

# GSDC Promoting Science

대용량데이터허브센터

2021. 03. 24.  
공 병 윤



GSDC 소개



분산 파일 시스템



작업 분산 처리 시스템



인공 지능 관련 분야



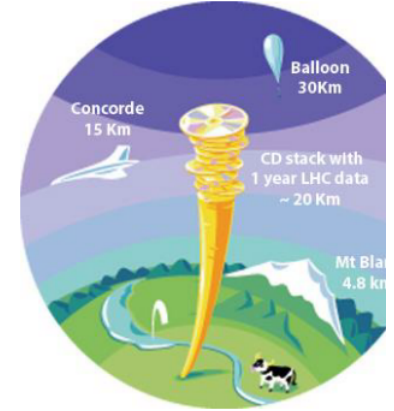
# GSDC 소개

---



선진연구소와  
국제협력

(CERN) 대용량 데이터  
1년 CD 20Km



글로벌 대용량

실험데이터 허브센터

(글로벌) 아시아 대표 허브



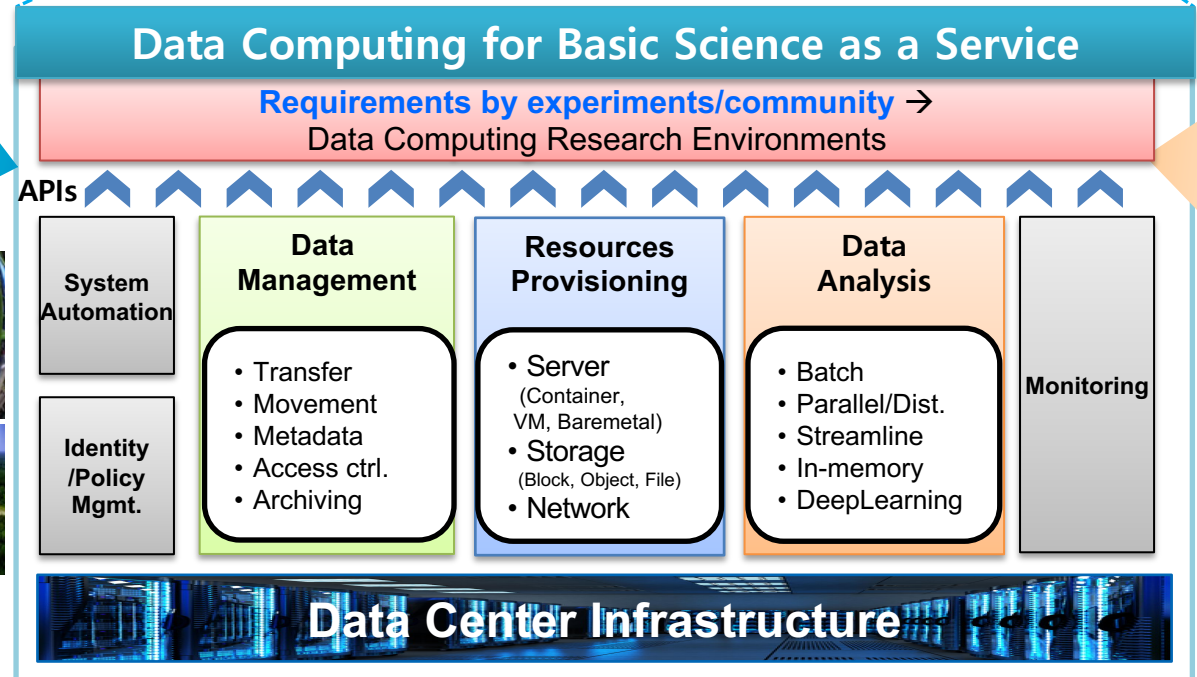
고부가가치  
대형연구시설장비



(국내) 기초과학 데이터 분석  
오픈플랫폼 제공



## 데이터 집약형 기초연구를 위한 데이터 공유 분석 IT인프라(HW, SW) 구축 및 서비스



국내외 고부가가치 대형연구시설장비





High Energy Physics

- 1. ALICE(A Large Ion Collider Experiment)**
  - 빅뱅을 재현하여 우주 초기 물질 상태 연구
  - 40개국, 175개 기관, 2,000명 참여 (국내 40여명)
- 2. CMS(Compact Muon Solenoid)**
  - 새로운 물리현상 탐구 (힉스 입자 입증)
  - 55개국, 232개 기관, 4,800명 참여 (국내 110여명)

Astro Physics

- 3. Belle(KEK)**
  - B 중간자 희귀 붕괴현상 등에 대한 연구
  - 25개국, 107개 기관, 800명 참여 (국내 40여명)
- 4. LIGO(Laser Interferometer Gravitational Wave Observatory)**
  - 중력파를 지상에서 검출하는 실험
  - 18개국, 109개 기관, 1,300명 참여 (국내 40여명)

Particle Physics

- 5. RENO(Reactor Experiment for Neutrino Oscillation)**
  - 원자로에서 방출되는 중성미자 검출(영광원자로)
  - 국내 실험 (40여명)

Medical Science

- 6. Genome Research**
  - 차세대 개인별 맞춤 치료를 위한 유전체 데이터 분석
  - (국내) 서울대, 삼성병원, 국립암센터 등 (80여명)

Biology

- 7. 구조생물학**
  - 전자현미경과 연계한 데이터 분석 서비스 구축
  - 2019년 공식 서비스 (PI 80여명)

General Purpose

- 8. 포항방사광가속기**
  - 포항방사광가속기와 연계한 데이터 분석 서비스 구축
  - 2020년 공식 서비스 (80여명)

G Global  
D Domestic

1

5PB  
1,400억

G

D

2

9PB  
5,000억

G

D

3

5PB  
5,000억

G

D

4

100TB  
1조

G

D

5

200TB  
116억

D

6

3PB  
5,000억

G

D

7

150TB  
250억

D

8

800TB  
5,800억

D



3-4 more domestic experiments under preparation  
e.g. volcanic hazard mitigation, brain research, disease control, etc.

## IT 인프라(HW/SW) 운영을 위한 **오픈 소스 역량 강화**

### 오픈 소스 기반 데이터 집약형 연구를 위한 데이터 컴퓨팅 환경 제공

	ALICE(10)	KiAF(5)	CMS(44)	LIGO(45)	BIO(60)	RENO(15)	Belle(7)	TEM(20)		
도메인 SW	Geant4	Madgraph	KAGALI	Tensorflow	HDF5	CrYOLO				
	AliTrain	Delphes	GWpy	R	CCP4	CryoSPARC				
	ROOT	PhEDEx	LALSuite	Jellyfish	Phenix	cisTEM				
	AliRoot	CMSSW	LSCSoft	Score	CrystFEL	Relion				
	Alien	CVMFS	Singularity	Song	Cheetah	EMAN				
시스템 모니터링	Elasticsearch	Icinga	Splunk	Gmon	grafana	Prometheus	PerfSonar	MRTG		
	Check_MK	Kibana	IBM TS3000 System Console							
	Ganglia	Logstash								
시스템 Frontend	Foreman	RackTables	Software Defined Storage	Cisco Nexus Application SW						
	RHEVM	IPA	Hitachi Storage Navigator	Dell Force10 Application SW						
	oVirt	Stash	EMC OneFS	Juniper JUNOS Application SW						
	Hp SIM	Dell OME	Jira	Hitachi NAS Platform						
	Kubernetes	Mesos	Confluence	IBM System Storage						
시스템 Backend	Auto Deploy Script	Puppet	Condor	Git	Glusterfs	VLAN				
			PBS/maui	Pulp	SAN Multipathing	DNS				
			LDAP	Docker	TSM	DHCP				
			Kickstart	iLO/iDRAC	XRootD	Syslog				
		IPMI	PXE	GPFS	SNMP					
OS	Scientific Linux	CentOS	RedHat	Atomic	Hitachi	EMC	IBM	Cisco	Dell	Juniper
H/W	Computing Server (600 nodes)				Storage (D:25PB, T:3PB)			Network Switch (80)		
	1U	2U	Blade	VM	Disk	Tape	10G	40G	100G	

60여종의 System 오픈S/W

100여종의 M/W, 도메인 오픈S/W

### 98% 오픈 소스 활용



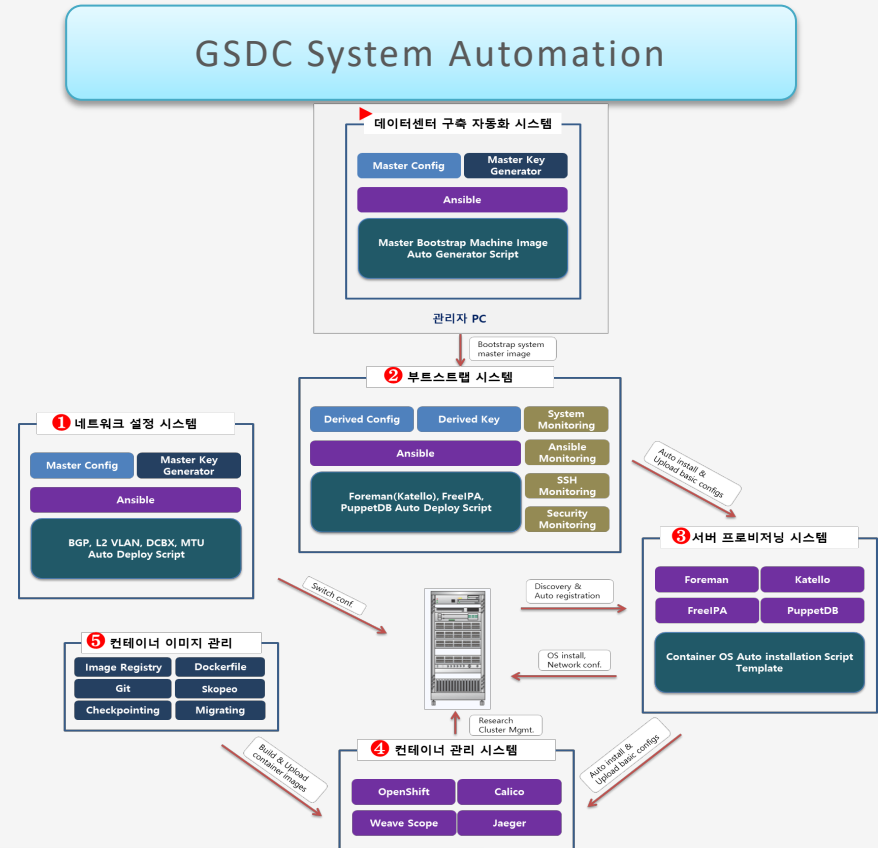
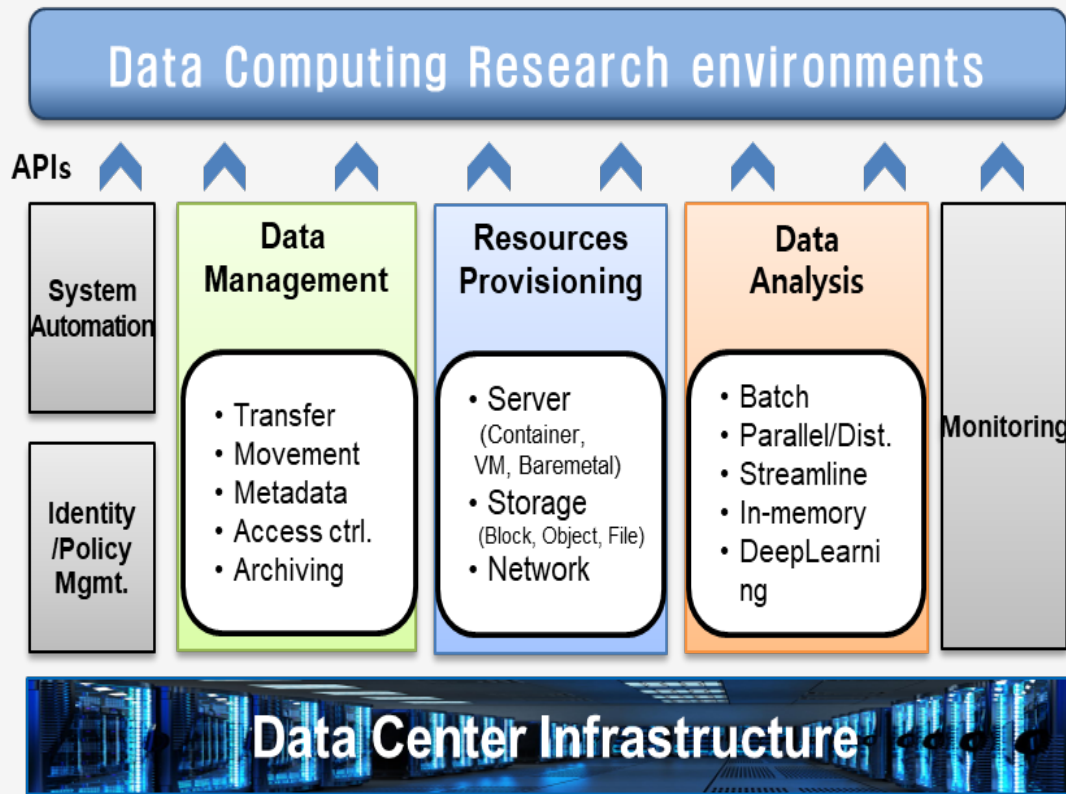
### 약 30억원/년 비용 절감



## 데이터센터 인프라 및 서비스 자동화 시스템 자체 개발 역량

100% 자체 기술을 통한 자동화 시스템 구축  
(약 10만 라인 자체개발 SW)

문제 해결 능력 향상 → 시스템/서비스 신뢰성 향상



## Data Intensive Research Infra as a Service

## 무중단 서비스를 위한 서비스관리 체계 개선

통합운영체계에 맞는 전문인력과 체계를 바탕으로  
24시간 365일 **무중단 서비스** 제공



서비스 가용성

97% ↑



### 복구체계(1시간)

대용량데이터본센터 정보자산 재난복구 대책

최소한 재난복구 대책 개요

모의훈련결과서

모의훈련과제서

재난복구대책 매뉴얼

### 정보보호관리체계

(주요정보통신기반시설 기준)

- 관리적/물리적/기술적
  - 서비스 보안 수준 확립(기술적)
  - 관리체계 및 정책 재정비 계획 수립
- 보안정책 관리 자동화 추진  
(NIST SCAP and OpenSCAP 적용)
- 보안이벤트 가시성 확보  
(보안장비확충 방안 수립)

보안성 제고 전략 수립

### 장애처리 프로세스 개선

[Prometheus, grafana]

서버, 배치시스템 정보 가시화

데이터센터 네트워크 맵

데이터센터 인프라 스카이뷰

데이터센터 모니터링

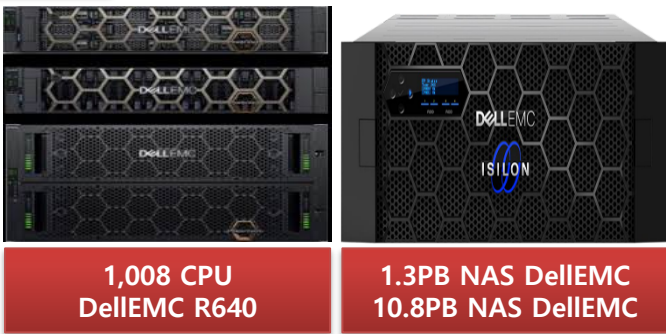
## 데이터 저장 및 분석 인프라 지속적 확장

데이터 저장 및 분석 인프라 [1,276코어(10.2억), 6.9PB 스토리지(9.6억), 3년 평균]

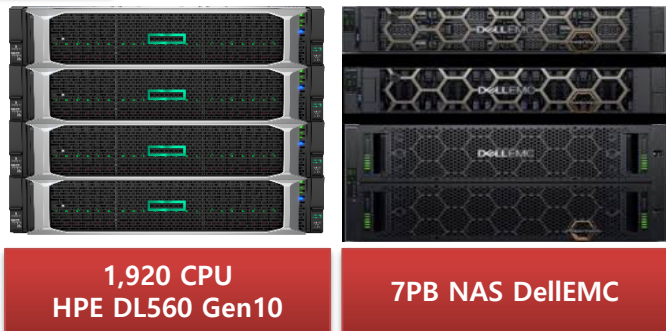
2020

- ① 저장장치 7PB
- ② 컴퓨팅 노드용 서버 1,920코어

2019

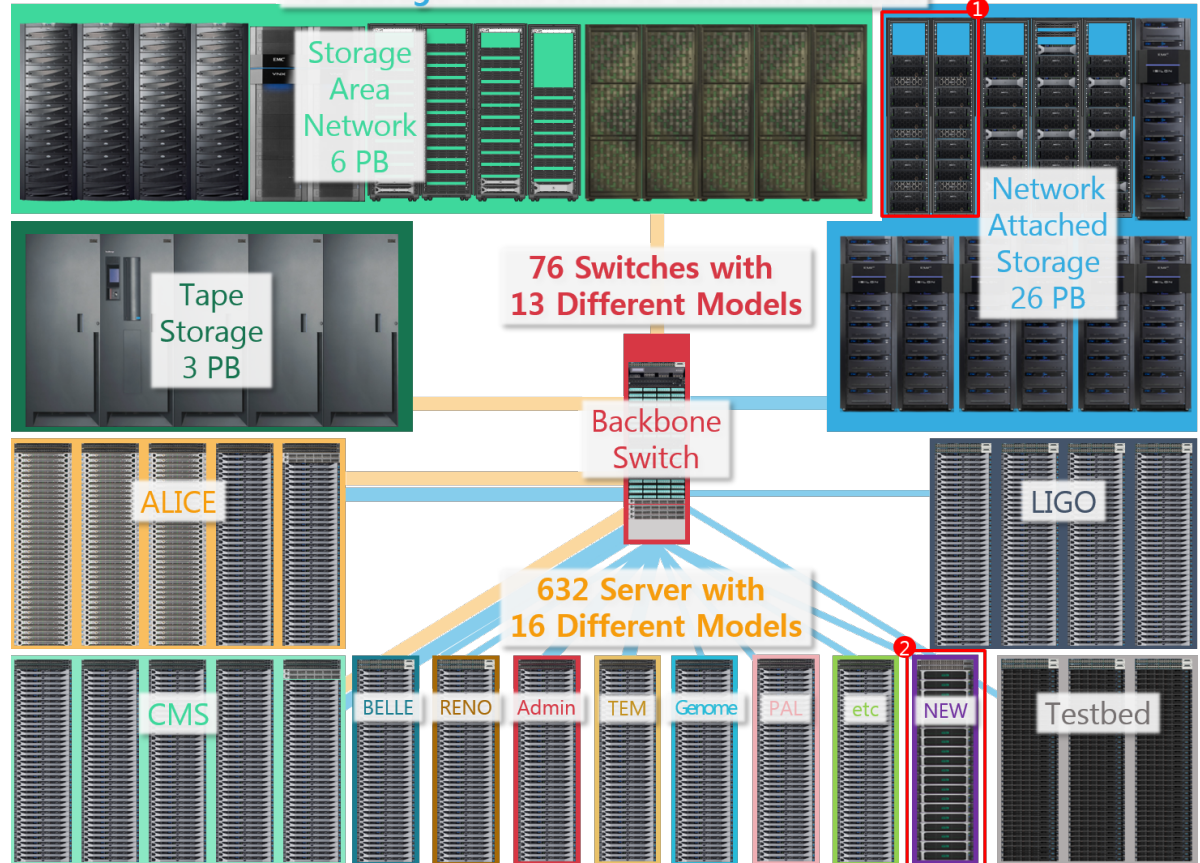


2020



다양한 벤더 제품의  
효과적이고 체계적인 관리가 필요

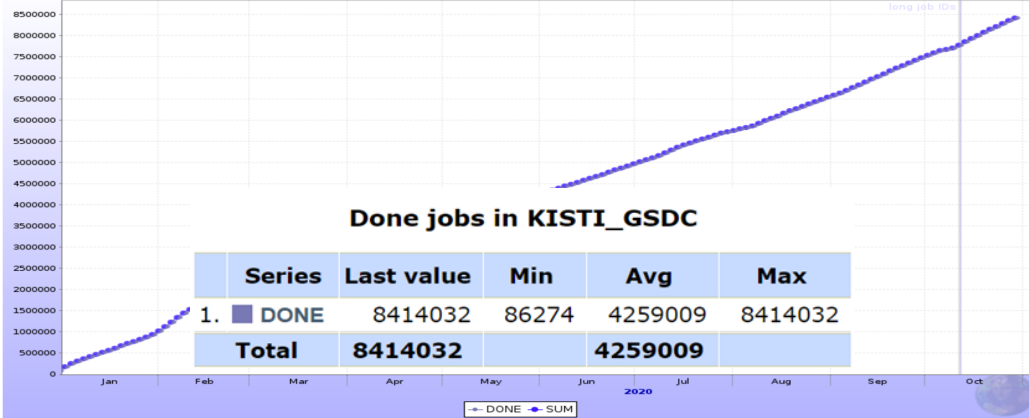
45 Storage Racks with 12 Different Models



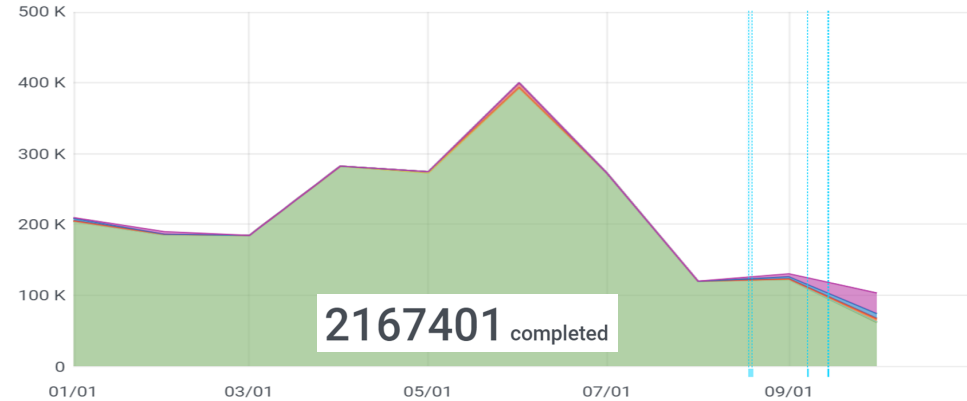


## WLCG Tier-1, Tier-2 데이터 분석 작업 처리

Done jobs in KISTI\_GSDC



※자료: <http://alimonitor.cern.ch/display?image=jfreechart-onetime-15057852129411397188.png>

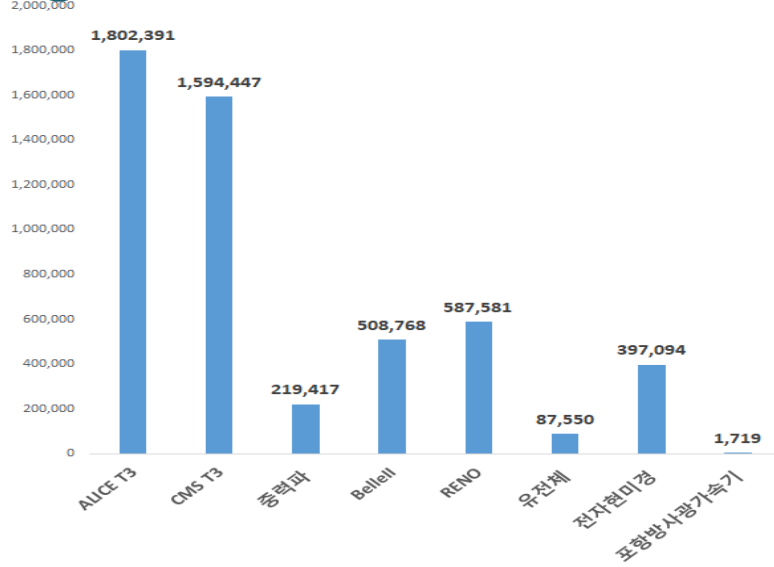


— Analysis — Cleanup — Folding@Home — LogCollect — Merge — Processing — Production  
 ※ 자료 : [https://monit-grafana.cern.ch/d/00000628/cms-job-monitoring-es-agg-data-official?orgId=11&from=1577836800000&to=1604188799000&group\\_by=CMS\\_JobType&var-Tier=All&var-CMS\\_WMTool=All&var-CMS\\_SubmissionTool=All&var-CMS\\_CampaignType=All&var-Site=T2\\_KR\\_KISTI&var-](https://monit-grafana.cern.ch/d/00000628/cms-job-monitoring-es-agg-data-official?orgId=11&from=1577836800000&to=1604188799000&group_by=CMS_JobType&var-Tier=All&var-CMS_WMTool=All&var-CMS_SubmissionTool=All&var-CMS_CampaignType=All&var-Site=T2_KR_KISTI&var-)

**WLCG Tier-1 데이터 작업 처리 841만건**

**WLCG Tier-2 데이터 작업 처리 216만건**

## 국내 연구자 직접 지원 데이터 분석 작업 처리



**국내 연구자 직접 지원 8개 분야  
데이터 작업 처리 519만건**

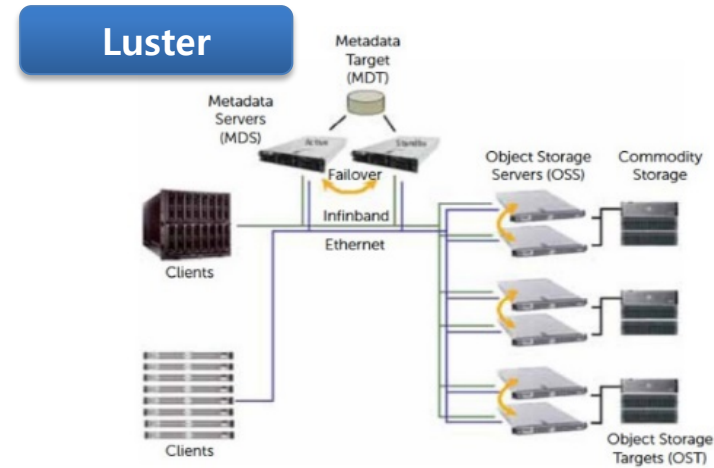


# 분산 파일 시스템

---

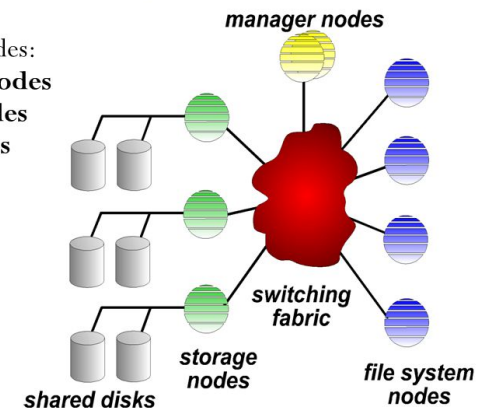
## 분산 파일 시스템이란?

- 여러 대로 나뉜진 디스크들을 하나의 파일시스템(File System)으로 제공하는 기술 및 시스템
- Lustre, GPFS 등의 존재
- 다음 3개의 시스템으로 구성
  - 클라이언트  
구성된 파일시스템을 사용할 수 있는 컴퓨터
  - 메타데이터 서버  
파일의 위치와 상태 정보를 서버로 파일에 대한 접근은 메타데이터 서버에 대한 질의로 시작
  - 스토리지 서버  
실제 데이터를 저장하는 서버로 파일들을 몇 개의 블록으로 쪼개어 각 스토리지 서버에 나눠서 저장



### GPFS Architecture - Special Node Roles

- Three types of nodes:
  - File system nodes
  - Manager nodes
  - Storage nodes



스탠퍼드 선형 가속기 연구소(SLAC)에서 BaBar 실험을 위해 개발

eXtended Root Daemon

- 기존 RootD를 대체

- ROOT라는 HEP 분야에서 많이 활용되는 Analysis Framework의 파일(.root)을 관리하기 위해 개발



- 파일 타입에 대한 제약이 없기 때문에 일반적인 데이터 관리 시스템으로 활용

xrootd 서버와 Objectivity/DB 서버의 Load Balancing을 위해 개발된 olbd 서버로 구성

- Open Load Balancing Daemon

- 현재는 cmsd (Cluster Management Service Daemon)



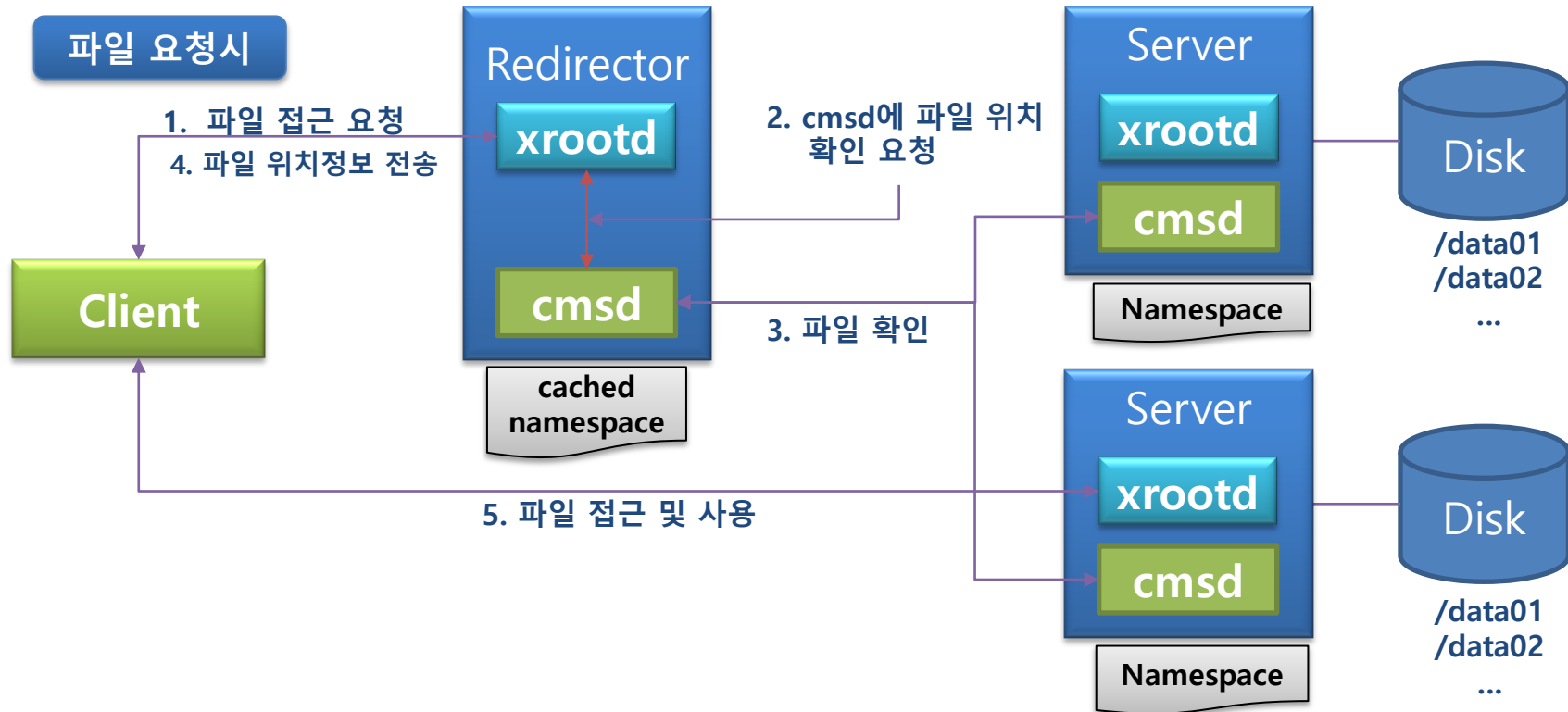
## 장점

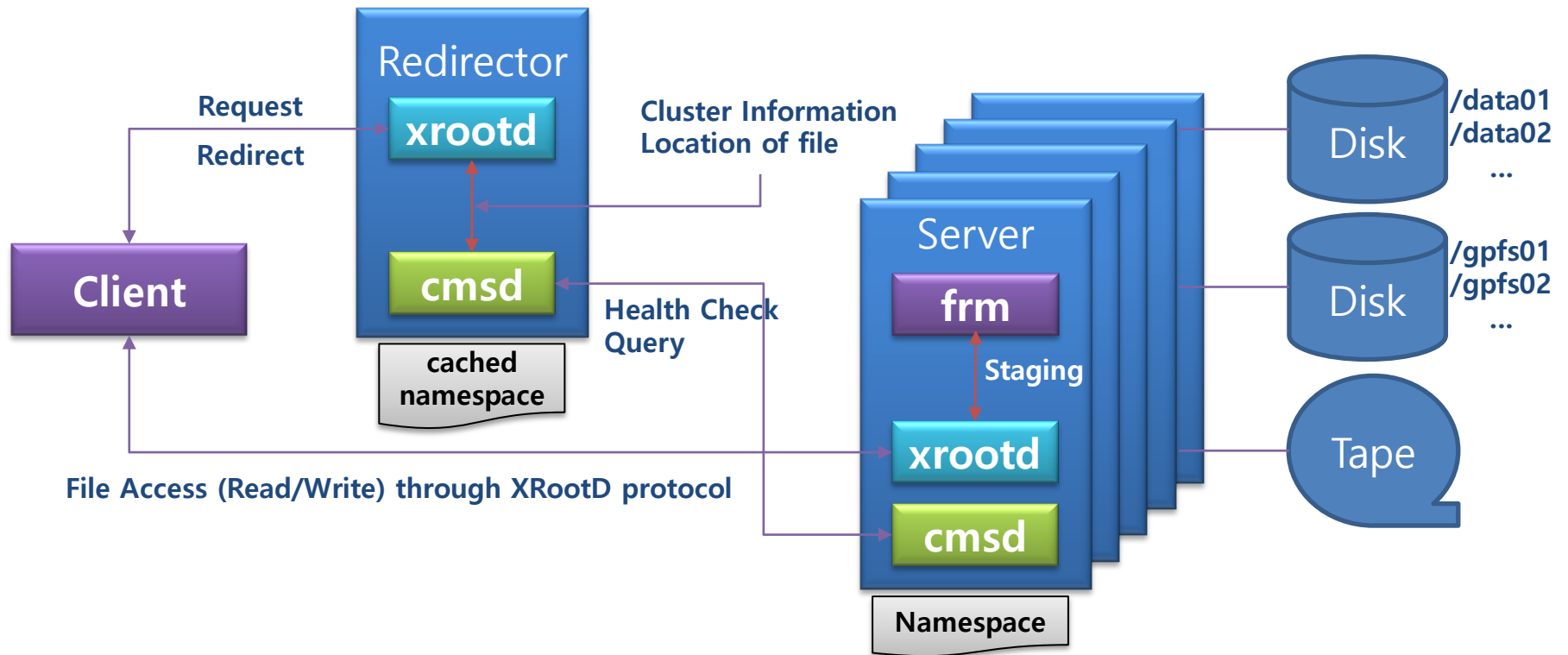
- 공개 SW이기 때문에 쉽게 구할 수 있음
- 간편한 설치 및 설정
- 단순한 구조로 인해 관리 용이
- 메타데이터 서버 관리가 필요 없음
  - 메타데이터 서버 없이 파일의 위치 검색 가능

## 단점

- 파일을 나눠 저장(striping)하지 않기 때문에 읽고 쓰기 성능이 striping을 지원하는 타 소프트웨어와 비교하여 좋지 않음
  - 파일 하나를 저장할 때 하나의 디스크 서버만 사용
    - 나머지 디스크 서버들이 관여하지 않음
    - 여러 파일을 동시에 저장해야만 최고 성능을 사용할 수 있음

(xroot://<fqdn\_redirector:port>/<path>/<filename>)









# 작업 분산 처리 시스템

---

☞ 시스템에서 컴퓨팅 태스크를 실행하고 스케줄링하는 소프트웨어



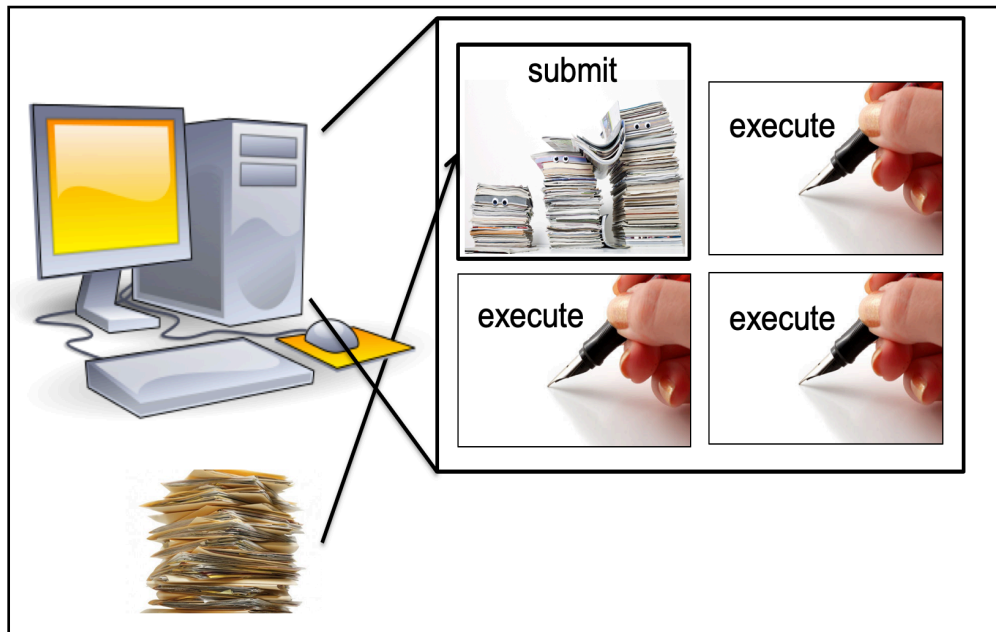
## 개념

- 컴퓨팅을 필요로 하는 작업들을 분산시켜 병렬처리 하기 위한 소프트웨어 프레임워크
- Wisconsin-Madison 대학의 HTCondor 팀에 의하여 개발되었고 현재 Apache 라이선스 2.0하에 오픈 소스 형태로 배포
- 1988년에 처음으로 제안 되었으며, 지난 30년 간 지속적으로 기능 추가 및 버그 수정 과정을 수행

## 특징

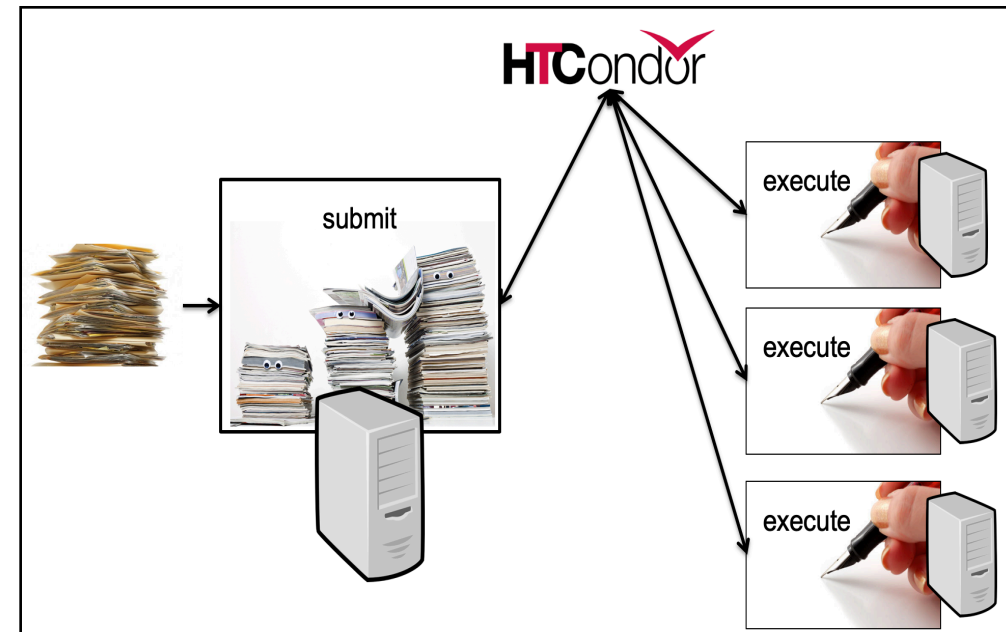
- 호환성 : C 프로그래밍 된 오픈소스로, 다른 프로그램에 비하여 높은 호환성
- 성능 : Job ClassAd에 해당 작업 실행에 필요한 자원량을 명시하여 최대한 작업들을 균등 분배
- 다양한 기능 : flocking, checkpoint

## on One Computer



- 작업(job)을 하나의 머신에 제출
- 하나의 머신에서 모든 작업을 처리

## HTCondor on Many Computers



- 작업(job)을 submit 머신에 제출
- HTCondor 가 모든 컴퓨팅 및 작업(job)을 고려하여 적절하게 스케줄링



- HTCondor가 사용자 대신 작업을 관리하고 실행
- 시스템에 작업을 스케줄링하여, 적절히 분배
- 하나 이상의 시스템에서 여러 명의 사용자가 제출한 작업을 효율적으로 스케줄링

## Job submit

- 사용자가 작업을 submit



# HTCondor

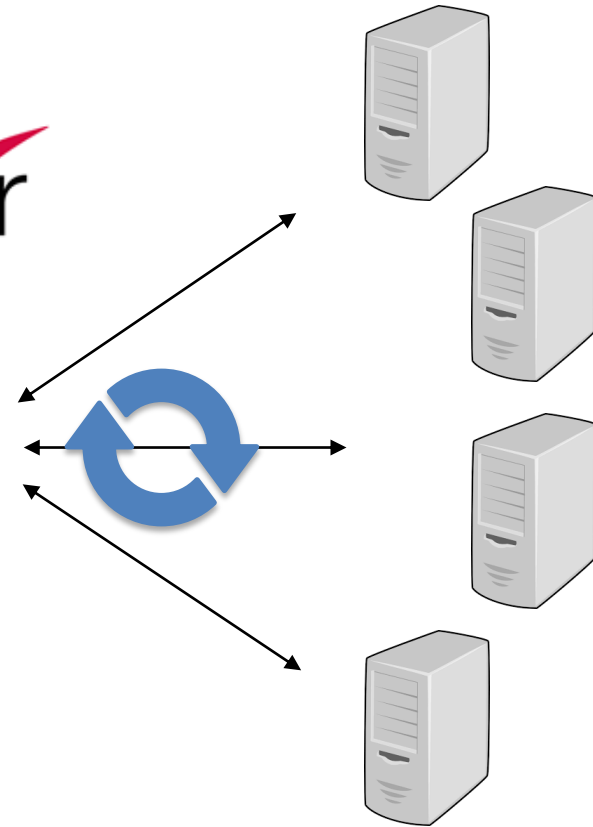


## MN(Master Node)

- Master Node를 통해 작업과 컴퓨팅 자원을 확인



# HTCondor





## Class Ads

- Job ClassAD, Machine ClassAD 정보를 저장 및 모니터링

```
executable = compare_states
arguments = wi.dat.us.dat.wi.dat.out

should_transfer_files = YES
transfer_input_files = us.dat.wi.dat
when_to_transfer_output = ONEXIT

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 20MB
request_memory = 20MB

queue 1
```

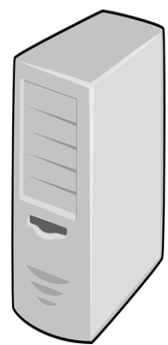
+

=

```
RequestCpus = 1
Err = "job.err"
WhenToTransferOutput = "ON_EXIT"
TargetType = "Machine"
Cmd =
"/home/alice/tests/htcondor_week/compare state
s"
JobUniverse = 5
Iwd = "/home/alice/tests/htcondor_week"
RequestDisk = 20480
NumJobStarts = 0
WantRemoteIO = true
OnExitRemove = true
TransferInput = "us.dat.wi.dat"
MyType = "Job"
Out = "job.out"
UserLog =
"/home/alice/tests/htcondor_week/job.log"
RequestMemory = 20
...
```

**HTCondor configuration\***

**Job Class Ad**



+

=

```
HasFileTransfer = true
DynamicSlot = true
TotalSlotDisk = 4300218.0
TargetType = "Job"
TotalSlotMemory = 2048
Mips = 17902
Memory = 2048
UtsnameSysname = "Linux"
MAX_PREEMPT = ( 3600 * ( 72 - 68 * (
WantGlidein =?= true ) ) )
Requirements = ( START ) && (
IsValidCheckpointPlatform ) && (
WithinResourceLimits )
OpSysMajorVer = 6
TotalMemory = 9889
HasGluster = true
OpSysName = "SL"
HasDocker = true
...
```

**HTCondor configuration**

**Machine Class Ad**

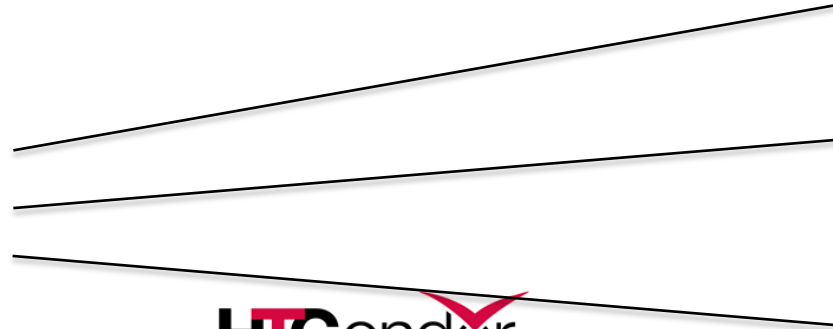
## Match Making

- 사용자 작업 정보 및 워크로드 정보를 매칭



## Job Execution

- SN and WN 간 직접 통신



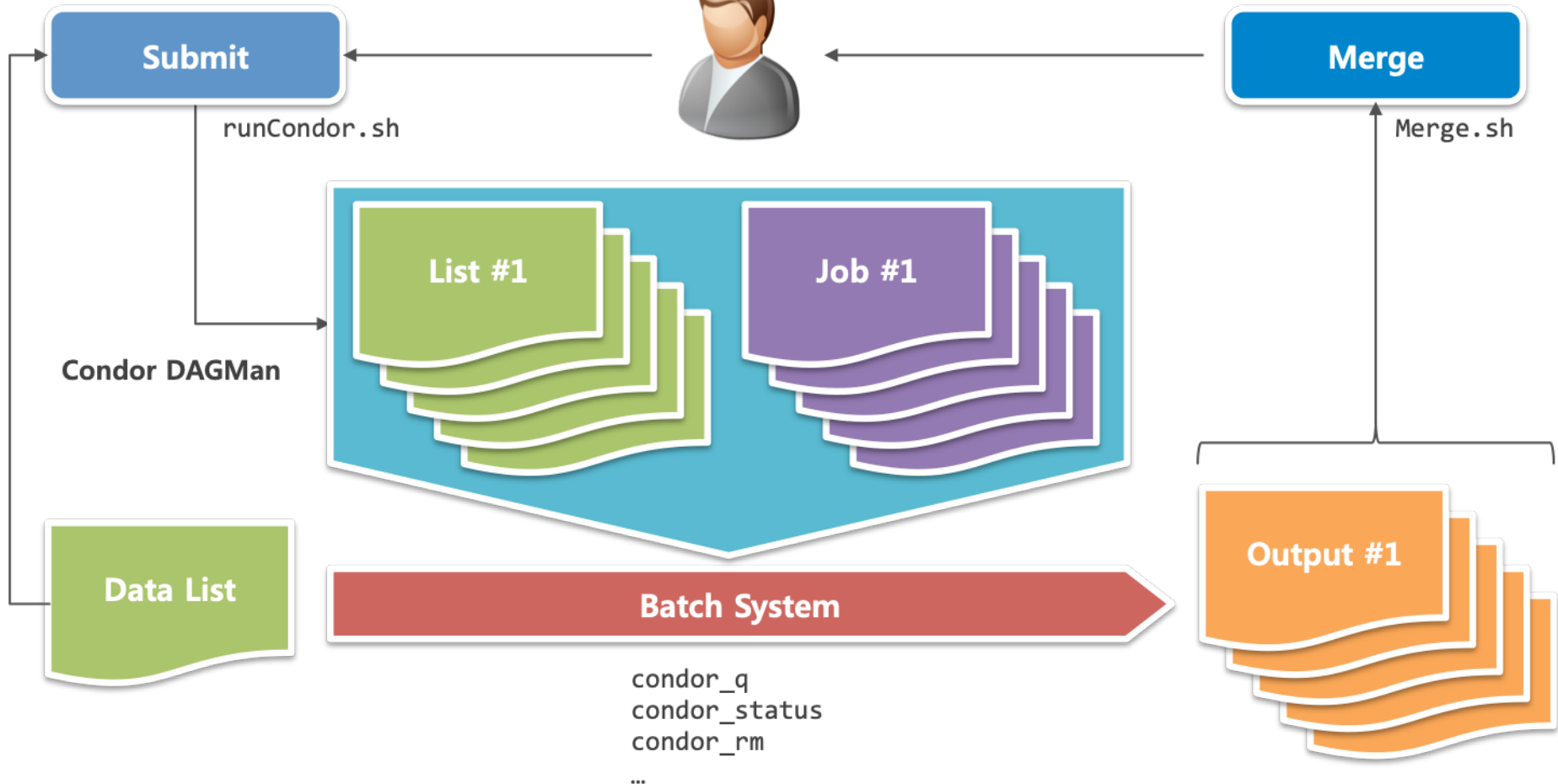
**HTC**ondor

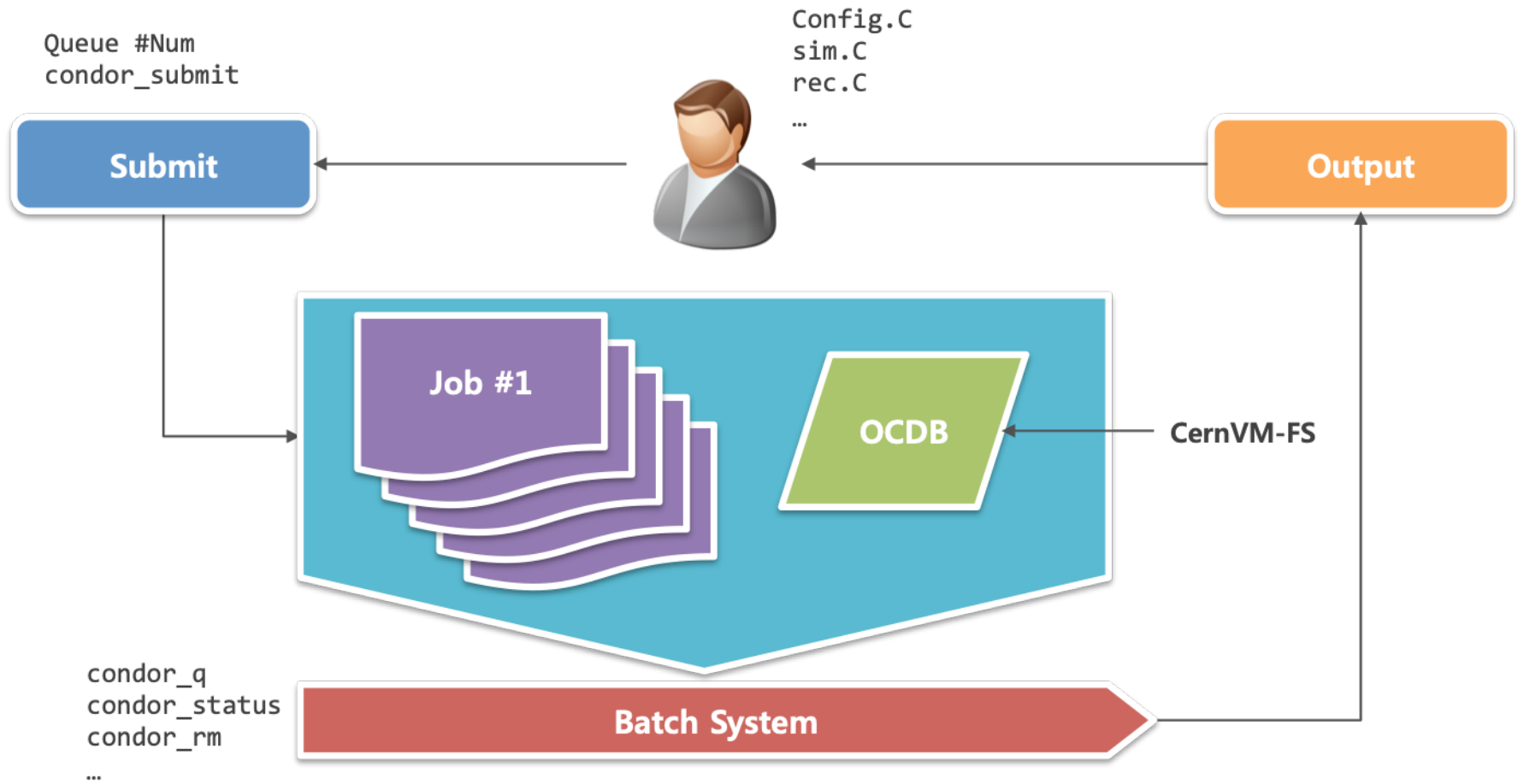


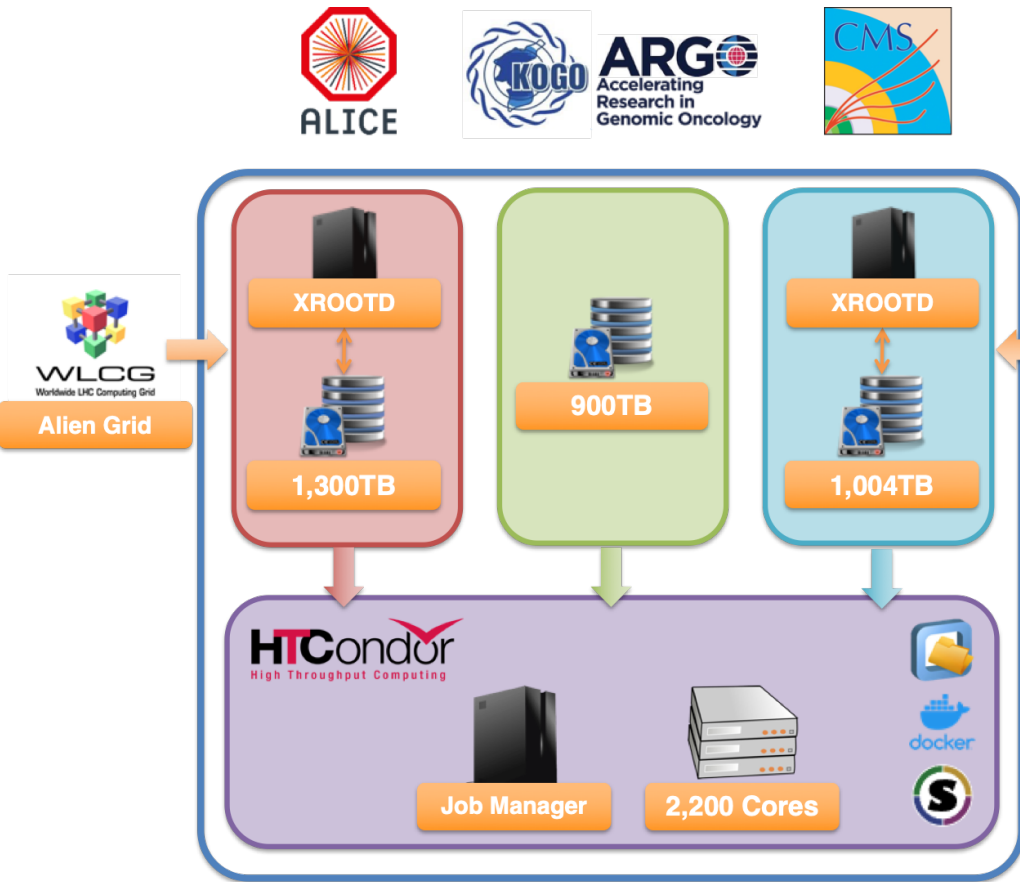
central manager



AliAnalysisTask.cxx  
AliAnalysisTask.h  
runAnalysis.C "local"

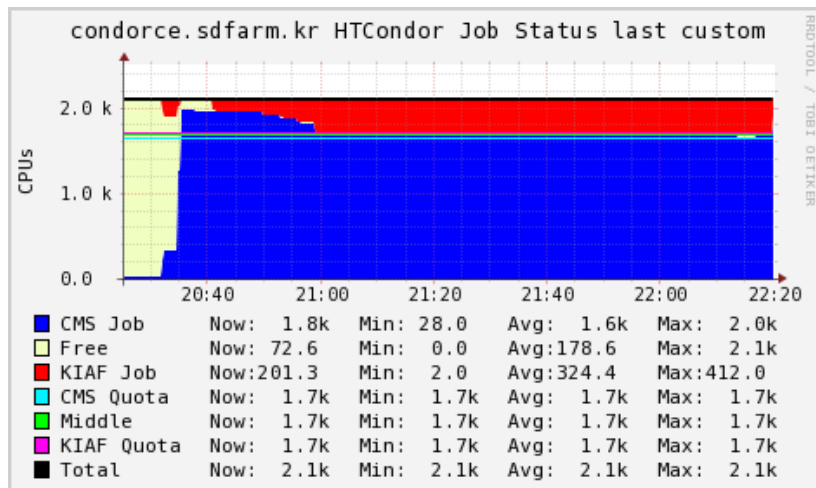
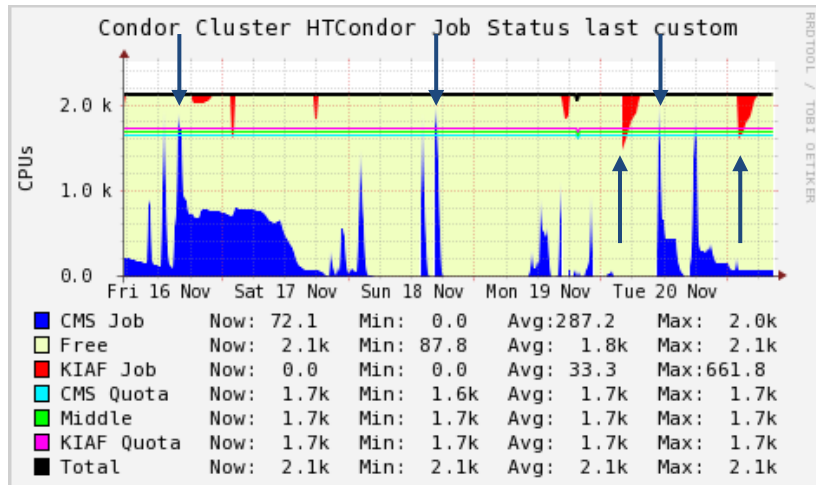






## 시스템 특징

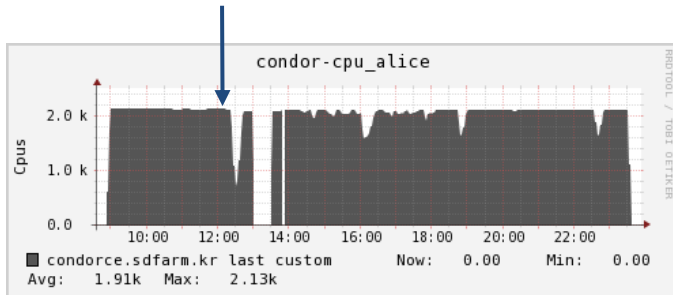
- 통합된 계산노드
  - 컨테이너 기반 가상화 프로그램 Singularity를 이용하여 OS 의존성 해결
- UI를 통한 연구 커뮤니티 구분
- 가상 머신을 이용한 배치시스템 관리 서버 구축
- 연구 그룹별 분리된 스토리지 시스템



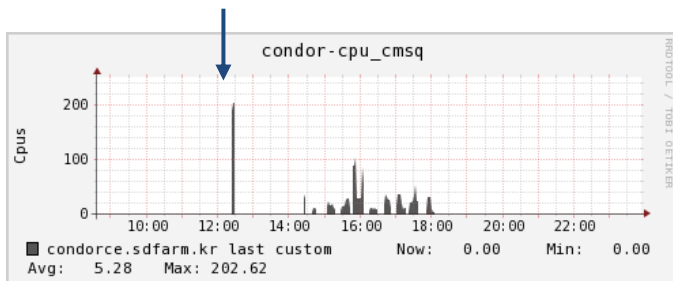
- 사용되고 있지 않은 자원은 어느 그룹 제한 없이 사용 가능
  - 일시적 유휴자원 효율성 증대
- 통합 자원 중 기존 할당 자원 사용 보장
  - 타 커뮤니티 추가 사용 작업 중 가장 최근 작업부터 취소하여 재할당
- 슬롯 메모리 크기에 따라 할당 우선순위 적용
  - 동적 할당 머신 효율성 증대



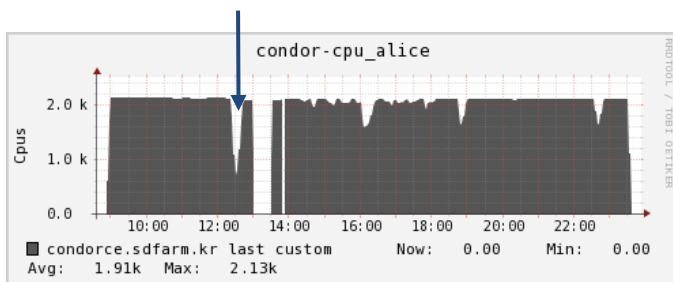
## 0. alice 작업 실행



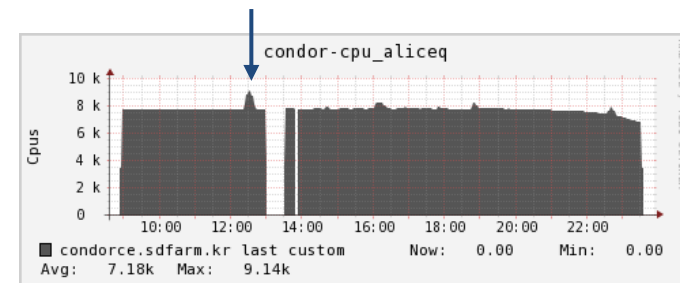
## 1. 새로운 cms 작업 제출



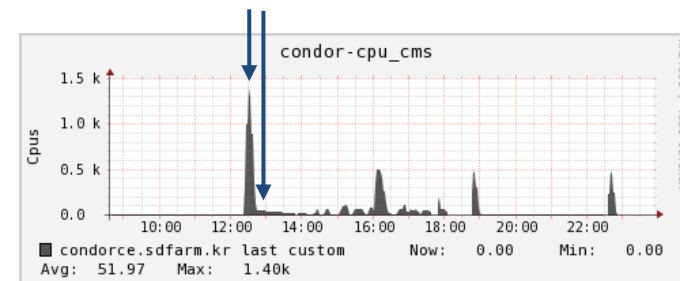
## 2. 기존 alice 작업 취소



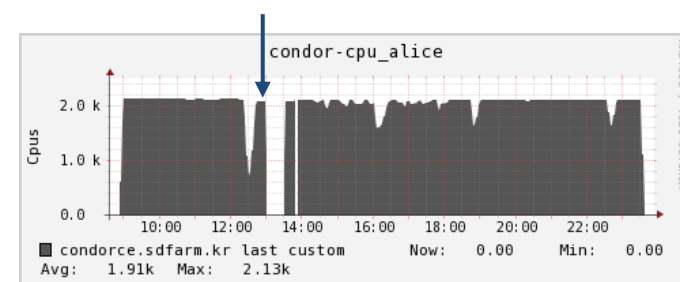
## 3. 취소된 작업은 alice 작업 큐에서 대기



## 4. cms 작업 실행 및 종료



## 5. alice 작업 재시작





# 인공지능 관련 분야

---



## 유지보수 관련

### ○ 유지보수 방법

- 일정에 따른 장비 교체  
-> 문제 발생 가능성↓, 유지보수 비용↑
- 고장에 따른 장비 교체  
-> 문제 발생 가능성↑, 유지보수 비용↓

### ○ 고장 발생 예측

- 문제 발생 가능성↓, 유지보수 비용↓

## 자원 스케줄링 관련

### ○ 작업 종료시간 예측

- 진행중인 작업의 종료시간을 예측

### ○ 스케줄링 정책 업데이트

- 작업 종료 시간을 예측하여 재 스케줄링



**THANK  
YOU**