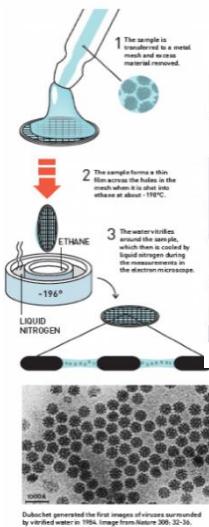
- 대형연구장비 (전자현미경, 가속기 등)와 연계한 데이터집약형 연구·실험 지원 확대 ('16)
- (과기정통부) 연구데이터 공유 · 활용 전략 시범 서비스 ('17-'18) : 대형연구장비분야 시범서비스
 - 데이터 수집, 데이터 분석 환경, 구조생물학분야 데이터 융합, 인력양성
- **GSDC TEM 서비스 공식 오픈 ('19. 09~)**

- 극저온투과전자현미경 (CryoEM) 데이터 저장 · 공유 · 분석 환경 구축 및 서비스

- (국내) KBSI CryoEM그룹, KAIST 의과학대학원/생명과학과/화학과, GIST, 서울대, 고려대, 연세대, 성균관대 등 11개 그룹

- GSDC 지원 내용

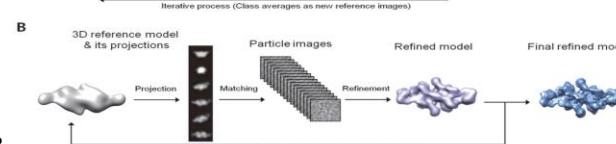
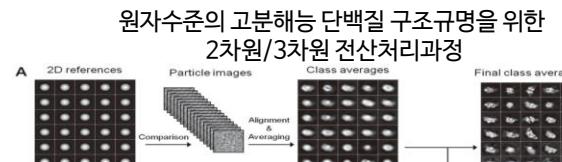
- 고속데이터전송망 (KREONET)을 통한 원시실험데이터의 고속 전송 · 공유 서비스
- 전자현미경 데이터 전용 분석 팜 구축 및 서비스



연간 100~200TB
전자현미경 원시데이터

KREONET

ScienceDMZ를 활용한
초고속데이터전송망
(KBSI ↔ GSDC)



전자현미경 데이터 분석 전용 팜



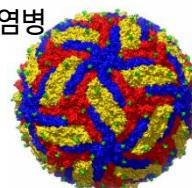
구조생물학연구



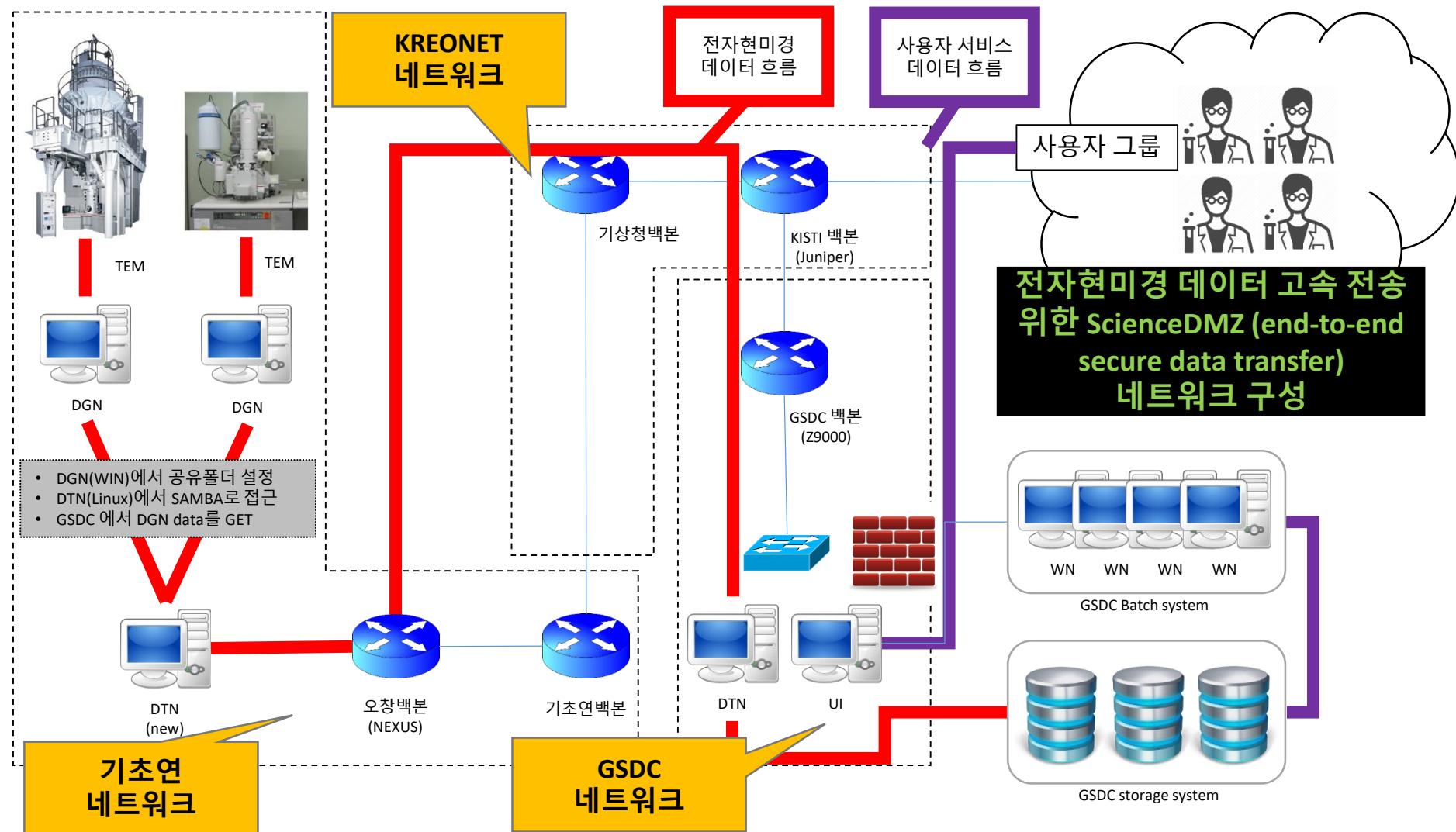
전염병



신약개발



KBSI-GSDC 고속 전송망 (전용 10Gbps) / 분석 팜 구축



GSDC TEM 분석 팜 활용 가이드

TEM 분석 팜 활용

<https://tem-docs.readthedocs.io>

0. TEM 사용자 계정 생성 요청
1. TEM 분석 팜 : 자원 현황 및 세부 구조
2. TEM 분석 팜 : 소프트웨어 및 분석 도구(System M/W, analysis tools)
3. TEM 분석 팜 로그인
4. Environment Modules
5. 배치 시스템(PBS), 배치 큐 (Queues)
6. Relion
 1. CPU support
 2. GPGPU support
7. cisTEM
 1. Run profiles
8. CryoSPARC (experimental)

0. TEM 사용자 계정 요청

- TEM-DOCS → Appendix → GSDC TEM Application Form 다운로드

GSDC User Account Application Form

Attention to: junglok.yu@kisti.re.kr

User's Information

Account Name (개정명)		VO (Research Group)	TEM
First Name (한글)		Last Name (한글)	
First Name (English)		Last Name (English)	
IP Addresses(IP주소)			
Organization (소속) (Full Name)	한글		
	English		
Department (부서) (Full Name)	한글		
	English		
Email		Phone (optional)	

Supervisor's Information

First Name (한글)		Last Name (한글)	
First Name (English)		Last Name (English)	
Organization		Department	
Email			

* Please, acquire Supervisor hand-written signature

User's Signature : _____ (date : . . .)

Supervisor's Signature : _____ (date : . . .)

개인정보 제공 및 이용 동의서

한국과학기술정보연구원 대용량데이터허브센터에서는 개인정보보호법 제15조 (개인정보의 수집·이용)에 의해 개인정보 수집 및 이용 동의를 받고 있습니다.

1. 개인정보 수집 및 이용 목적

A. GSDC TEM 서비스 계정 발급, 서비스 관련 지원 및 대응, 통계, 공지사항 전파 등을 위해 필요한 정보를 수집합니다.
B. 처리한 개인정보는 상기 목적 이외의 용도로는 사용되지 않으며 이용 목적이 변경될 시에는 사전 동의를 구할 예정입니다.

요청하신 계정 발급이 완료 되었습니다.

- - VO : tem
- account name : yrkim
- initial password :


- firewall registration:
141.223.129.13
- TEM users guide : <https://tem-docs.readthedocs.io>

@ tem-ui.sdfarm.kr에 접속(ssh port 4280)하셔서, 패스워드를 변경하시기 바랍니다.

@ 각 계정별로 20TB의 저장공간 퀴터가 제공되며, 이 저장공간은 raw data 및 분석 데이터 저장에 사용할 수 있습니다.

@ 자세한 TEM 분석 팜 접속 및 활용은 위 TEM 사용자 가이드를 참조하시면 됩니다.

1. TEM 분석 팜 : 자원 현황

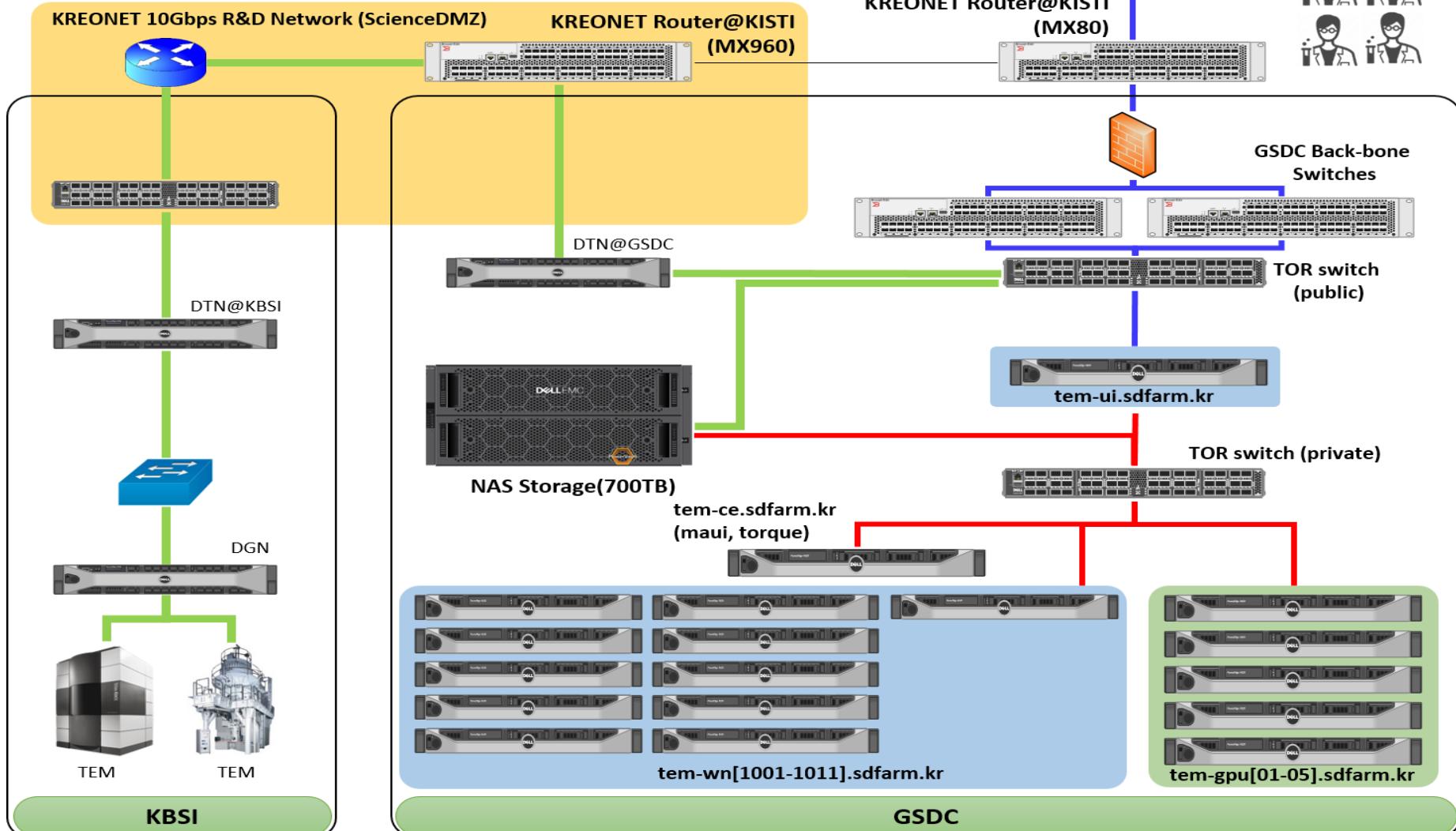
	‘16	‘17	‘18	‘19	’20(현재)
+CPU (Physical Core)	308	364	364	364	364 (+100 예정)
GPU (EA)	-	6 (P100)	6(P100)	6(P100)	10 (P100+P40)
† Storage(TB)	10	500	500	700 (500)	800 (500)
사용자	4	4	10	15	26
작업처리건수	78	196	368,659	764,747	402,541

† UI/CE 서버를 포함한 물리코어. GPU를 호스팅하기 위한 서버 물리코어는 제외됨.

† 괄호안은 KBSI 전자현미경 데이터 백업/아카이빙 용도임. 나머지는 사용자 데이터 분석용.

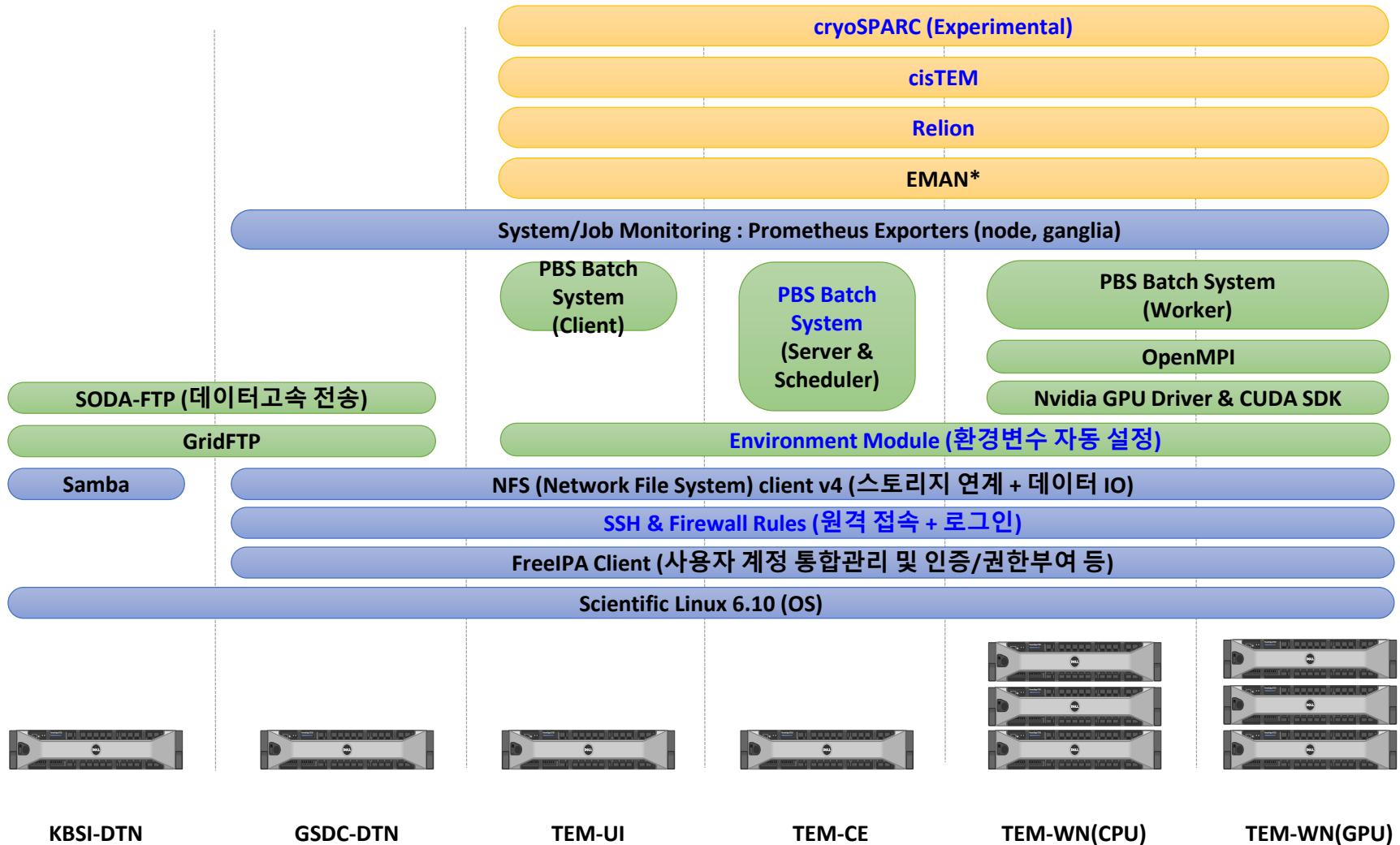
1. TEM 분석 팜 세부 구성도 (1/2)

— Data Flow
— User Access
— Private Area



* DGN : Data Generation Node, DTN : Data Transfer Node

1. TEM 분석 팜 세부 구성도 (2/2)



2. TEM 분석 팜 : HW 자원

- HW Specification (as of 2020. 11.)

Category	Name	Specification	Resources size
Login	tem-ui.sdfarm.kr	<ul style="list-style-type: none">• CPU : Intel(R) Xeon(R) CPU E5-2697v3 @ 2.60GHz 14Core * 2 CPUs• RAM : DDR4 8GB * 24 (192GB)• HDD : 12G SAS HDD 1.2TB * 2EA (RAID-1)	28 cores
Computing (master)	tem-ce.sdfarm.kr	<ul style="list-style-type: none">• CPU : Intel(R) Xeon(R) CPU E5-2697v3 @ 2.60GHz 14Core * 2 CPUs• RAM : DDR4 8GB * 24 (192GB)• HDD : 12G SAS HDD 1.2TB * 2EA (RAID-1)	28 cores
Computing (workers)	tem-wn[1001-1011].sdfarm.kr	<ul style="list-style-type: none">• CPU : Intel(R) Xeon(R) CPU E5-2697v3 @ 2.60GHz 14Core * 2 CPUs• RAM : DDR4 8GB * 24 (192GB)• HDD : 12G SAS HDD 1.2TB * 2EA (RAID-1)	308 cores
	tem-gpu[01-05].sdfarm.kr	<ul style="list-style-type: none">• CPU : Intel® Xeon® CPU E5-2690v4 @ 2.60GHz 14Core * 2 CPUs• RAM : DDR4 16GB * 24 (384GB)• SSD : 6G SATA SSD 800GB * 2EA (RAID-1)• GPU : NVIDIA P100 * 2ea (each tem-gpu[01-03] has 2 P100 GPU devices)• GPU : NVIDIA P40 * 2ea (each tem-gpu[04-05] has 2 P40 GPU devices))	<ul style="list-style-type: none">• 140 cores• 10 GPGPUs
Storage	Dell EMC Isilon NAS	Network attached storage 800 TB	
Total		504 CPU cores, 10 GPGPUs, 800TB Storage	

2. TEM 분석 팜 : System, M/W

- Cluster management S/W

Category	Name	Description	Version (module path)
OS	Scientific Linux	Operating system	6.x
System M/W	Environment module	<ul style="list-style-type: none">Module environmenthttps://modules.readthedocs.io/en/latest	v3.2.10
	OpenPBS(torque)	<ul style="list-style-type: none">Cluster resources managementhttp://www.adaptivecomputing.com/products/torque	v6.1.2
	OpenMPI	<ul style="list-style-type: none">Messaging Pass Interface(MPI)Reference implementation for MPI standardhttps://www.open-mpi.org	v1.8.8 (mpi/gcc/openmpi/1.8.8)
	cuda	<ul style="list-style-type: none">Compute Unified Device Architecture(CUDA)NVIDIA CUDA Runtime & Toolkithttps://developer.nvidia.com/cuda-toolkit	9.1 (cuda/9.1)
	python	<ul style="list-style-type: none">Python runtime	v2.6.6

2. TEM 분석 팜 : Data Analysis Tools

- Data analysis tools

Category	Name	Description	Version (module path)
Tools	Relion	<p>A stand-alone computer program that employs an empirical Bayesian approach to refinement of (multiple) 3D reconstructions or 2D class averages in electron cryo-microscopy (cryo-EM).</p> <ul style="list-style-type: none">• https://www3.mrc-lmb.cam.ac.uk/relion/index.php	v3.0.7 (apps/gcc/4.4.7/relion/cpu/3.0.7) (apps/gcc/4.4.7/relion/gpu/3.0.7) (apps/gcc/4.4.7/relion/gpu/3.0.7p)
	cisTEM	<p>User-friendly software to process cryo-EM images of macromolecular complexes and obtain high-resolution 3D reconstructions.</p> <ul style="list-style-type: none">• https://cistem.org	v1.0.0 (apps/gcc/4.4.7/cistem/1.0.0)
	CryoSPARC	<p>CryoSPARC is the state-of-the-art platform used globally for obtaining 3D structural information from single particle cryo-EM data.</p> <ul style="list-style-type: none">• https://cryosparc.com	v2.14.2

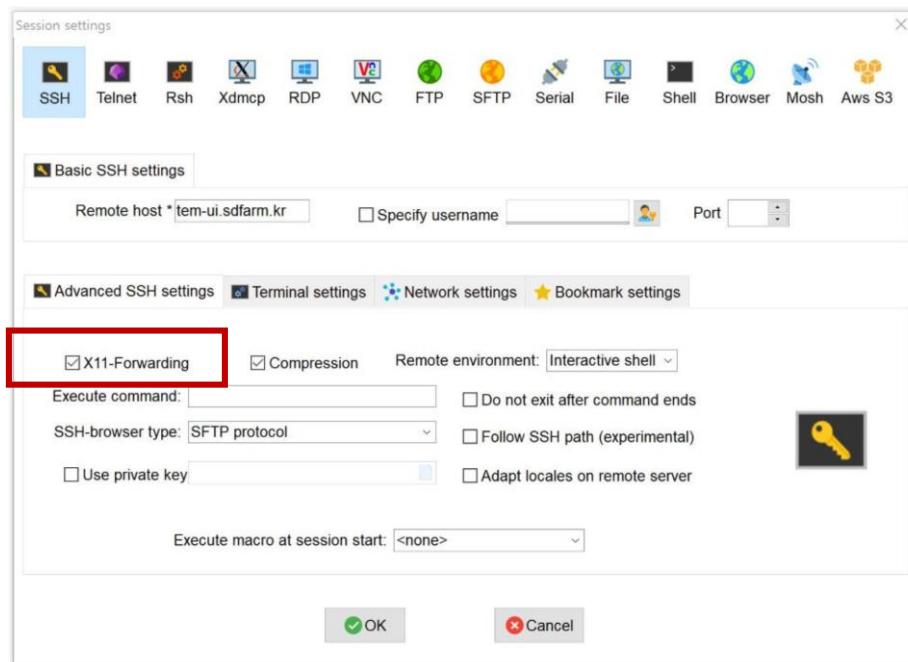
3. TEM 분석 팜 로그인

- 리눅스/맥 사용자

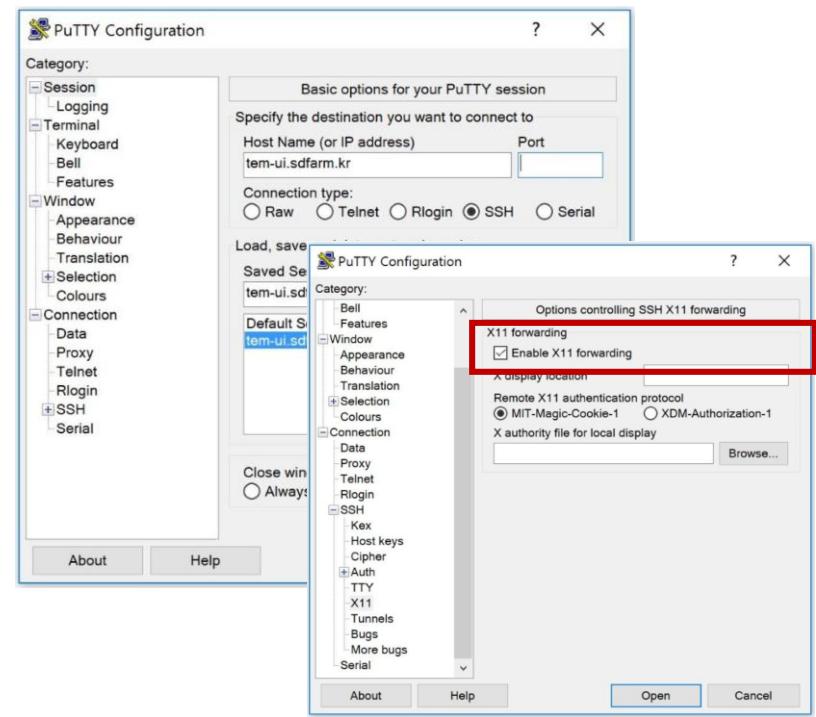
```
$> ssh -Y -o Port=<port> <userID>@tem-ui.sdfarm.kr
```

-Y (or -X) options : enable trusted X11 (or untrusted X11) forwarding

- 윈도우 사용자



<moba-xterm 클라이언트>



<Putty 클라이언트>

3. TEM 분석 팜 로그인 : 쉘

- UI 서버 (로그인 서버, tem-ui.sdfarm.kr) 로그인 성공 후 터미널(쉘) 화면

```
/[=][=][=][=]\/[=] [=][=][=][=][V=][=][=]/[=][=][=][=]\
* Official GSDC TEM users guide : https://tem-docs.readthedocs.io
=====
* Hostname.....: tem-ui.sdfarm.kr
* OS Release....: Scientific Linux release 6.10 (Carbon)
* System uptime.: 73 days 3 hours 20 minutes 31 seconds
* Users.....: Currently 16 user(s) logged on
* Processes.....: 1383 running
* CPU usage.....: 1.76, 2.12, 1.49 (1, 5, 15 min)
* Memory (used/total)...: 30965 MB / 193583 MB
* Swap in use.....: 502 MB
-----
* TEM disk (used/total)..: 407 TB / 800 TB (51%)
* Current user.....: tem
* Home directory.....: /tem/home/tem
* Disk Quota limit.....: 20480 GB
* Disk usage.....: 920 GB (4.4937 %)
* # of Files.....: 1696550
=====

[[tem@tem-ui ~]$ pwd
/tem/home/tem
[tem@tem-ui ~]$ ]
```

3. TEM 분석 팜 로그인 : 디렉토리 구조

- /tem 디렉토리 구조

- \$) df -h (파일시스템 디스크 사용공간 리포팅)

```
[tem@tem-ui ~]$ df -h
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/rootvg-root
                     7.9G  6.3G  1.2G  85% /
tmpfs                 95G    0  95G   0% /dev/shm
/dev/sda1              504M 124M 355M  26% /boot
/dev/mapper/rootvg-home
                     2.0G  3.0M  1.9G   1% /home
/dev/mapper/rootvg-opt
                     2.0G 131M  1.7G   8% /opt
/dev/mapper/rootvg-tmp
                     2.0G  11M  1.8G   1% /tmp
/dev/mapper/rootvg/usr
                     20G   14G  4.8G  75% /usr
/dev/mapper/rootvg-var
                     9.8G  5.1G  4.3G  54% /var
/dev/mapper/rootvg-audit
                     976M  31M  894M   4% /var/log/audit
/dev/mapper/rootvg-vartmp
                     2.0G  3.1M  1.9G   1% /var/tmp
pool0.gsn2.sdfarm.kr:/ifs/service/tem
                     800T 396T 405T  50% /tem
pool0.gsn.sdfarm.kr:/ifs/gsdc/home/junglok.yu
                     10T  8.3T  1.8T  83% /share/junglok.yu
pool0.gsn.sdfarm.kr:/ifs/gsdc/home/hjhan
                     10T  8.3T  1.8T  83% /share/hjhan
[tem@tem-ui ~]$
```

```
[tem@tem-ui tem]$ tree -L 2
.
└── el7
    ├── Gctf_v1.18_b2
    ├── Modules
    ├── MotionCor2_v1.3.1
    ├── ResMap-1.1.4
    ├── ctffind-4.1.14
    ├── openmpi-1.8.8
    ├── qsub-relion-3.0.7-cpu.bash
    ├── qsub-relion-cpu.bash
    ├── relion-3.0.7
    ├── relion-3.1.0
    ├── summovie_1.0.2
    └── unblur_1.0.2
    └── home
        ├── barca2625
        ├── blee
        ├── brain
        ├── dtm_test
        ├── gistsmsjin
        ├── gistxray
        ├── hksong1
        ├── hsjeong
        ├── ihee
        ├── jhyun002
        ├── junsun873
        ├── kaistmacro
        ├── kanglab
        ├── kbscryo
        ├── linux
        ├── linux_grid
        ├── linux_test
        ├── msbl
        ├── sangholee
        ├── selcia20
        ├── shroh
        ├── snustrucbio
        ├── songj
        ├── songj2
        └── sonai3
            └── tem
                ├── tem_gsac
                ├── tem_test
                ├── test01
                ├── test02
                ├── waveware
                ├── waveware1
                ├── wtleys
                └── yrkim
    └── repo
        ├── DataCollection
        └── cistem
```

```
[tem@tem-ui ~]$ tree -L 1
.
├── Modules
├── Sources
└── Applications
    └── _Backups
    └── _Benchmarks
└── SystemLibs
    ├── anaconda2
    ├── ansible-project
    ├── proj.tar.gz
    ├── qsub-relion3-cpu.bash
    └── qsub-relion3-gpu.bash
    └── tmp
9 directories, 3 files
```

4. Environment Modules (1/3)

- Environment Module
 - 응용프로그램(또는 관련된 라이브러리)에 대한 환경변수 (path, link 정보 등)을 구조적으로 관리하고 자동 설정하는 도구
 - 쉘 환경 설정에 대한 진입장벽 없이 리눅스 사용자들이 쉘 환경을 편하게 사용할 수 있음
 - 응용 프로그램의 path, 관련 라이브러리 path 및 기타 환경변수 등
 - 모듈 : module file로 관리되며, 응용프로그램(또는 라이브러리) 실행을 위한 환경 변수 및 의존성을 자동 설정함

4. Environment Modules (2/3)

- 사용 가능한 모듈 리스트업 : \$) module avail (또는 \$) module av)

```
[tem@tem-ui ~]$ module avail
-----
3.2.10
-----
/tem/home/tem/Modules/Modules/versions -----
anaconda/2.7
apps/gcc/4.4.7/cistem/1.0.0
apps/gcc/4.4.7/relion/cpu/3.0.7
[tem@tem-ui ~]$
```

모듈 상세 정보

- Syntax : \$) module show [module path]
- 예 : \$) module show apps/gcc/4.4.7/relion/gpu/3.0.7

```
[tem@tem-ui ~]$ module show apps/gcc/4.4.7/relion/gpu/3.0.7
-----
/tem/home/tem/Modules/Modules/default/modulefiles/apps/gcc/4.4.7/relion/gpu/3.0.7:

module-whatis      Setsups `relion-3.0.7' environment variables
module           load mpi/gcc/openmpi/1.8.8
module           load cuda/9.1
setenv            relion_version 3.0.7
prepend-path       PATH /tem/home/tem/_Applications/relion-3.0.7/gpu/bin
prepend-path       LD_LIBRARY_PATH /tem/home/tem/_Applications/relion-3.0.7/gpu/lib
setenv             LANG en_US.UTF-8
setenv             RELION_QUEUE_USE yes
setenv             RELION_QUEUE_NAME own_queue_name
setenv             RELION_QSUB_COMMAND qsub
setenv             RELION_QSUB_TEMPLATE /tem/home/tem/_Applications/relion-3.0.7/gpu/bin/qsub-relion3-gpu.bash
setenv             RELION_QSUB_EXTRA_COUNT 3
setenv             RELION_QSUB_EXTRA1 Number of Nodes
setenv             RELION_QSUB_EXTRA2 Number of processes per each node
setenv             RELION_QSUB_EXTRA3 Number of GPUs per node
setenv             RELION_QSUB_EXTRA1_DEFAULT 1
setenv             RELION_QSUB_EXTRA2_DEFAULT 3
setenv             RELION_QSUB_EXTRA3_DEFAULT 2
setenv             RELION_CTFIND_EXECUTABLE /tem/home/tem/_Applications/ctffind-4.1.13/bin/ctffind
setenv             RELION_GCTF_EXECUTABLE /tem/home/tem/_Applications/Gctf_v1.18_b2/bin/Gctf_v1.18_b2_sm60_cu9.1
setenv             RELION_RESMAP_EXECUTABLE /tem/home/tem/_Applications/ResMap-1.1.4/ResMap-1.1.4-linux64
setenv             RELION_MOTIONCOR2_EXECUTABLE /tem/home/tem/_Applications/MotionCor2/MotionCor2_Cuda9.1_v1.0.5
setenv             RELION_UNBLUR_EXECUTABLE /tem/home/tem/_Applications/unblur_1.0.2/bin/unblur_openmp_7_17_15.exe
setenv             RELION_SUMMOVIE_EXECUTABLE /tem/home/tem/_Applications/summovie_1.0.2/bin/sum_movie_openmp_7_17_15.exe
conflict          apps/gcc/4.4.7/relion
-----
[tem@tem-ui ~]$
```

4. Environment Modules (3/3)

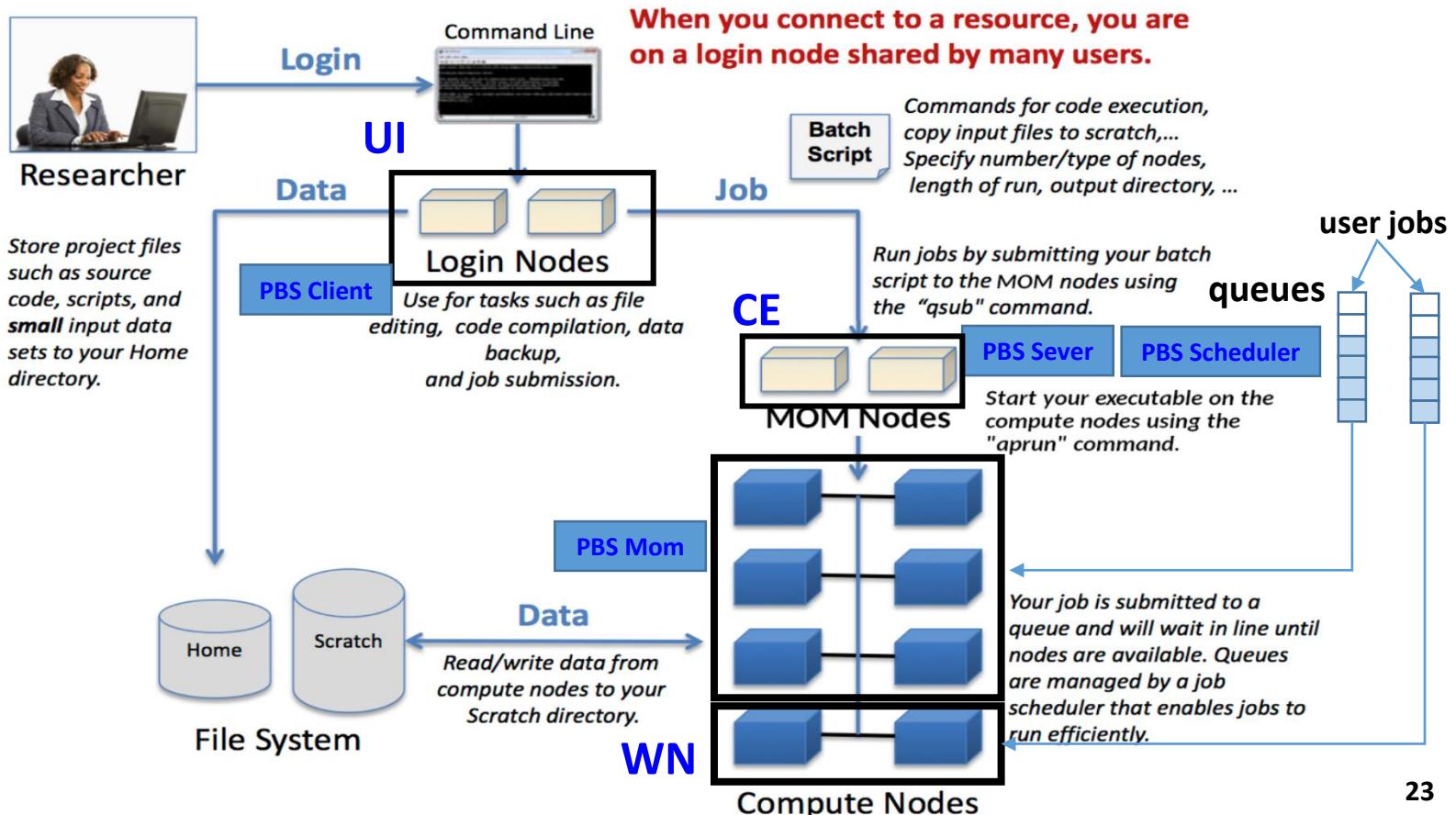
- 모듈 로딩하기
 - Syntax : **\$> module load [module path]**
 - 예 : **\$> module load apps/gcc/4.4.7/relion/gpu/3.0.7**
- 현재 로딩된 모듈들 보기
 - **\$> module list**

```
[tem@tem-ui ~]$ module load apps/gcc/4.4.7/relion/gpu/3.0.7
[tem@tem-ui ~]$ module list
Currently Loaded Modulefiles:
  1) mpi/gcc/openmpi/1.8.8           2) cuda/9.1                  3) apps/gcc/4.4.7/relion/gpu/3.0.7
[tem@tem-ui ~]$
[tem@tem-ui ~]$
[tem@tem-ui ~]$
[tem@tem-ui ~]$ which relion
~/Applications/relion-3.0.7/gpu/bin/relion
[tem@tem-ui ~]$ which mpirun
~/SystemLibs/openmpi-1.8.8/bin/mpirun
[tem@tem-ui ~]$ which nvcc
/usr/local/cuda-9.1/bin/nvcc
[tem@tem-ui ~]$ █
```

- 모듈 언로딩하기
 - Syntax : **\$> module unload [module path]**
- 모든 모듈 언로딩하기
 - Syntax : **\$> module purge**

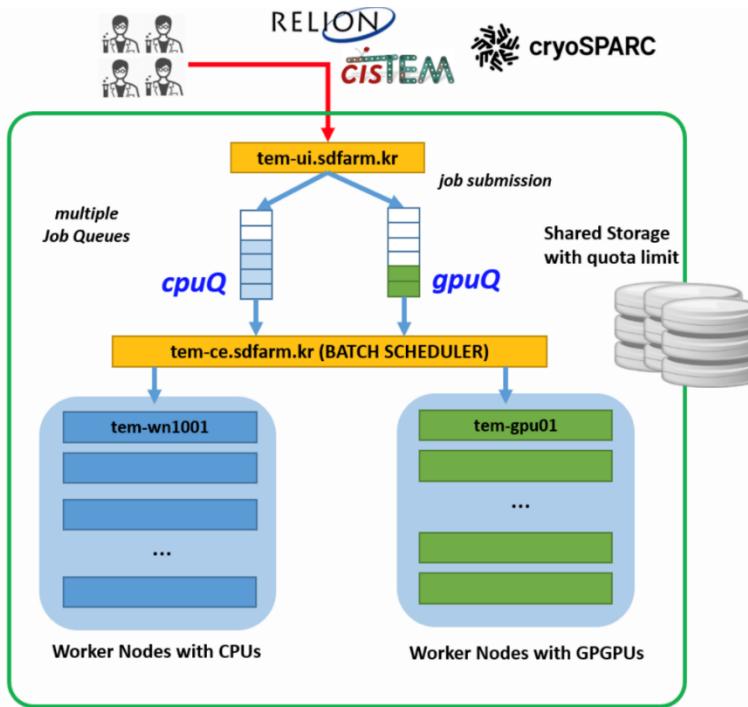
5. 배치 시스템

- 배치 시스템
 - 계산자원 (CPU, GPU, 메모리, 점유시간 등) 관리
 - 사용자 작업 명세서 (batch script)를 바탕으로 제출된 작업별로 자원을 할당(스케줄링)하는 시스템
- PBS, SLURM, HTCondor, etc.



5. 배치 시스템 : 큐 (Queue)

큐 : 계산자원들을 임의의 특성에 따라 파티셔닝 (또는 분할) 하여 매핑시켜 놓은 가상공간



Category	Queue Name	Assigned Computing Resources	Remarks
Shared	cpuQ	<ul style="list-style-type: none">tem-wn[1001-1011].sdfarm.kr (28 cores and 192GB memory per node)	<ul style="list-style-type: none">308 Physical CPU cores
	gpuQ	<ul style="list-style-type: none">tem-gpu[01-03].sdfarm.kr (28 cores, 2 P100 GPGPUs and 384GB memory)tem-gpu04.sdfarm.kr (24 cores, 2 P40 GPGPUs and 128GB memory)tem-gpu05.sdfarm.kr (36 cores, 2 P40 GPGPUs and 256GB memory)	<ul style="list-style-type: none">144 Physical CPU cores10 GPGPUsP100 has 16GB device memoryP40 has 24GB device memory

5. 배치 시스템 : 작업 명세서 (job script)

- PBS 작업 명세서 (job script)
 - 할당 받고자 하는 계산 자원에 대한 요구사항 기술 (PBS Directives로 표현, prefix '#PBS')
 - 실행하고자 하는 작업에 대한 기술

- Node and PPN (Processor per node)

- Wall clock time

- Memory

- Job name

- #PBS -N jobName

- Queue name

- #PBS -q cpuQ

- Log files

- #PBS -j oe

To request a single core on the farm:
`#PBS -l nodes=1:ppn=1`

To request one whole node on the farm:
`#PBS -l nodes=1:ppn=28`

To request 4 whole nodes on the farm:
`#PBS -l nodes=4:ppn=28`

To request 3 whole nodes with 2 GPUs on the farm:
`#PBS -l nodes=3:ppn=28:gpus=2`

To request 1 node with use of 6 cores and 1 GPU:
`#PBS -l nodes=1:ppn=6:gpus=1`

To request 20 hours of wall clock time:
`#PBS -l walltime=20:00:00`

To request 4GB memory:
`#PBS -l mem=4GB`
or
`#PBS -l mem=4000MB`

To request 24GB memory:
`#PBS -l mem=24000MB`

5. 배치 시스템 : 작업 명세서 (job script)

- PBS 작업 명세서 (job script)
 - 할당 받고자 하는 계산자원에 대한 요구사항 기술 (Directives로 표현)
 - 실행하고자 하는 작업에 대한 기술
- 예시

<serial job>

```
#PBS -N jobName
#PBS -l walltime=40:00:00
#PBS -l nodes=1:ppn=1
#PBS -q batch

cd $PBS_O_WORKDIR
/usr/bin/time ./mysci > mysci.hist
```

<parallel job>

```
#PBS -N jobName
#PBS -l walltime=10:00:00
#PBS -l nodes=4:ppn=4
#PBS -q batch

module load mpi/gcc/openmpi/1.8.8
cd $PBS_O_WORKDIR
mpirun -machinefile $PBS_NODEFILE ./a.out
```

5. 배치 시스템 : 작업 관리

- 작업 제출 (qsub)

myscript.job : the script file name of a PBS batch job

```
$> qsub myscript.job
```

In response to this command you'll see a line with your job ID:

```
123456.tem-ce.sdfarm.kr
```

- 작업 상태 모니터링 (qstat)

By itself, qstat lists all jobs in the system in standard or alternate format:

```
$> qstat
```

or

```
$> qstat -a
```

To list all the jobs belonging to a particular user:

```
$> qstat -u tem_user
```

To list the status of a particular job, in standard or alternate format:

```
$> qstat 123456
```

```
$> qstat -a 123456
```

To get all the details about a particular job (full status):

```
$> qstat -f 123456
```

To list the status of all the queues

```
$> qstat -Qf
```

- 작업 취소 (qdel)

```
$> qdel 123456
```

6. Relion

- RELION (for REregularised LIkelihood OptimisatioN)
 - stand-alone computer program that employs an empirical Bayesian approach to refinement of (multiple) 3D reconstructions or 2D class averages in electron cryo-microscopy (cryo-EM)
 - (from Relion official site https://www3.mrc-lmb.cam.ac.uk/relion/index.php?title=Main_Page)
- Module 경로 및 상세 정보 확인 → Module 로딩 → 분석 프로그램/도구 실행
- 활용 가능한 모듈 확인
 - \$> module avail

```
[tem@tem-ui ~]$ module avail
----- /tem/home/tem/Modules/Modules/versions -----
3.2.10

----- /tem/home/tem/Modules/Modules/default/modulefiles -----
anaconda/2.7
apps/gcc/4.4.7/cistem/1.0.0
apps/gcc/4.4.7/relion/cpu/3.0.7
apps/gcc/4.4.7/relion/gpu/3.0.7p
cuda/9.1
modules
mpi/gcc/openmpi/1.8.8
[tem@tem-ui ~]$
```

6. Relion

- Relion 프로그램 상세 정보
 - \$> module show apps/gcc/4.4.7/relion/cpu/3.0.7

```
$> module show apps/gcc/4.4.7/relion/cpu/3.0.7

-----
/tcm/home/tem/Modules/Modules/default/modulefiles/apps/gcc/4.4.7/relion/cpu/3.0.7:

module-whatis      Setsups `relion-3.0.7' environment variables
module      load mpi/gcc/openmpi/1.8.8
setenv      relion_version 3.0.7
prepend-path      PATH /tcm/home/tem/_Applications/relion-3.0.7/cpu/bin
prepend-path      LD_LIBRARY_PATH /tcm/home/tem/_Applications/relion-3.0.7/cpu/lib
setenv      LANG en_US.UTF-8
setenv      RELION_QUEUE_USE yes
setenv      RELION_QUEUE_NAME own_queue_name
setenv      RELION_QSUB_COMMAND qsub
setenv      RELION_QSUB_TEMPLATE /tcm/home/tem/_Applications/relion-3.0.7/cpu/bin/qsub-relion3-cpu.bash
setenv      RELION_QSUB_EXTRA_COUNT 2
setenv      RELION_QSUB_EXTRA1 Number of Nodes
setenv      RELION_QSUB_EXTRA2 Number of processes per each node
setenv      RELION_QSUB_EXTRA1_DEFAULT 2
setenv      RELION_QSUB_EXTRA2_DEFAULT 16
setenv      RELION_CTFFIND_EXECUTABLE /tcm/home/tem/_Applications/ctffind-4.1.13/bin/ctffind
setenv      RELION_GCTF_EXECUTABLE /tcm/home/tem/_Applications/Gctf_v1.18_b2/bin/Gctf_v1.18_b2_sm60_cu9.1
setenv      RELION_RESMAP_EXECUTABLE /tcm/home/tem/_Applications/ResMap-1.1.4/ResMap-1.1.4-linux64
setenv      RELION_MOTIONCOR2_EXECUTABLE /tcm/home/tem/_Applications/MotionCor2/MotionCor2_Cuda9.1_v1.0.5
setenv      RELION_UNBLUR_EXECUTABLE /tcm/home/tem/_Applications/unblur_1.0.2/bin/unblur_openmp_7_17_15.exe
setenv      RELION_SUMMOVIE_EXECUTABLE /tcm/home/tem/_Applications/summovie_1.0.2/bin/sum_movie_openmp_7_17_15.exe
conflict      apps/gcc/4.4.7/relion
```

6. Relion

- Relion 프로그램 상세 정보

- \$> module show apps/gcc/4.4.7/relion/gpu/3.0.7

```
$> module show apps/gcc/4.4.7/relion/gpu/3.0.7
```

```
-----  
/tem/home/tem/Modules/Modules/default/modulefiles/apps/gcc/4.4.7/relion/gpu/3.0.7:  
  
module-whatis      Setsups `relion-3.0.7' environment variables  
module      load mpi/gcc/openmpi/1.8.8  
module      load cuda/9.1  
setenv      relion version 3.0.7  
prepend-path      PATH /tem/home/tem/_Applications/relion-3.0.7/gpu/bin  
prepend-path      LD_LIBRARY_PATH /tem/home/tem/_Applications/relion-3.0.7/gpu/lib  
setenv      LANG en_US.UTF-8  
setenv      RELION_QUEUE_NAME tem  
setenv      RELION_QSUB_COMMAND qsub  
setenv      RELION_QSUB_TEMPLATE /tem/home/tem/_Applications/relion-3.0.7/gpu/bin/qsub-relion3-gpu.bash  
setenv      RELION_QSUB_EXTRA_COUNT 3  
setenv      RELION_QSUB_EXTRA1 Number of Nodes  
setenv      RELION_QSUB_EXTRA2 Number of processes per each node  
setenv      RELION_QSUB_EXTRA3 Number of GPUs per node  
setenv      RELION_QSUB_EXTRA1_DEFAULT 1  
setenv      RELION_QSUB_EXTRA2_DEFAULT 3  
setenv      RELION_QSUB_EXTRA3_DEFAULT 2  
setenv      RELION_CTFIND_EXECUTABLE /tem/home/tem/_Applications/ctffind-4.1.13/bin/ctffind  
setenv      RELION_GCTF_EXECUTABLE /tem/home/tem/_Applications/Gctf_v1.18_b2/bin/Gctf_v1.18_b2_sm60_cu9.1  
setenv      RELION_RESMAP_EXECUTABLE /tem/home/tem/_Applications/ResMap-1.1.4/ResMap-1.1.4-linux64  
setenv      RELION_MOTIONCOR2_EXECUTABLE /tem/home/tem/_Applications/MotionCor2/MotionCor2_Cuda9.1_v1.0.5  
setenv      RELION_UNBLUR_EXECUTABLE /tem/home/tem/_Applications/unblur_1.0.2/bin/unblur_openmp_7_17_15.exe  
setenv      RELION_SUMMOVIE_EXECUTABLE /tem/home/tem/_Applications/summovie_1.0.2/bin/sum_movie_openmp_7_17_15.exe  
conflict      apps/gcc/4.4.7/relion  
-----
```

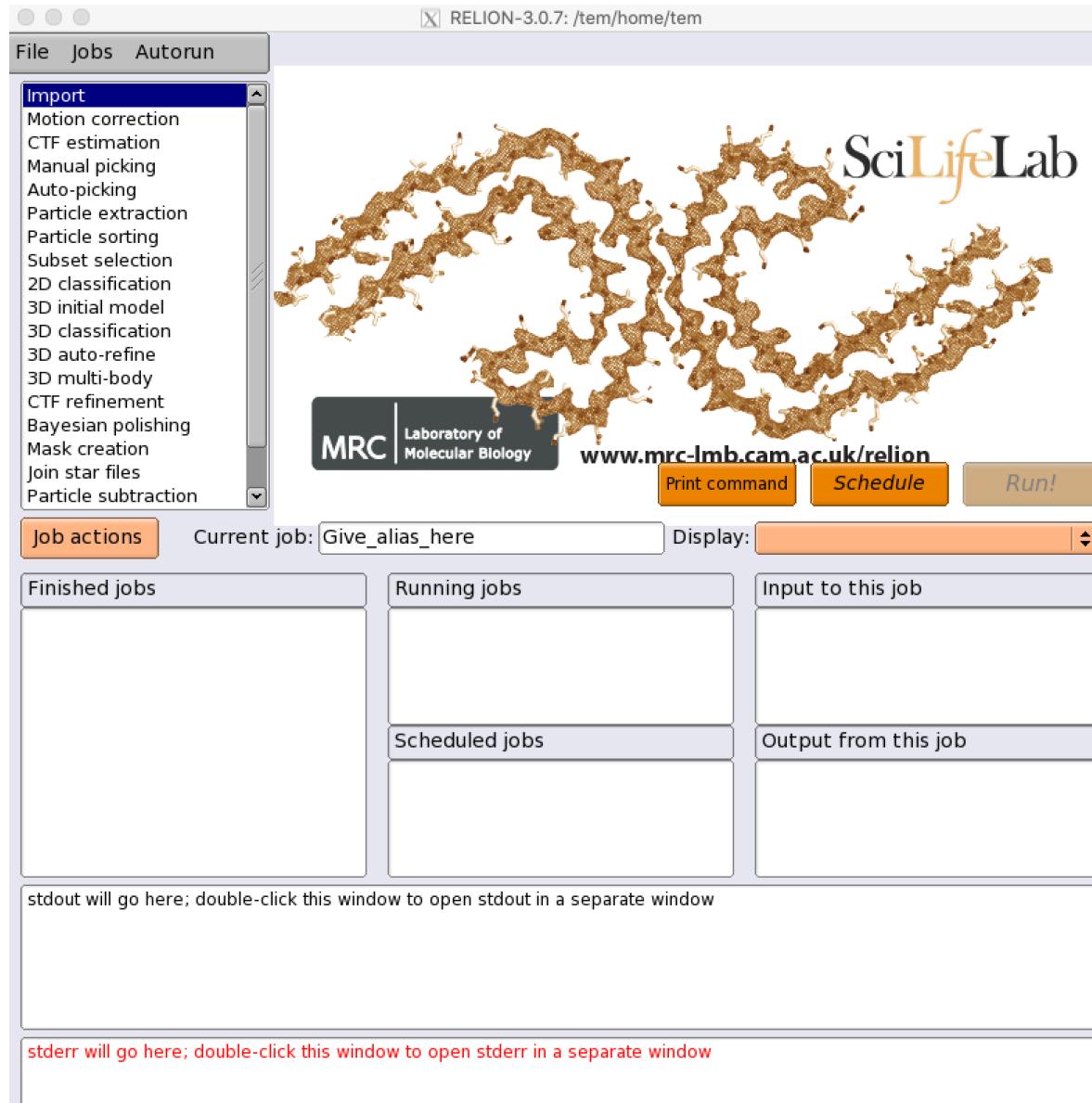
6. Relion

- Relion 모듈 로딩
 - (CPU 버전) \$> module load apps/gcc/4.4.7/relion/cpu/3.0.7
 - (GPU 버전) \$> module load apps/gcc/4.4.7/relion/gpu/3.0.7
- 로딩된 모듈 확인
 - \$> module list

```
[tem@tem-ui ~]$ module load apps/gcc/4.4.7/relion/gpu/3.0.7
[tem@tem-ui ~]$ module list
Currently Loaded Modulefiles:
    1) mpi/gcc/openmpi/1.8.8          3) apps/gcc/4.4.7/relion/gpu/3.0.7
    2) cuda/9.1
[tem@tem-ui ~]$ which relion
~/_Applications/relion-3.0.7/gpu/bin/relion
[tem@tem-ui ~]$ █
```

- Relion GUI 프로그램 시작
 - \$> relion

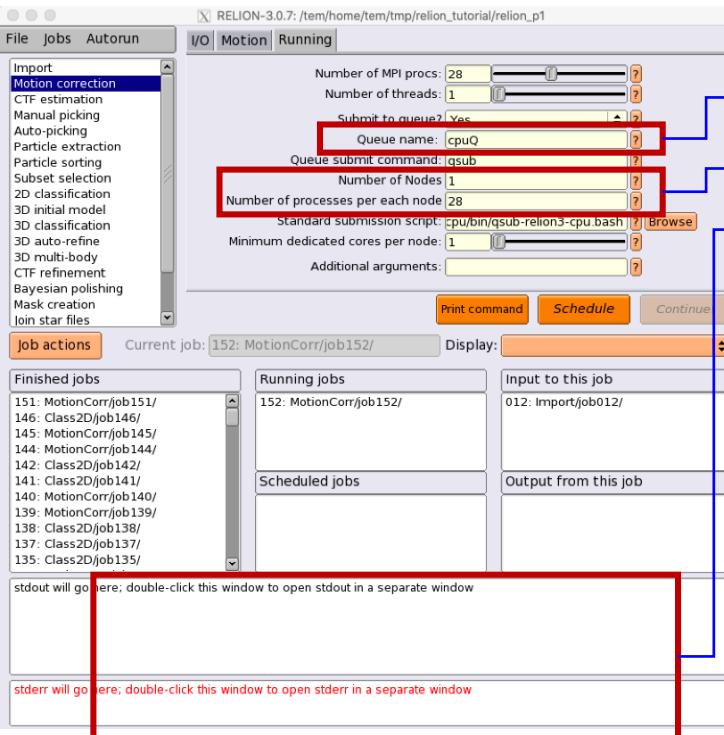
6. Relion



6. Relion : CPU 분석 작업 제출

/tem/home/tem/_Applications/relion-3.0.7/cpu/bin/qsub-relion3-cpu.bash

- CPU 작업 템플릿



```
#!/bin/bash

### Inherit all current environment variables
#PBS -V

### Job name
#PBS -N XXXnameXXX

### Queue name
#PBS -q XXXqueueXXX

### CPU cluster use : Specify the number of nodes (XXXextra1XXX) and the number of processes per each
#PBS -l nodes=XXXextra1XXX:ppn=XXXextra2XXX:XXXqueueXXX

#PBS -o ${PBS_JOBNAME}/run.out
#PBS -e ${PBS_JOBNAME}/run.err

#####
### Print Environment Variables
#####

echo -----
echo -n 'Job is running on node ' ; cat $PBS_NODEFILE
echo -----

echo PBS: qsub is running on $PBS_O_HOST
echo PBS: originating queue is $PBS_O_QUEUE
echo PBS: executing queue is $PBS_QUEUE
echo PBS: working directory is $PBS_O_WORKDIR
echo PBS: execution mode is $PBS_ENVIRONMENT
echo PBS: job identifier is $PBS_JOBID
echo PBS: job name is $PBS_JOBNAME
echo PBS: node file is $PBS_NODEFILE
echo PBS: current home directory is $PBS_O_HOME
echo PBS: PATH = $PBS_O_PATH
echo -----


#####
# Switch to the working directory;
cd ${PBS_O_WORKDIR}/${PBS_JOBNAME}
touch run.out
touch run.err
cd ${PBS_O_WORKDIR}
#####

### Run:
module load apps/gcc/4.4.7/relion/cpu/3.0.7
mpirun --prefix /tem/home/tem/_SystemLibs/openmpi-1.8.8 -machinefile $PBS_NODEFILE XXXcommandXXX

echo "Done!"
```

6. Relion : GPU 분석 작업 제출

RELION-3.0.7: /tem/home/tem/tmp/relion_tutorial/relion_p1

I/O Motion Running

Import

- Motion correction
- CTF estimation
- Manual picking
- Auto-picking
- Particle extraction
- Particle sorting
- Subset selection
- 2D classification
- 3D initial model
- 3D classification
- 3D auto-refine
- 3D multi-body
- CTF refinement
- Bayesian polishing
- Mask creation
- Join star files
- Particle subtraction

Job actions Current job: Give_alias_here Display:

Finished jobs

- 152: MotionCorr/job152/
- 151: MotionCorr/job151/
- 146: Class2D/job146/
- 145: MotionCorr/job145/
- 144: MotionCorr/job144/
- 142: Class2D/job142/
- 141: Class2D/job141/
- 140: MotionCorr/job140/
- 139: MotionCorr/job139/
- 138: Class2D/job138/
- 137: Class2D/job137/

stdout will go here; double-click this window to open stdout in a separate window

stderr will go here; double-click this window to open stderr in a separate window

RELION-3.0.7: /tem/home/tem/tmp/relion_tutorial/relion_p1

I/O Motion Running

Import

- Motion correction
- CTF estimation
- Manual picking
- Auto-picking
- Particle extraction
- Particle sorting
- Subset selection
- 2D classification
- 3D initial model
- 3D classification
- 3D auto-refine
- 3D multi-body
- CTF refinement
- Bayesian polishing
- Mask creation
- Join star files
- Particle subtraction

Job actions Current job: 153: MotionCorr/job153/ Display:

Finished jobs

- 152: MotionCorr/job152/
- 151: MotionCorr/job151/
- 146: Class2D/job146/
- 145: MotionCorr/job145/
- 144: MotionCorr/job144/
- 142: Class2D/job142/
- 141: Class2D/job141/
- 140: MotionCorr/job140/
- 139: MotionCorr/job139/
- 138: Class2D/job138/
- 137: Class2D/job137/

stdout will go here; double-click this window to open stdout in a separate window

stderr will go here; double-click this window to open stderr in a separate window

Number of MPI procs: 6 **Number of threads:** 1 **Submit to queue?** Yes **Queue name:** gpuQ **Queue submit command:** qsub

Number of Nodes: 2 **Number of processes per each node:** 3 **Number of GPUs per node:** 2

Standard submission script: ../../qsub-relion3-gpu.bash **Minimum dedicated cores per node:** 1

Additional arguments:

Print command **Schedule** **Continue!**

6. Relion : GPU 분석 작업 제출

- GPU 작업 템플릿

```
#!/bin/bash          /tem/home/tem/_Applications/relion-3.0.7/gpu/bin/qsub-relion3-gpu.bash

### Inherit all current environment variables
#PBS -V

### Job name
#PBS -N XXXnameXXX

### Queue name
#PBS -q XXXqueueXXX

### GPU use : Specify the number of nodes (XXXextra1XXX), the number of processes per each node (XXXextra2XXX)
#PBS -l nodes=XXXextra1XXX:pnn=XXXextra2XXX:gpus=XXXextra3XXX:XXXqueueXXX

#PBS -o ${PBS_JOBNAME}/run.out
#PBS -e ${PBS_JOBNAME}/run.err

#####
### Print Environment Variables
#####
echo -----
echo -n 'Job is running on node ' ; cat $PBS_NODEFILE
echo -----
echo PBS: qsub is running on $PBS_O_HOST
echo PBS: originating queue is $PBS_O_QUEUE
echo PBS: executing queue is $PBS_QUEUE
echo PBS: working directory is $PBS_O_WORKDIR
echo PBS: execution mode is $PBS_ENVIRONMENT
echo PBS: job identifier is $PBS_JOBID
echo PBS: job name is $PBS_JOBNAME
echo PBS: node file is $PBS_NODEFILE
echo PBS: current home directory is $PBS_O_HOME
echo PBS: PATH = $PBS_O_PATH
echo PBS: PBS_GPUFILE=$PBS_GPUFILE
echo PBS: CUDA_VISIBLE_DEVICES=$CUDA_VISIBLE_DEVICES
echo -----


#####
# Switch to the working directory;
cd ${PBS_O_WORKDIR}/${PBS_JOBNAME}
touch run.out
touch run.err
cd $PBS_O_WORKDIR
#####

### Run:
module load apps/gcc/4.4.7/relion/gpu/3.0.7
mpirun --prefix /tem/home/tem/openmpi-1.8.8 -machinefile $PBS_NODEFILE XXXcommandXXX

echo "Done!"
```

6. Relion : 분석 작업 모니터링

- \$) qstat -a
 - \$) qstat -ns
-
- 템플릿으로부터 만들어진 작업 스크립트 확인
 - 각 작업별 output, error 로그 확인

```
[tem@tem-ui relion_p1]$ cd MotionCorr/
[tem@tem-ui MotionCorr]$ ls
job014  job018  job102  job110  job117  job123  job139  job145  job152
job016  job019  job106  job111  job121  job124  job140  job149  job153
job017  job020  job109  job114  job122  job134  job144  job151
[tem@tem-ui MotionCorr]$ cd job153
[tem@tem-ui job153]$
[tem@tem-ui job153]$ ls
corrected_micrographs_all_rlnAccumMotionEarly.eps      logfile.pdf
corrected_micrographs_all_rlnAccumMotionLate.eps        logfile.pdf.lst
corrected_micrographs_all_rlnAccumMotionTotal.eps       Movies
corrected_micrographs_hist_rlnAccumMotionEarly.eps      note.txt
corrected_micrographs_hist_rlnAccumMotionLate.eps       run.err
corrected_micrographs_hist_rlnAccumMotionTotal.eps      run.job
corrected_micrographs.star                            run.out
default_pipeline.star
job_pipeline.star
[tem@tem-ui job153]$
```

7. cisTEM

- cisTEM is user-friendly software to process cryo-EM images of macromolecular complexes and obtain high-resolution 3D reconstructions from them
- from cisTEM official site <https://cistem.org>
- Module 경로 및 상세 정보 확인 → Module 로딩 → 분석 프로그램/도구 실행
- 활용 가능한 모듈 확인
 - \$> module avail

```
[tem@tem-ui ~]$ module avail
----- /tem/home/tem/Modules/Modules/versions -----
3.2.10

----- /tem/home/tem/Modules/default/modulefiles -----
anaconda/2.7           apps/gcc/4.4.7/relion/gpu/3.0.7p
apps/gcc/4.4.7/cistem/1.0.0  cuda/9.1
apps/gcc/4.4.7/relion/cpu/3.0.7  modules
apps/gcc/4.4.7/relion/gpu/3.0.7  mpi/gcc/openmpi/1.8.8
[tem@tem-ui ~]$
```

7. cisTEM



- cisTEM 프로그램 상세정보

- \$> module show apps/gcc/4.4.7/cistem/1.0.0

```
$> module show apps/gcc/4.4.7/cistem/1.0.0

-----
/tem/home/tem/Modules/Modules/default/modulefiles/apps/gcc/4.4.7/cistem/1.0.0:

module-whatis      Setsups `cistem-1.0.0' environment variables
module           load mpi/gcc/openmpi/1.8.8
prepend-path       PATH /tem/home/tem/_Applications/cistem-1.0.0-beta
conflict          apps/gcc/4.4.7/cistem
-----
```

- cisTEM 모듈 로딩

- \$> module load apps/gcc/4.4.7/cistem/1.0.0

- 로딩된 모듈 확인

- \$> module list

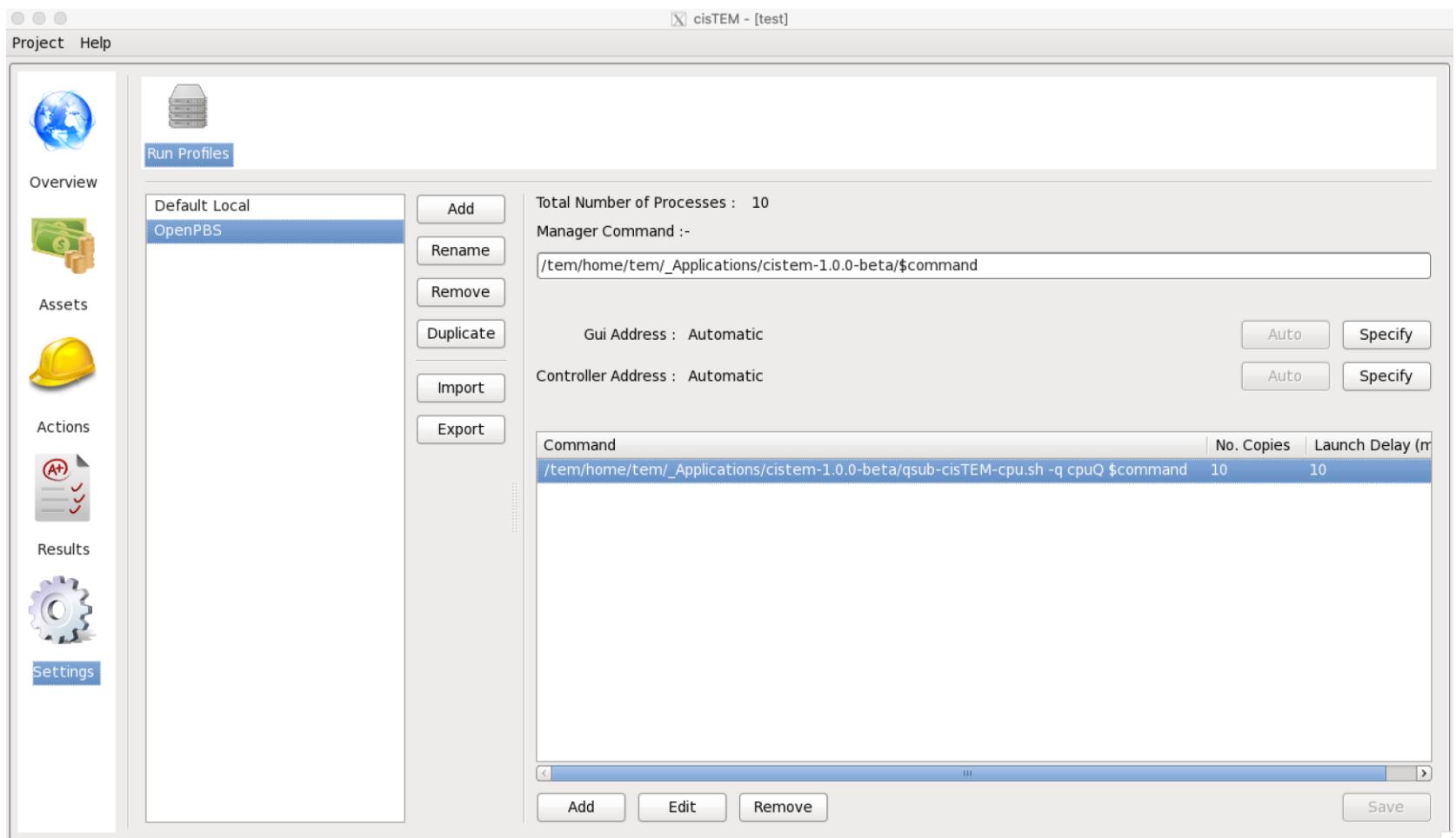
```
$> module load apps/gcc/4.4.7/cistem/1.0.0
$> module list
Currently Loaded Modulefiles:
  1) mpi/gcc/openmpi/1.8.8          2) apps/gcc/4.4.7/cistem/1.0.0
```

- cisTEM GUI 프로그램 시작

- \$> which cisTEM
 - \$> cisTEM

7. cisTEM : 분석 작업 제출

- PBS 배치시스템 연동을 위한 Run Profile 준비
 - Settings → Add → (이름설정) → “Manager Command” → Add → (EDIT)



7. cisTEM : 분석 작업 제출

In cisTEM settings, add a new “Run Profile” (called TORQUE here) with the following parameters :

- Manager Command: /tem/home/tem/_Applications/cistem-1.0.0-beta/\$command
- Gui Address: Automatic
- Controller Address: Automatic
- Command -> Edit:
 - Command: /tem/home/tem/_Applications/cistem-1.0.0-beta/qsub-cisTEM-cpu.sh -q cpuQ \$command
 - No. Copies: 84
 - Delay (ms): 10

/tem/home/tem/_Applications/cistem-1.0.0-beta/qsub-cisTEM-cpu.sh

```
#!/bin/bash
queue=
while getopts ":q:" OPTION
do
  case "${OPTION}" in
    q) queue="${OPTARG}";;
    esac
done
shift $((OPTIND-1))

cat - <<EOF | qsub
#!/bin/bash
#PBS -N cisTEM.${1}
${queue:+#PBS -l nodes=1:ppn=1:${queue}}
${queue:+#PBS -q ${queue}}

module load apps/gcc/4.4.7/cistem/1.0.0
${@}
EOF
```

/tem/home/tem/_Applications/cistem-1.0.0-beta/qsub-cisTEM-cpu-noout.sh

```
#!/bin/bash
queue=
while getopts ":q:" OPTION
do
  case "${OPTION}" in
    q) queue="${OPTARG}";;
    esac
done
shift $((OPTIND-1))

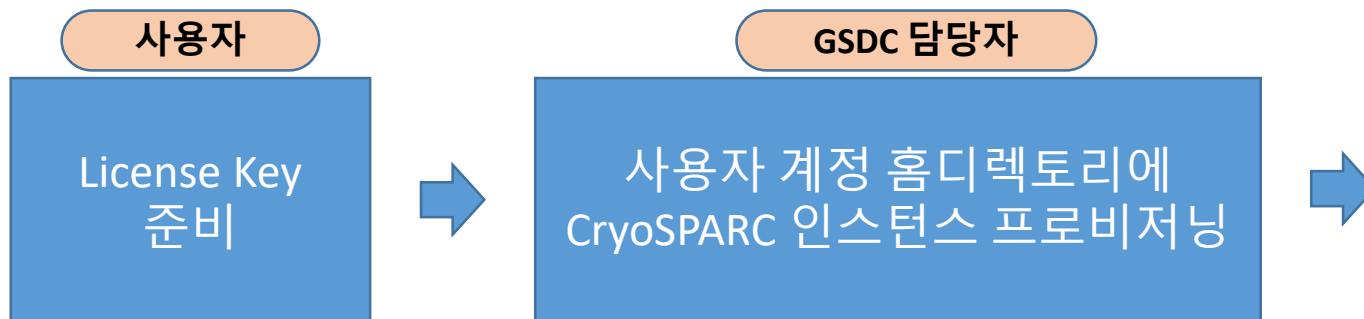
cat - <<EOF | qsub
#!/bin/bash
#PBS -N cisTEM.${1}
#PBS -e /dev/null
#PBS -o /dev/null
${queue:+#PBS -l nodes=1:ppn=1:${queue}}
${queue:+#PBS -q ${queue}}

module load apps/gcc/4.4.7/cistem/1.0.0
${@}
EOF
```

8. CryoSPARC (experimental)

- CryoSPARC is the state-of-the-art platform used globally for obtaining 3D structural information from single particle cryo-EM data (from the official site)
- 다중 사용자간 Data protection 문제 (CryoSPARC 2.x)
- **공식(공용 웹 서비스) 형태가 아닌 사용자별 cryoSPARC 웹 인스턴스 제공**
- CryoSPARC
 - cyrosparc_master + cryosparc_worker + cryosparc_database

8. CryoSPARC (experimental)



- \$ (user) > ls -al ~/.cryosparc

안녕하세요. GSDC 유정록입니다.

요청하신 CryoSPARC 서비스 설정이 완료되었습니다.

CryoSPARC 사용자 계정 및 CRYOSPARC_BASE_PORT 정보입니다.

email : songj3@sdfarm.kr

password : [REDACTED]

CRYOSPARC_BASE_PORT : 39090

<https://tem-docs.readthedocs.io/en/latest/cryoSPARC.html>

사용자 메뉴얼을 참고하셔서, 위 초기 패스워드를 반드시 변경하신 후 사용 바랍니다.

CryoSPARC 웹 application은 SSH 터널링을 사용해 접근하실 수 있습니다.

* 패스워드 변경 관련 : <https://tem-docs.readthedocs.io/en/latest/cryoSPARC.html#trouble-shooting>

* SSH 터널링 관련 : <https://tem-docs.readthedocs.io/en/latest/cryoSPARC.html#launching-cryosparc-instance>

다른 문의사항이 있다면, 연락 부탁드립니다.

감사합니다.

⚠ Warning

!! CAUTION !! DO NOT delete or modify cryoSPARC instance base directory, /tem/home/<user>/.cryosparc. The cryoSPARC base directory contains database. If this directory is deleted, all the project, job and workflow information will be corrupted and lost.

8. CryoSPARC (experimental)

- CryoSPARC 웹 인스턴스 설치 검증 : 환경 변수 확인
 - \$> cryosparcm env

```
$> cryosparcm env

export "CRYOSPARC_HTTP_PORT=39000"
export "CRYOSPARC_MASTER_HOSTNAME=tem-ui.sdfarm.kr"
export "CRYOSPARC_CLICK_WRAP=true"
export "CRYOSPARC_COMMAND_VIS_PORT=39003"
export "CRYOSPARC_INSECURE=true"
export "CRYOSPARC DEVELOP=false"
export "CRYOSPARC_DB_PATH=/tem/home/<user>/.cryosparc/cryosparc_database"
export "CRYOSPARC_HTTP_RTP_PORT=39006"
export "CRYOSPARC LICENSE_ID=<license_key>"
export "CRYOSPARC_MONGO_PORT=39001"
export "CRYOSPARC_MONGO_CACHE_GB=4"
export "CRYOSPARC_HEARTBEAT_SECONDS=60"
export "CRYOSPARC_COMMAND_PROXY_PORT=39004"
export "CRYOSPARC_ROOT_DIR=/tem/home/<user>/.cryosparc/cryosparc2_master"
export "CRYOSPARC_COMMAND_CORE_PORT=39002"
export "CRYOSPARC_BASE_PORT=39000"
export "CRYOSPARC_PATH=/tem/home/<user>/.cryosparc/cryosparc2_master/deps/anaconda/bin:/tem/home/<user>/.cryosparc/cryosparc2_master"
export "CRYOSPARC_LIVE_ENABLED=false"
export "CRYOSPARC_COMMAND RTP_PORT=39005"
export "CRYOSPARC_SUPERVISOR_SOCK_FILE=/tmp/cryosparc-supervisor-627a9991e2f2f069094732dfd78d1696.sock"
export "CRYOSPARC_LD_LIBRARY_PATH="
export "LD_LIBRARY_PATH=:"
export "LD_PRELOAD=/tem/home/<user>/.cryosparc/cryosparc2_master/deps/anaconda/lib/libpython2.7.so.1.0"
export "PYTHONPATH="
export "PYTHONNOUSER SITE=true"
```

8. CryoSPARC (experimental)

- CryoSPARC 웹 인스턴스 설치 검증 : 인스턴스 상태 확인
 - \$) cryosparcm status

- \$) cryosparcm stop
- \$) cryosparcm start

```
$> cryosparcm status
-----
CryoSPARC System master node installed at
/tem/home/<user>/.cryosparc/cryosparc2_master
Current cryoSPARC version: v2.14.2
-----

cryosparcm process status:
app                      STOPPED  Not started
app_dev                  STOPPED  Not started
command_core              RUNNING   pid 171073, uptime 1 day, 5:35:11
command_proxy              RUNNING   pid 171175, uptime 1 day, 5:35:02
command_rtp                 STOPPED  Not started
command_vis                 RUNNING   pid 171170, uptime 1 day, 5:35:03
database                  RUNNING   pid 170997, uptime 1 day, 5:35:14
watchdog_dev                STOPPED  Not started
webapp                     RUNNING   pid 171178, uptime 1 day, 5:35:00
webapp_dev                  STOPPED  Not started
-----

global config variables:
export CRYOSPARC_LICENSE_ID=<license_key>
export CRYOSPARC_MASTER_HOSTNAME="tem-ui.sdfarm.kr"
export CRYOSPARC_DB_PATH="/tem/home/<user>/.cryosparc/cryosparc_database"
export CRYOSPARC_BASE_PORT=39000
export CRYOSPARC DEVELOP=false
export CRYOSPARC_INSECURE=true
export CRYOSPARC_CLICK_WRAP=true
```

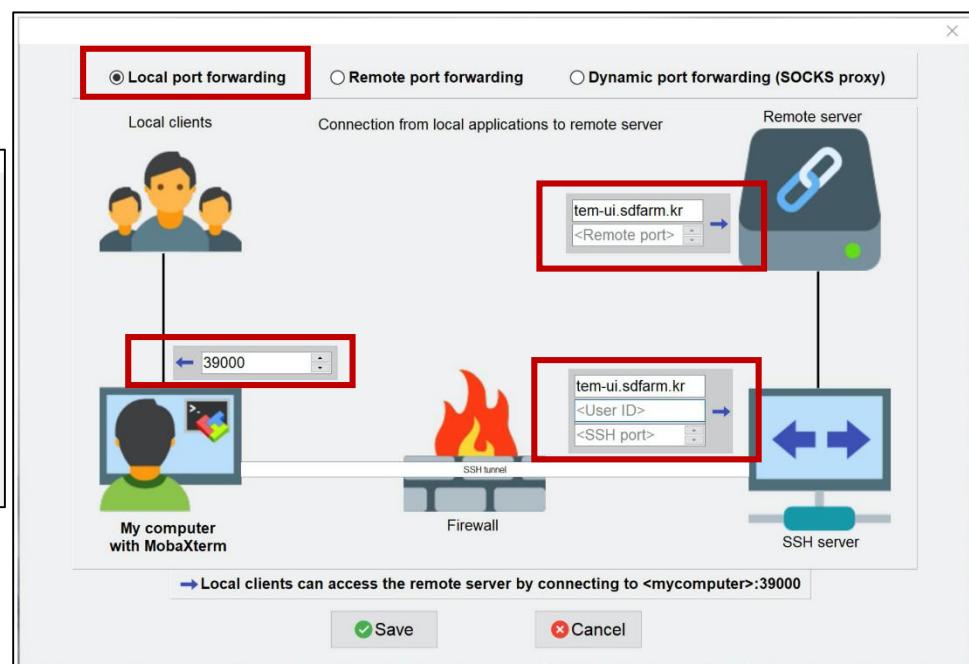
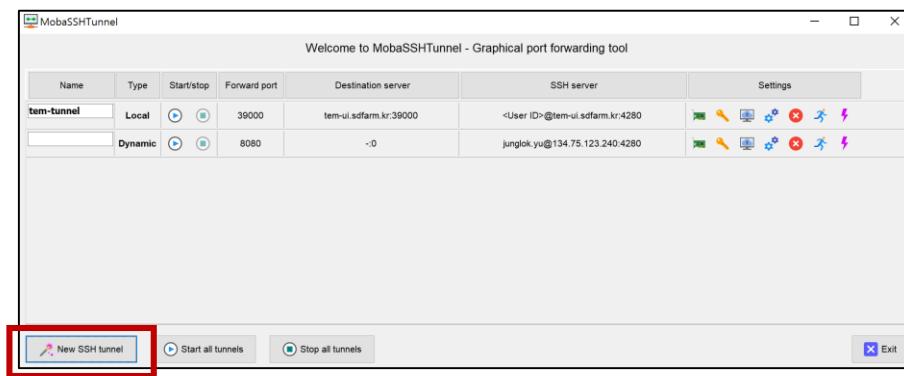
8. CryoSPARC 웹 서비스 연결

```
internet  
[ localhost ]===== [ firewall | tem-ui.sdfarm.kr ]
```

- HTTP sessions over SSH 터널링
- 리눅스/맥 사용자

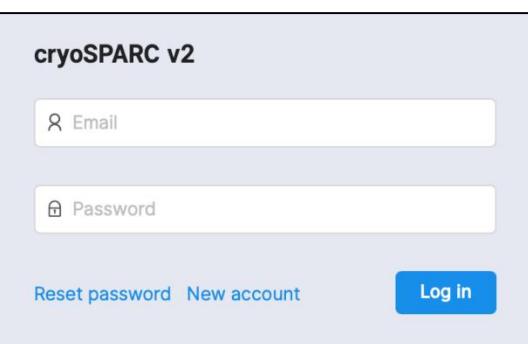
```
localhost$> ssh -N -f -L localhost:39000:tem-ui.sdfarm.kr:<CRYOSPARC_BASE_PORT> -o Port=<ssh_port> <userid>@tem-ui.sdfarm.kr  
## -N : Do not execute a remote command. This is useful option for just forwarding ports.  
## -f : Requests ssh to go to background just before command execution.  
## -L [bind_address:]port:host:hostport
```

- 윈도우 사용자
 - MobaXterm→Tools→MobaSSHTunnel



8. CryoSPARC

웹 브라우저 <http://localhost:39000>



The dashboard shows various statistics: 1 project, 1 workspace, 2 jobs, and 52 completed jobs. It also displays a 'Change Log' for version v2.15.0, dated May 13, 2020, which includes several new features. Below the log is a list of 'My Recent Jobs', showing tasks like 'Patch motion correction (multi)', 'Import Movies', and multiple '2D Classification' tasks. To the right, a 'Links' sidebar provides links to 'General Reference', 'Job Reference', 'Tutorials', '3D Variability Analysis Tutorial', 'Particle Picking Calibration', 'Negative Stain and Phase Plate Data Workflows', 'Discussion Forum', 'EMPIAR', and 'EMDB'. A specific job detail for 'P2 J1' is shown on the far right, including its creation date (Tue Oct 27, 20 04:46:18 PM) and a 'Delete' action.

8. CryoSPARC

- Resource Manager → Instance Information

The screenshot shows the cryoSPARC Resource Manager interface. At the top, there is a navigation bar with icons for Dashboard, Projects, Workspaces, and Resource Manager. The Resource Manager icon is highlighted.

The main area has tabs for Current Jobs, Job History, Instance Information (which is selected), and Notification Manager. Below these tabs, it displays "Lane TEM-FARM (cluster)".

Under "Lane TEM-FARM (cluster)", there is a section titled "Target 1: TEM-FARM". It lists various configuration parameters:

- Cores: ?
- Memory: ?
- GPUs: ?
- Worker bin path: /tem/home/test02/.cryosparc/cryosparc2_worker/bin/cryosparcw
- Hostname: TEM-FARM
- Name: TEM-FARM
- Cache path: Cache quota (MB)
- Cache quota (MB): 10000
- SSH String: Cache Reserve (MB)
- Cache Reserve (MB): 10000

On the right side of the interface, there is a "DATA MANAGEMENT" section with two entries:

- P1: 51.42 GB
- P2: 0 Bytes

At the bottom, there are links for Details and Job Builder, and a message: "New update available: v2.15.0". The user account "test02" is also visible at the bottom right.

8. CryoSPARC

CryoSPARC에서 정의한 변수들

```
#!/usr/bin/env bash
##### cryoSPARC cluster submission script template for PBS
## Available variables:
## {{ run_cmd }}           - the complete command string to run the job
## {{ num_cpu }}            - the number of CPUs needed
## {{ num_gpu }}            - the number of GPUs needed.
##                               Note: the code will use this many GPUs starting from dev id 0
##                               the cluster scheduler or this script have the responsibility
##                               of setting CUDA_VISIBLE_DEVICES so that the job code ends up
##                               using the correct cluster-allocated GPUs.
## {{ ram_gb }}             - the amount of RAM needed in GB
## {{ job_dir_abs }}         - absolute path to the job directory
## {{ project_dir_abs }}    - absolute path to the project dir
## {{ job_log_path_abs }}   - absolute path to the log file for the job
## {{ worker_bin_path }}    - absolute path to the cryosparc worker command
## {{ run_args }}            - arguments to be passed to cryosparcw run
## {{ project_uid }}         - uid of the project
## {{ job_uid }}              - uid of the job
## {{ job_creator }}         - name of the user that created the job (may contain spaces)
## {{ cryosparc_username }} - cryosparc username of the user that created the job (usually an email)
## PBS -l select=1:ncpus={{ num_cpu }}:ngpus={{ num_gpu }}:mem={{ (ram_gb*1000)|int }}mb:gputype=P100
## What follows is a simple PBS script:
```

작업의
자원 요구사항에 대한 지시어들

```
#PBS -N cryosparc_{{ project_uid }}_{{ job_uid }}
{%- if num_gpu == 0 %}#PBS -l nodes=1:ppn={{ num_cpu }}:cpuQ
#PBS -q cpuQ
{%- else %}#PBS -l nodes=1:ppn={{ num_cpu }}:gpus={{ num_gpu }}:gpuQ
#PBS -q gpuQ
{%- endif %}
#PBS -o {{ job_dir_abs }}
#PBS -e {{ job_dir_abs }}
```

```
#####
## Print Environment Variables
#####
echo -----
echo -n 'Job is running on node ' ; cat $PBS_NODEFILE
echo -----
echo PBS: qsub is running on $PBS_O_HOST
echo PBS: originating queue is $PBS_O_QUEUE
echo PBS: executing queue is $PBS_QUEUE
echo PBS: working directory is $PBS_O_WORKDIR
echo PBS: execution mode is $PBS_ENVIRONMENT
echo PBS: job identifier is $PBS_JOBID
echo PBS: job name is $PBS_JOBNAME
echo PBS: node file is $PBS_NODEFILE
echo PBS: current home directory is $PBS_O_HOME
echo PBS: PATH = $PBS_O_PATH
echo PBS: PBS_GPUFILE=$PBS_GPUFILE
echo PBS: CUDA_VISIBLE_DEVICES=$CUDA_VISIBLE_DEVICES
echo -----
```

작업 명령어

```
 {{ run_cmd }}
```

2019 GSDC 데이터 컴퓨팅 SCHOOL

2019 GSDC School of Data Computing

2019.07.29(월)~08.02(금)

국가과학기술인력개발원(KIRD)

모집기간 2019.06.03~06.30

모집인원 30여명(모집기간 종료 후 선발, 개별 통보)

참가비 무료(숙식제공)

신청방법 홈페이지 (<http://school.gsdc.kr/register.html>)를 통한 접수

* 최종 선발자는 개별 연락을 통해 안내되며, 선발 후 개별 연락을 통해 전달되는 "참가확인서"(참가 여부 확인 및 지도교수 추천)를 제출 하셔야 합니다.

교육대상

- 클러스터 컴퓨팅 자원을 활용하는 기초과학 분야 대학(원)생 이상
- 데이터센터 컴퓨팅 시스템에 관심있는 컴퓨터 전공 학부 3학년 이상

교육내용 데이터 공유, 분석 환경 구축 및 활용에 필요한 컴퓨팅 기술 교육 프로그램

- * 자세한 내용 홈페이지 참고 (<http://school.gsdc.kr>)

기타 교육 수료증 발급 및 교육 우수자 포상

문의 gsdc-school@kisti.re.kr



	1일차 (7월 29일)	2일차 (7월 30일)	3일차 (7월 31일)	4일차 (8월 1일)	5일차 (8월 2일)
09:00					
09:30			오리엔테이션	오리엔테이션	오리엔테이션
10:00		오리엔테이션	Lab Review	Lab Review	
10:30		데이터센터 네트워크 김 진 (KISTI-GSDC)	데이터 관리 시스템 1 류건모 (KISTI-GSDC)		
11:00			COFFEE BREAK		
11:30	현장 등록				
12:00	설문조사 1	분산처리시스템 1 배상우 (KISTI-GSDC)	데이터 관리 시스템 2 류건모 (KISTI-GSDC)		
12:30					
13:00				중식 및 휴식	
13:30	스쿨 오리엔테이션	분산처리시스템 2 배상우 (KISTI-GSDC)	Lab 2 "데이터 관리 시스템 구축"	보안 기술 나상호 (KISTI-GSDC)	설문조사 2
14:00	GSDC 소개 및 데이터센터 관리 한희준 (KISTI-GSDC)				
14:30					폐회식
15:00					
15:30	COFFEE BREAK		COFFEE BREAK		
16:00	리눅스 클러스터 구축 기초 1 김정현 (KISTI-GSDC)	Lab 1 "분산처리시스템 구축"			
16:30					
17:00			통합 Lab 1 (조별과제)		
17:30	리눅스 클러스터 구축 기초 2 김정현 (KISTI-GSDC)				시험
18:00					
18:30			석식 및 휴식		
19:00					
19:30					
20:00					

※ 프로그램 일정과 강사는 사정에 따라 변동될 수 있습니다.

감사합니다.
Q&A



Korea Institute of
Science and Technology Information

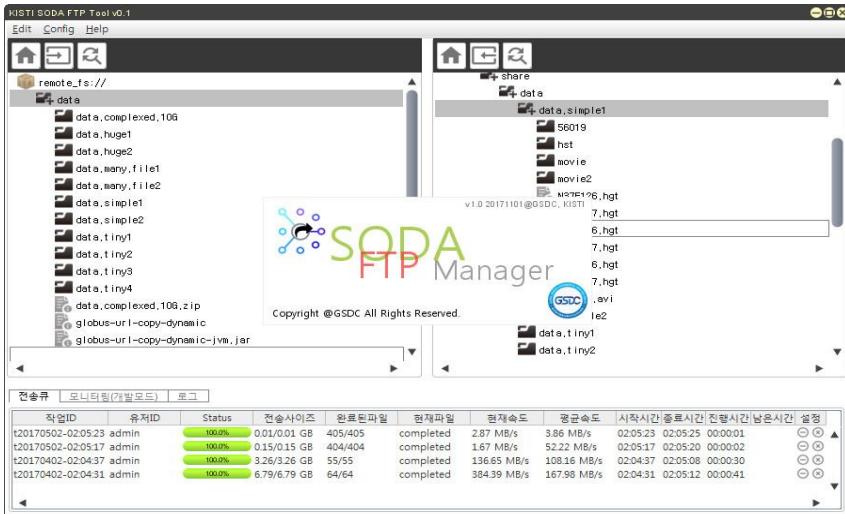
5. 전자현미경 데이터 전송 저장

SODA Toolkit

데이터센터 운영의 자동화 및 사용자 편의성 향상을 위한 도구 (Soda Toolkit) 개발

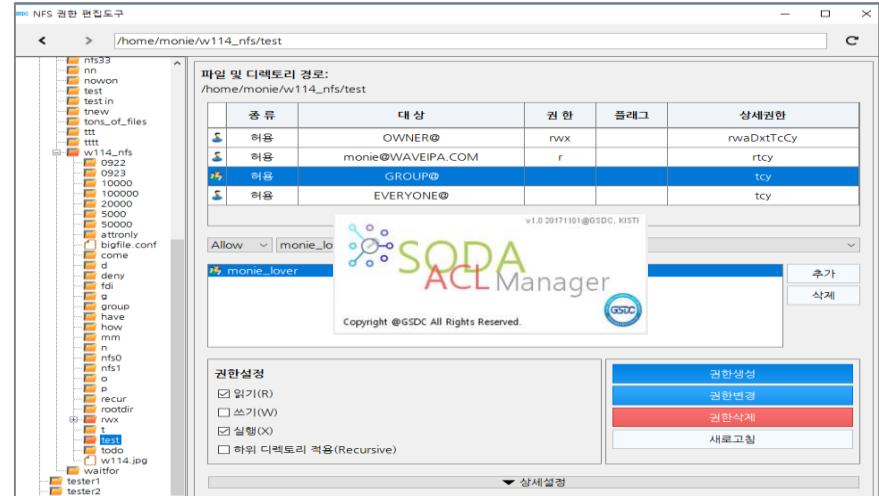
● Soda-FTP (실험데이터 고속 전송)

- 빅데이터 전송에 효과적인 **gridftp** 프로토콜 지원
- 보안(Security) 전송을 위한 **Science-DMZ** 모델 적용
- 멀티플랫폼(OS)을 지원하는 **GUI 제공**



● Soda-ACL (실험 데이터 공유/접근권한 설정)

- HTC/HPC환경의 사용자 통합관리시스템과 연동된 데이터 권한설정 기능 제공
- NAS 전용스토리지 기반의 **NFS 환경** 지원



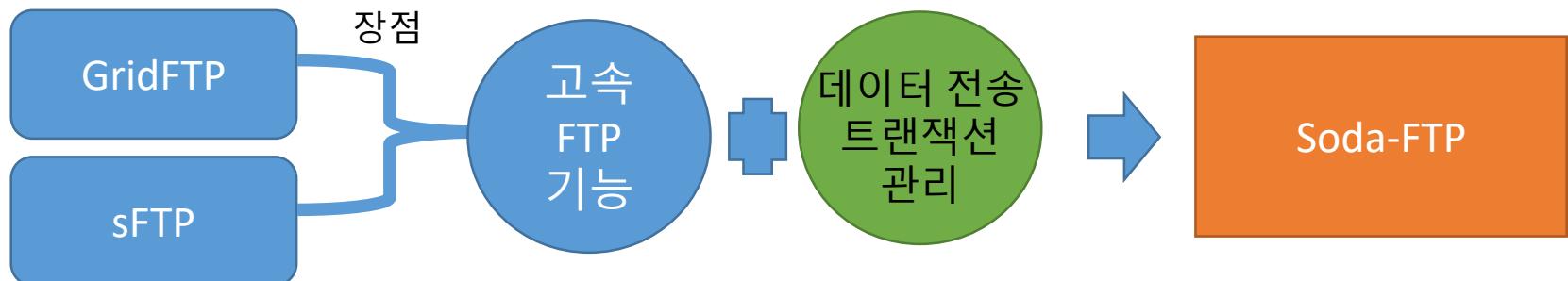
- KBSI 전자현미경 데이터 전송/공유에 활용 (2018.02 ~)
- 포항가속기연구소 4세대 빔라인 대용량 데이터 전송/공유에 활용 (2019.05~)

SODA-FTP (고속 데이터 전송)

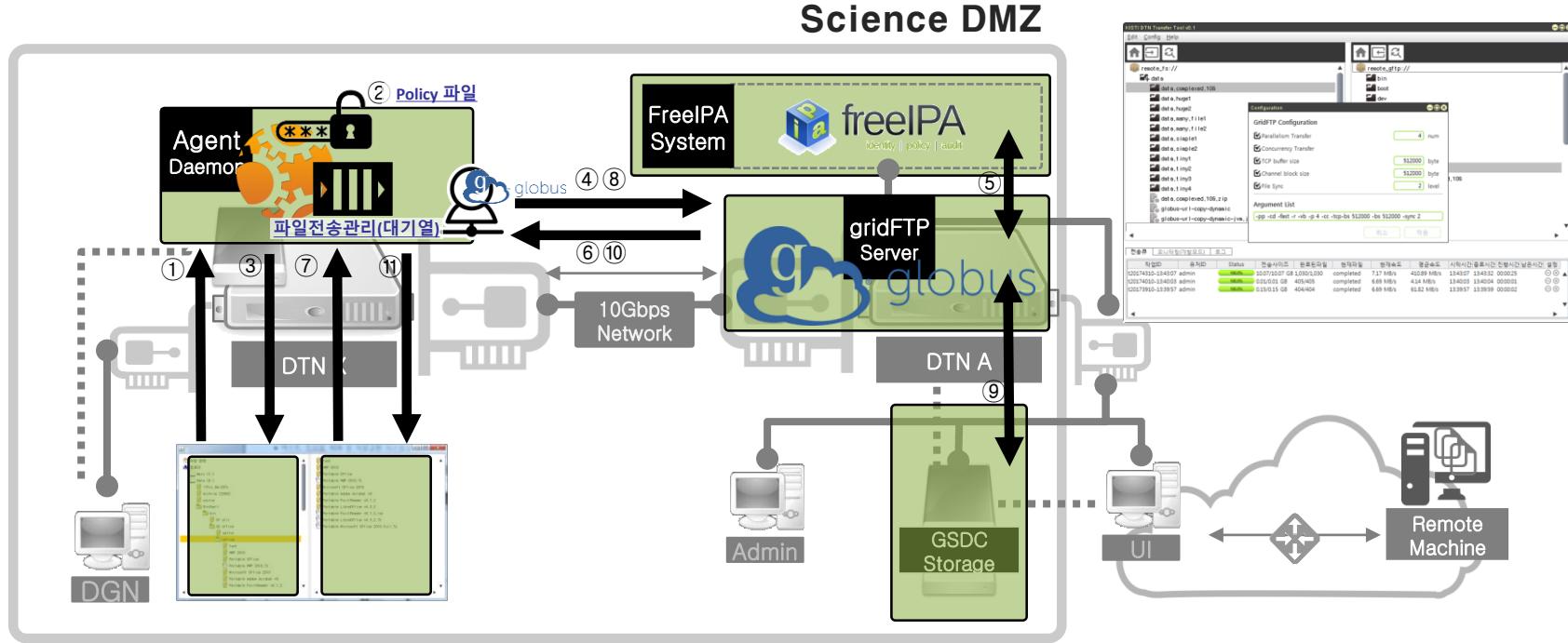
GridFTP 기반으로 시스템 개발

- GridFTP 문제점 : No “interactive GUI client”, single command per invocation
 - Support “interactive GUIclient” for multi-user’s ftp job management
 - Soda-FTP Agent : ftp batch scheduling system with transaction queue

Provide a single point of control GUI for transaction monitoring and management



SODA-FTP (고속 데이터 전송)



- ① DGN에서 전송도구를 수행하여 DTN X 의 Agent Daemon 에 연결함
- ② 이때 policy file 을 통해 자체인증을 수행함 (ID/Password/Home Directory/Target DTN Auth-Info)
- ③ **Source Storage**에 대한 브라우징 기능 수행
- ④ gridFTP Client 는 DTN A 의 GridFTP Server 에 ID/PWD 를 가지고 연결함
- ⑤ gridFTP 는 freeIPA 의 인증절차를 수행
- ⑥ 인증절차 완료시 gridFTP 명령어 수행준비 완료
- ⑦ **Target Storage**에 대한 브라우징 요청
- ⑧ gridFTP Client 를 통한 명령어 전송
- ⑨ GSDC Storage 는 DTN A 와 NFS 로 연결되어 있음
- ⑩ gridFTP Server 는 해당 계정의 FileSystem 정보를 gridFTP Client 에 제공
- ⑪ Agent Daemon 은 해당 정보를 전송도구 로 전달

SODA-FTP 데모

