

HTCondor 입문

한국과학기술정보연구원
대용량데이터허브실

배상욱

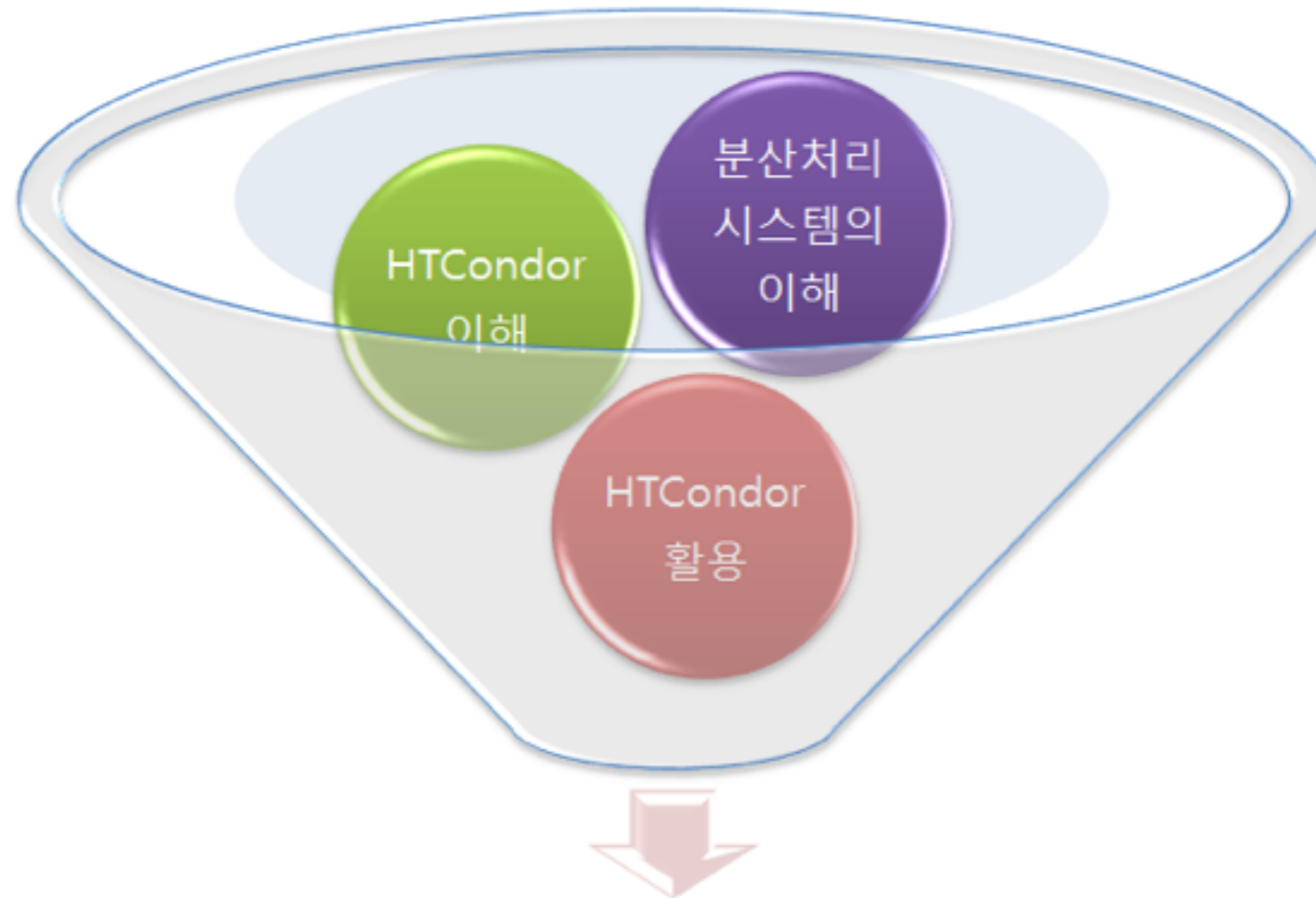
2016.2.3.<수>

Contents



- **HTCondor 소개**
- **HTCondor 설치**
- **HTCondor 설정**
- **Simple Example**
- **실습**

강의 목표



- GSDC 지원 실험의 활용 극대화
- 국내의 물리 실험 환경 개선
- 그리드 환경 시스템의 활용 증가 및 개선

HTC Condor 소개



HTCondor 개념 #1

- 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 28년 간 지속적으로 기능 추가 및 버그 수정 과정을 수행

- 특징

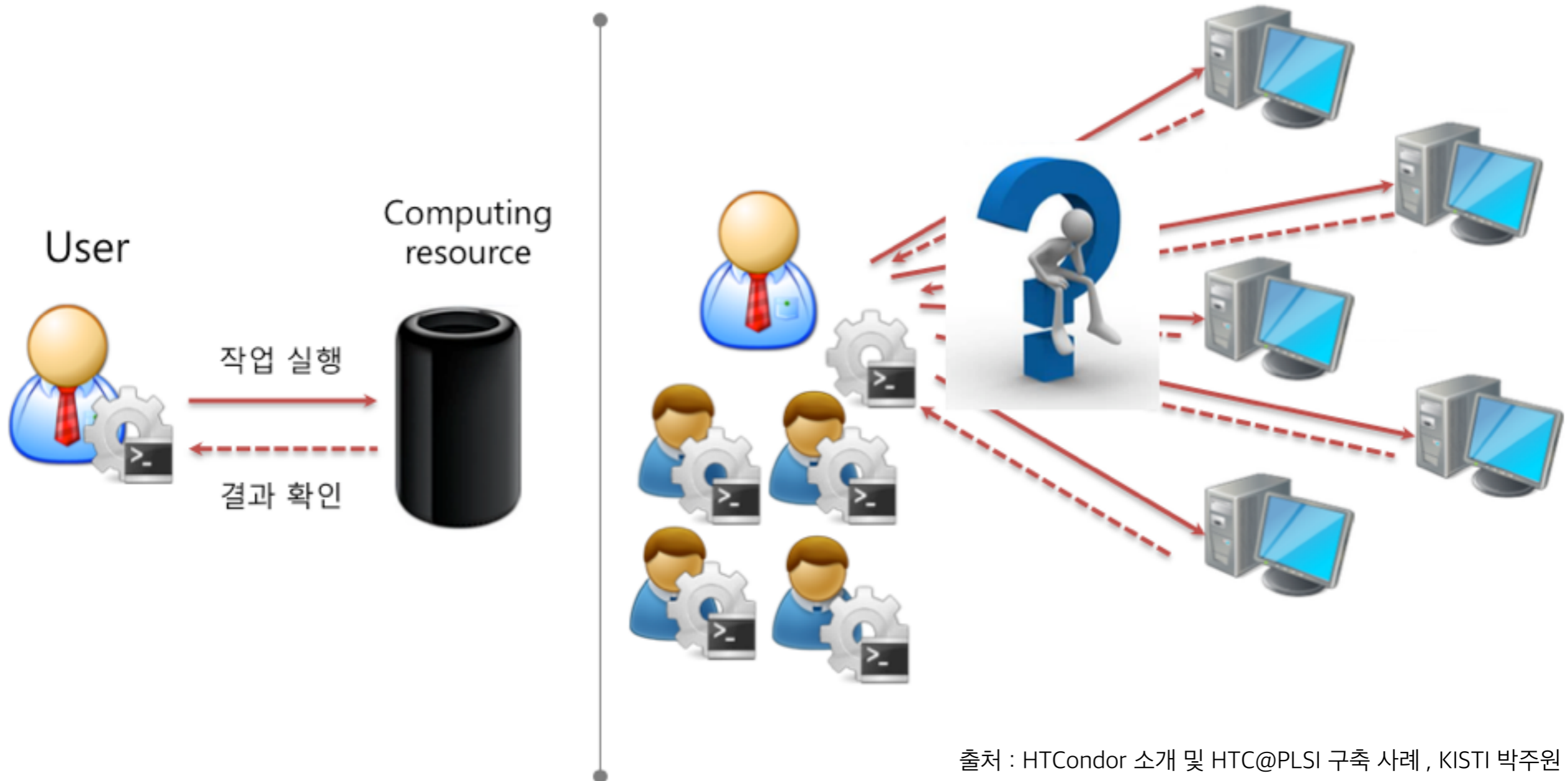
- 호환성 : C 프로그래밍 된 오픈소스로, 다른 프로그램에 비하여 높은 호환성
- 성능 : Job ClassAd에 해당 작업 실행에 필요한 자원량을 명시하여 최대한 작업들을 균등 분배

HTCondor 개념 #2

- 기능
 - 작업 관리 메커니즘
 - 스케줄링 정책
 - 자원 모니터링 및 자원 관리 기능

HTCondor 특징 #1

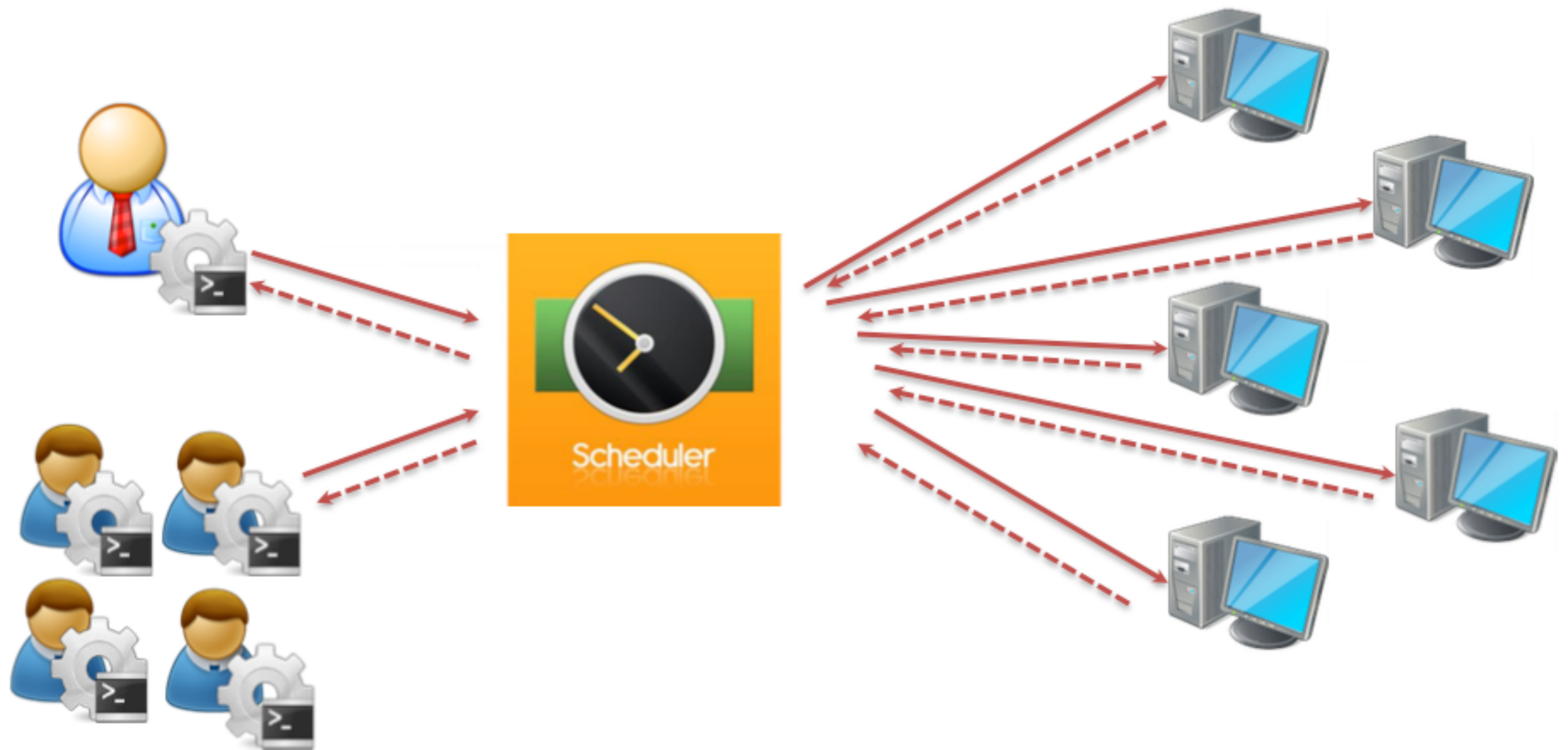
- ClassAD



출처 : HTCondor 소개 및 HTC@PLSI 구축 사례 , KISTI 박주원

HTCondor 특징 #2

- Check Point



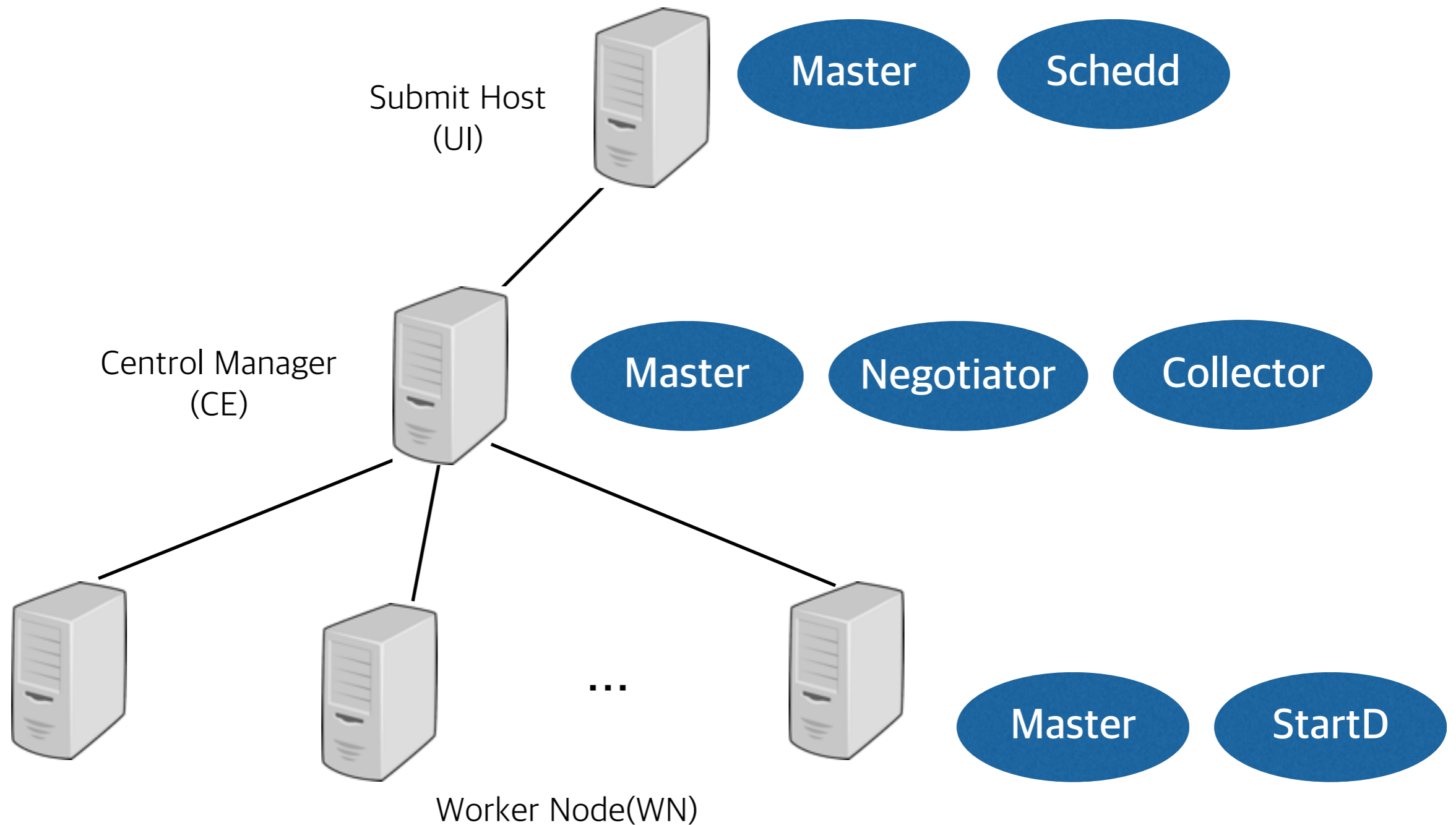
출처 : HTCondor 소개 및 HTC@PLSI 구축 사례 , KISTI 박주원

용어 설명

용어	설명	비고
Job	어떤 한 시뮬레이션, Work 같은 개념. 유닉스 프로세스랑 비슷	
ClassAds	Condor에서 사용하는 유연하고 표현성 높은 언어 (자원요청, 자원제공 등)	
Machine or Resource	클러스터 같은 실제 Job을 돌리는 노드	
Match Making	Job을 Machine에 할당	
Control Manager	일종의 master node, 실제 Job을 중앙에서 Machine에 Matching시킴	Head Node CE
Submit Host	Job 을 submit한 node	Submit Node UI
Execute Host	실제 Job을 담당받아 돌리는 node	Worker Node

HTCondor 구조 #1

- **Condor Daemon Layout**



HTCondor 구조 #2

- **condor_master**



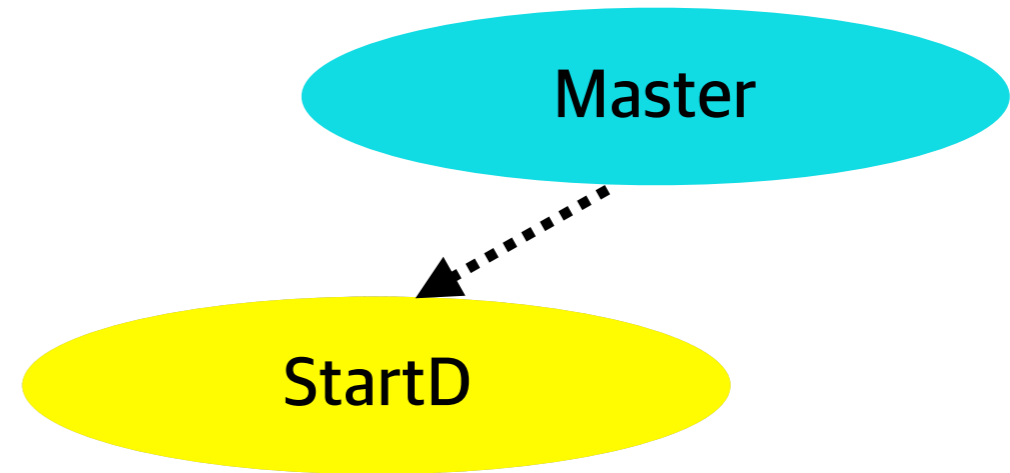
Master

- 기본적인 Daemon
- 모든 Condor Daemon의 시작
- 문제가 있거나 daemon의 종료가 있을 경우, 데몬을 재 시작하고, 관리자에게 email을 전송
- 새로운 바이너리가 발견될 경우, 재시작하여 새로운 버전으로 시작

HTCondor 구조 #3

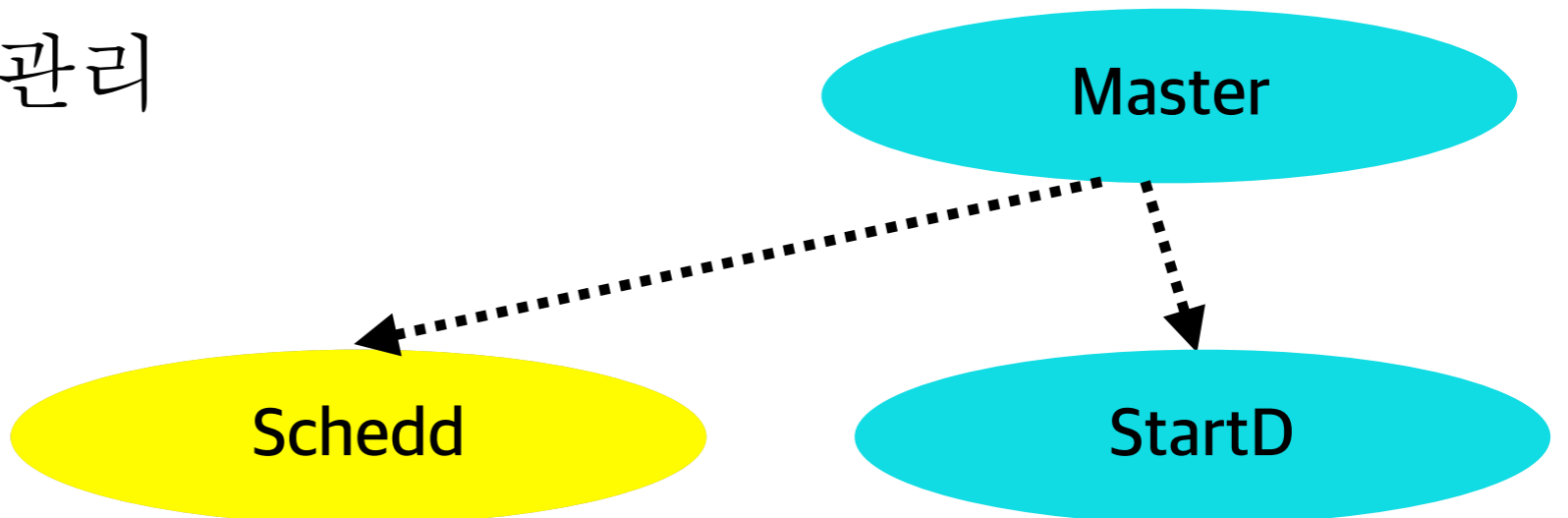
- **condor_startd**

- Job의 시작, 중지, 재시작
- 실제 작업 수행



- **condor_schedd**

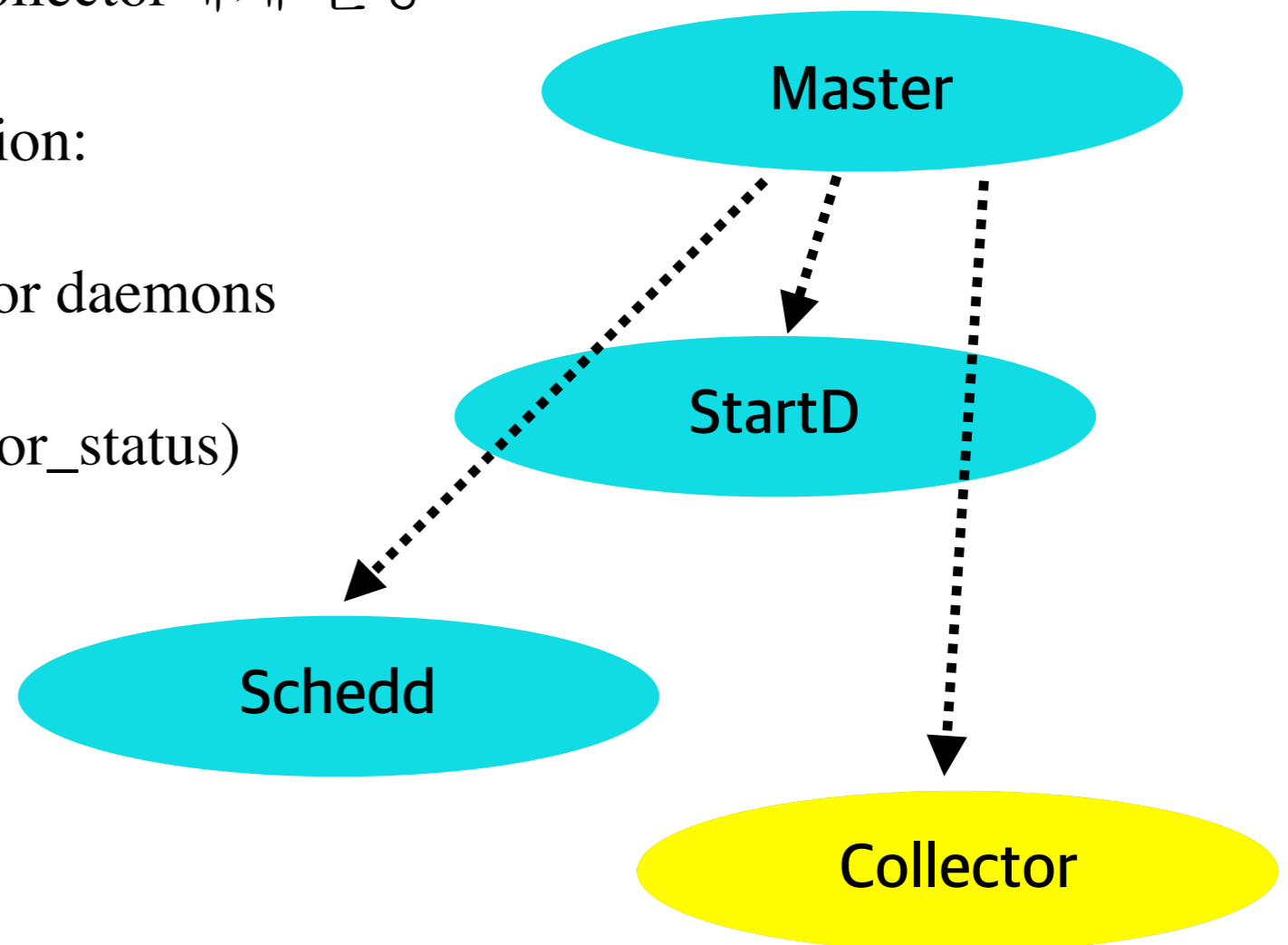
- User command를 수행
 - ✓ ex>condor_submit, condor_rm, condor_q 등
- 큐에 있는 job을 관리



HTCondor 구조 #4

- **condor_collector**

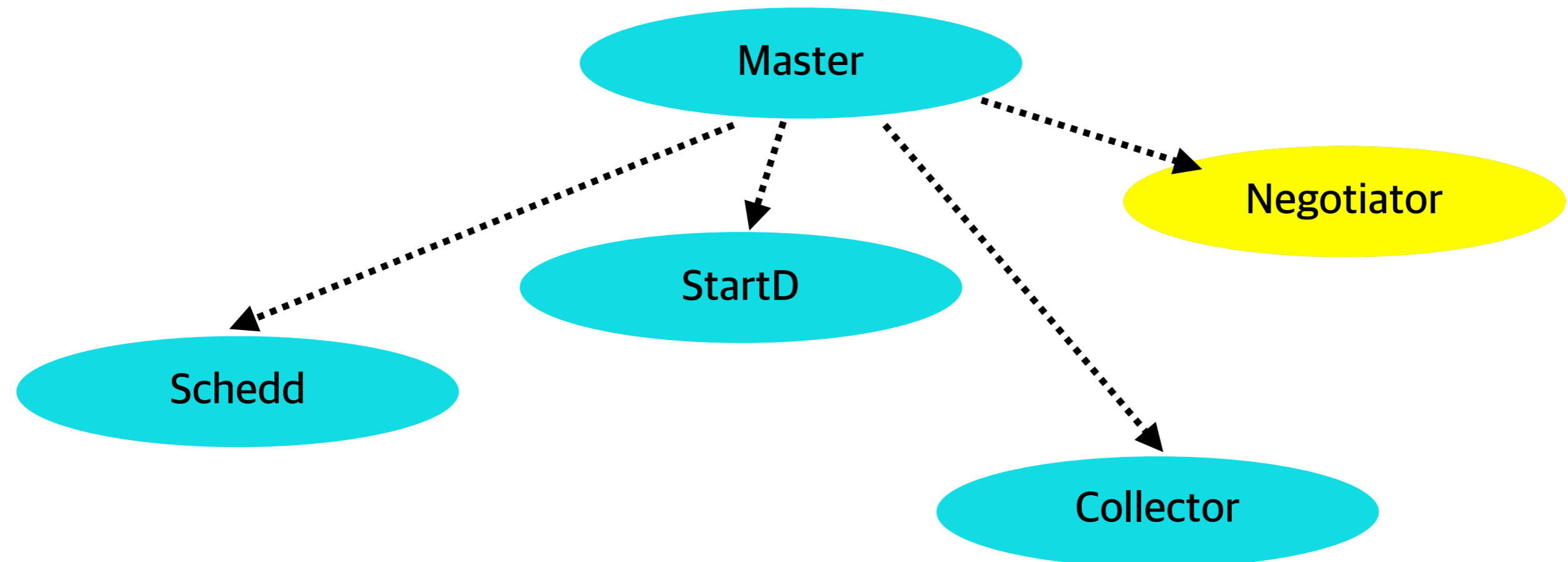
- 모든 다른 condor daemon 으로부터 정보 수집
- 주기적으로 “ClassAD”를 collector에게 전송
- Services queries for information:
 - ✓ Queries from other Condor daemons
 - ✓ Queries from users (condor_status)



HTCondor 구조 #5

- **condor_negotiator**

- condor에서 “matchmaking”을 수행
- collector로부터 사용가능한 상태의 머신 정보를 수신
- 복수의 job에 대한 적절한 분배



HTC Condor 설치



구성 환경

- 노드 정보

	Host Name	IP
Central Manager (CE)	<u>school-xx-ce.experiment.com</u>	X.X.X.X
Submit Host (UI)	<u>school-xx-ui.experiment.com</u>	X.X.X.X
Execute Host (WN)	<u>school-xx-wn.experiment.com</u>	X.X.X.X



Submit Node



Head Node



Worker Node

설치#0

- 서버 접속
 - #ssh studentXX@gw-uiXX.sdfarm.kr -p 4280
 - CE Server (Submit Node) 접속
 - ✓ ssh root@school-XX-ce.sdfarm.kr
 - UI Server (Submit Node) 접속
 - ✓ ssh root@school-XX-ui.sdfarm.kr
 - WN Server (Submit Node) 접속
 - ✓ ssh root@school-XX-wn.sdfarm.kr

설치#1

- 버전 확인

- #cat /etc/redhat-release

```
[wookie@school-15-ce ~]$ cat /etc/redhat-release  
Scientific Linux release 6.6 (Carbon)
```

- 설치 준비

- #wget http://www.cs.wisc.edu/condor/yum/repo.d/condor-stable-rhel6.repo -O /etc/yum.repos.d/condor-stable-rhel6.repo

```
[wookie@school-15-ce ~]$ wget http://pulp-exp.experiment.com/repos/linux/condor/condor-stable-rhel6.repo  
--2015-02-02 04:37:35-- http://pulp-exp.experiment.com/repos/linux/condor/condor-stable-rhel6.repo  
Resolving pulp-exp.experiment.com... 172.16.0.3  
Connecting to pulp-exp.experiment.com|172.16.0.3|:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 185 [text/plain]  
Saving to: `condor-stable-rhel6.repo'  
  
100%[----->] 185 --.-K/s in 0  
  
2015-02-02 04:37:35 (20.9 MB/s) - `condor-stable-rhel6.repo' saved [185/185]  
  
[wookie@school-15-ce ~]$
```

설치#2

- 설치
 - #sudo yum install condor

```
[wookie@school-15-ce ~]$ sudo yum install condor
Loaded plugins: security
Setting up Install Process
htcondor-stable | 2.9 kB 00:00
htcondor-stable/primary_db | 217 kB 00:00
```

...

```
Dependency Installed:
audit-libs-python.x86_64 0:2.3.7-5.el6
cyrus-sasl-md5.x86_64 0:2.1.23-15.el6
gnutls-utils.x86_64 0:2.8.5-14.el6_5
libgomp.x86_64 0:4.4.7-11.el6
libsemanage-python.x86_64 0:2.0.43-4.2.el6
libvirt-client.x86_64 0:0.10.2-46.el6_6.2
nc.x86_64 0:1.84-22.el6
perl-Compress-Raw-Zlib.x86_64 1:2.021-136.el6
perl-Date-Manip.noarch 0:6.24-1.el6
perl-HTML-Tagset.noarch 0:3.20-4.el6
perl-IO-Compress-Zlib.x86_64 0:2.021-136.el6
perl-Package-Constants.x86_64 1:0.02-136.el6
perl-URI.noarch 0:1.40-2.el6
perl-XML-Simple.noarch 0:2.18-6.el6
perl-libwww-perl.noarch 0:5.833-2.el6
setools-console.x86_64 0:3.3.7-4.el6
setools-libs-python.x86_64 0:3.3.7-4.el6
cvs.x86_64 0:1.11.23-16.el6
gettext.x86_64 0:0.17-18.el6
libcgroup.x86_64 0:0.40.rc1-12.el6
libselinux-python.x86_64 0:2.0.94-5.8.el6
libtool-ltdl.x86_64 0:2.2.6-15.5.el6
mailcap.noarch 0:2.1.31-2.el6
perl-Archive-Tar.x86_64 0:1.58-136.el6
perl-Compress-Zlib.x86_64 0:2.021-136.el6
perl-HTML-Parser.x86_64 0:3.64-2.el6
perl-IO-Compress-Base.x86_64 0:2.021-136.el6
perl-IO-Zlib.x86_64 1:1.09-136.el6
perl-Time-HiRes.x86_64 4:1.9721-136.el6
perl-XML-Parser.x86_64 0:2.36-7.el6
perl-YAML-Syck.x86_64 0:1.07-4.el6
policycoreutils-python.x86_64 0:2.0.83-19.47.el6
setools-libs.x86_64 0:3.3.7-4.el6
yajl.x86_64 0:1.0.7-3.el6

Complete!
```

HTC Condor 설정



기본적인 설정 내용

- 일반적인 설정

- /etc/condor/config.d/cluster.conf

- ✓ HTCCondor 환경을 구성하기 위한 기본적인 설정

- ✓ ex> Master Name, Domain Name 등

- 데몬 설정

- /etc/condor/config.d/local.conf

- ✓ 환경 구성에 따른 노드별 데몬 설정

- ✓ ex> MASTER, COLLECTOR, NEGOTIATOR, SCHEDD

Central Manager (CE)

참조 : <https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallCondor>

- **CE 설정**

- `#sudo wget http://pulp-exp.experiment.com/repos/linux/condor/cluster.conf -O /etc/condor/config.d/cluster.conf`
- `#sudo vim /etc/condor/config.d/cluster.conf`
 - ✓ 6 : `UID_DOMAIN = experiment.com`
 - ✓ 13 : `ALLOW_WRITE = *.experiment.com`
 - ✓ 14 : `CONDOR_ADMIN = root@$(FULL_HOSTNAME)`
 - ✓ 16 : `CONDOR_HOST = school-XX-ce.experiment.com`
 - ✓ 19 : `IN_HIGHPORT = 9999`
 - ✓ 20 : `IN_LOWPORT = 9000`
 - ✓ 39 : `REQUIRE_LOCAL_CONFIG_FILE = False`

Central Manager (CE)

- **CE 설정 (con't)**

- **#sudo vim /etc/condor/config.d/local.conf**
- **DAEMON_LIST = MASTER, COLLECTOR, NEGOTIATOR**
- **#ls -al**

```
[wookie@school-15-ce config.d]$ ls -al
합계 16
drwxr-xr-x. 2 root root 4096 2015-02-02 07:16 .
drwxr-xr-x. 4 root root 4096 2015-02-02 06:31 ..
-rw-r--r--. 1 root root  517 2015-02-02 07:09 cluster.conf
-rw-r--r--. 1 root root   44 2015-02-02 07:16 local.conf
[wookie@school-15-ce config.d]$
```

- **#sudo condor_store_cred -c add**

Submit Host (UI)

- UI 설정

- **#sudo wget http://pulp-exp.experiment.com/repos/linux/condor/cluster.conf -O /etc/condor/config.d/cluster.conf**
- **#sudo vim /etc/condor/config.d/cluster.conf**
 - ✓ **6 : UID_DOMAIN = experiment.com**
 - ✓ **13 : ALLOW_WRITE = *.experiment.com**
 - ✓ **14 : CONDOR_ADMIN = root@\$(FULL_HOSTNAME)**
 - ✓ **16 : CONDOR_HOST = school-~~XX~~-ce.experiment.com**
 - ✓ **19 : IN_HIGHPORT = 9999**
 - ✓ **20 : IN_LOWPORT = 9000**
 - ✓ **39 : REQUIRE_LOCAL_CONFIG_FILE = False**

Submit Host (UI)

- UI 설정 (con't)

- #sudo vim /etc/condor/config.d/local.conf

- ✓ **DAEMON_LIST = MASTER, SCHEDD**

- ls -al

```
[root@fermicloud054 config.d]# ll
total 8
-rw-r--r-- 1 root root 1883 Jan 27 10:01 cluster.conf
-rw-r--r-- 1 root root  29 Jan 27 10:07 local.conf
[root@fermicloud054 config.d]#
```

- #condor_store_cred -c add

Execute Host (WN)

- WN 설정

- **#sudo wget http://pulp-exp.experiment.com/repos/linux/condor/cluster.conf -O /etc/condor/config.d/cluster.conf**
- **#sudo vim /etc/condor/config.d/cluster.conf**
 - ✓ **6 : UID_DOMAIN = experiment.com**
 - ✓ **13 : ALLOW_WRITE = *.experiment.com**
 - ✓ **14 : CONDOR_ADMIN = root@\$(FULL_HOSTNAME)**
 - ✓ **16 : CONDOR_HOST = school-XX-ce.experiment.com**
 - ✓ **19 : IN_HIGHPORT = 9999**
 - ✓ **20 : IN_LOWPORT = 9000**
 - ✓ **39 : REQUIRE_LOCAL_CONFIG_FILE = False**

Execute Host (WN)

- **WN 설정 (con't)**

- `#sudo vim /etc/condor/config.d/local.conf`

- ✓ **DAEMON_LIST = MASTER, STARTD**

- `ls -al`

```
[root@fermicloud050 config.d]# ll
total 8
-rw-r--r-- 1 root root 1883 Jan 27 10:02 cluster.conf
-rw-r--r-- 1 root root  29 Jan 27 10:06 local.conf
```

- `#condor_store_cred -c add`

서비스 시작

- 방화벽 설정

- #sudo vim /etc/sysconfig/iptables

- ✓ -A INPUT -s 172.16.XX.0/24 -m state --state ESTABLISHED,NEW -p tcp -m tcp --dport 9000:10000 -j ACCEPT

- ✓ -A INPUT -s 172.16.XX.0/24 -m state --state ESTABLISHED,NEW -p udp -m udp --dport 9000:10000 -j ACCEPT

- 서비스 시작

- Head Node & Submit node & Worker node

- ✓ #sudo service condor start

- ✓ #sudo chkconfig condor on

- 계정 생성

- Head Node & Submit node & Worker node

- ✓ adduser #####

- ✓ passwd #####

테스트 #1

- UI

- #condor_status

```
[wookie@school-13-ui ~]$ condor_status
Name                OpSys      Arch      State      Activity LoadAv Mem      ActvtyTime
slot1@school-13-wn  LINUX     X86_64   Unclaimed Idle       0.000  938    0+00:04:41
slot2@school-13-wn  LINUX     X86_64   Unclaimed Idle       0.000  938    0+00:05:05
Total Owner Claimed Unclaimed Matched Preempting Backfill
X86_64/LINUX      2         0         0         2         0         0         0         0
Total             2         0         0         2         0         0         0         0
[wookie@school-13-ui ~]$
```

- #condor_q

```
[wookie@school-13-ui ~]$ condor_q
-- Submitter: school-13-ui.experiment.com : <172.16.13.1:9486> : school-13-ui.experiment.com
ID      OWNER      SUBMITTED  RUN_TIME ST PRI SIZE CMD
0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended
[wookie@school-13-ui ~]$
```

Simple Example



일반적인 명령어

명령	설명
condor_status	Pool 상태 보기
condor_q	Job Queue 보기
condor_submit	새로운 Job submit
condor_rm	Job 지우기
condor_prio	User Priority 변경
condor_history	완료된 Job 정보 보기
condor_checkpoint	체크포인트
condor_compile	Condor Lib를 이용한 컴파일

Job submission #1

- **Execution File**

```
#!/bin/bash
echo `date`
sleep 5
echo `date`
~
```

- **Descriptions File**

```
#####
#Submit description file for hello program#
#####
Executable      = hello
Universe         = standard
Output           = hello.out
Log              = hello.log
Queue
~
```

Job submission #2

- Descriptions File

- Universe

- ✓ 실행 파일을 수행할 환경 지정

```
#####  
#Submit description file for hello program#  
#####  
Executable      = hello  
Universe        = standard  
Output          = hello.out  
Log             = hello.log  
Queue  
~
```

- ✓ standard : 리모트에서 수행 중인 job의 상태를 호스트에 저장할 수 있는 환경으로 **checking-point**와 **remote system call** 가능. standard 환경에서 실행되는 파일은 반드시 condor_compile과 링크되어야 함

- ✓ vanilla: condor_compile로 링크될 수 없는 경우로 셸스크립트가 대표적인 예 (일반적인 경우 사용)

- ✓ grid: 그리드 Job을 실행할 때 사용

- ✓ java: Java 프로그램을 실행하는 환경

Job submission #3

- Descriptions File

- Universe (con't)

- ✓ local: 경량의 Job을 바로 수행해보기 위한 사용환경으로 리모트 시스템과 매치메이킹(Match Making)을 하지않고 호스트에서 바로 수행
 - ✓ scheduler: local과 유사하나 condor_stardd 데몬이 Job 을 핸들링하지 않는 차이점이 존재
 - ✓ parallel: MPI Job을 수행하기 위한 환경
 - ✓ vm: VMWare나 Xen과 같은 가상머신을 수행하기 위한 환경

- Output

- ✓ 실행결과 저장할 곳 지정

```
#####  
#Submit description file for hello program#  
#####  
Executable      = hello  
Universe        = standard  
Output          = hello.out  
Log             = hello.log  
Queue  
~
```

Job submission #4

- Descriptions File

- Log

- ✓ 로그 저장할 곳을 지정

- Queues

- ✓ Job을 Queue에 넣음. 몇 번 수행할지 지정. 지정하지 않으면 한개의 Job 수행

```
#####  
#Submit description file for hello program#  
#####  
Executable      = hello  
Universe        = standard  
Output          = hello.out  
Log             = hello.log  
Queue  
~
```

예제 (1) #1

Getting Started:

1. Choose a **universe** for the job
2. Make the job **batch-ready**
 - includes making the input data available and accessible
3. Create a **submit description file**
4. Run **condor_submit** to put the job(s) in the queue



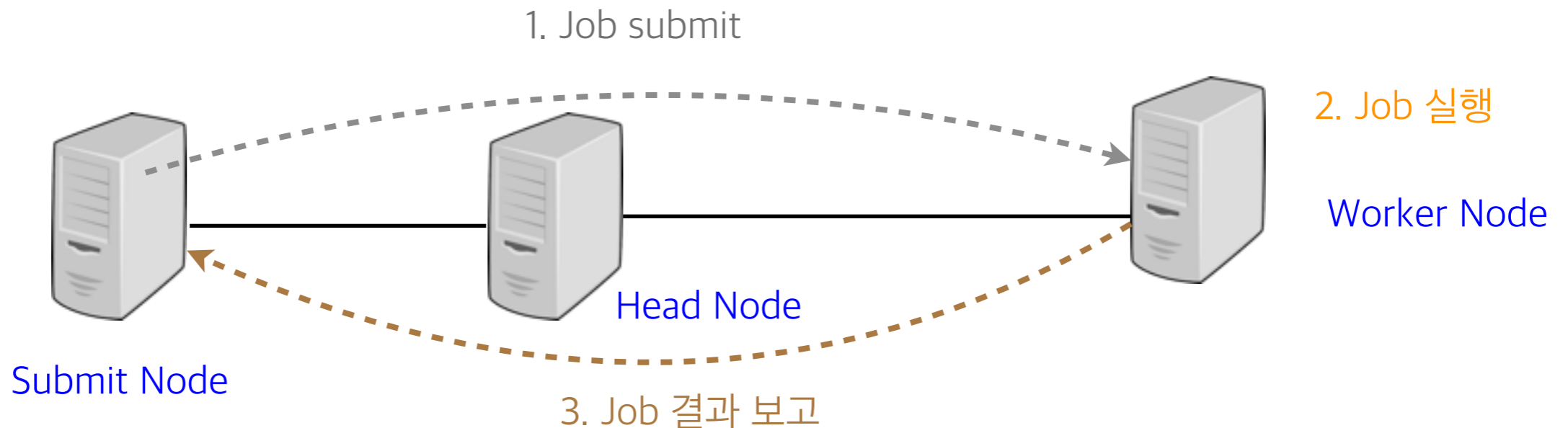
예제 (1) #2

현재 시간을 출력하고, 5초간 슬립 상태였다가 다시 현재의 시간 출력

- #cat out.txt

```
[wookie@school-13-ui ~]$ cat out.txt  
Tue Feb 3 02:59:48 UTC 2015  
Tue Feb 3 02:59:53 UTC 2015  
[wookie@school-13-ui ~]$
```

- 환경구성



예제 (1) #3

현재 시간을 출력하고, 5초간 슬립 상태였다가 다시 현재의 시간 출력

- #mkdir /home/XXX/test1
- #vim /home/XXX/test1/date.sh
- #vim /home/XXX/test1/date.jds

```
#!/bin/bash
```

```
echo `date`
```

```
sleep 5
```

```
echo `date`
```

```
~
```

```
executable = date.sh
```

```
universe = vanilla
```

```
output = out.txt
```

```
error = error.txt
```

```
log = log.txt
```

```
should_transfer_files = YES
```

```
when_to_transfer_output = ON_EXIT
```

```
transfer_input_files = date.sh
```

```
queue
```

예제 (1) #4

현재 시간을 출력하고, 5초간 슬립 상태였다가 다시 현재의 시간 출력

- `#cd /home/XXX/test1`
- `#condor_submit date.jds`

```
[wookie@school-13-ui ~]$ condor_submit date.jds
Submitting job(s).
1 job(s) submitted to cluster 3.
```

- `#condor_q`

```
[wookie@school-13-ui ~]$ condor_q

-- Submitter: school-13-ui.experiment.com : <172.16.13.1:9046> : school-13-ui.experiment.com
ID      OWNER      SUBMITTED  RUN_TIME ST PRI SIZE CMD
  3.0   wookie     2/3 02:59   0+00:00:04 R  0   0.0  date.sh

1 jobs; 0 completed, 0 removed, 0 idle, 1 running, 0 held, 0 suspended
[wookie@school-13-ui ~]$ condor_q
```

예제 (1) #5

현재 시간을 출력하고, 5초간 슬립 상태였다가 다시 현재의 시간 출력

- ls

```
[wookie@fermicloud054 test5]$ ls  
date.jds  date.sh  error.txt  log.txt  out.txt
```

- #cat out.txt

```
[wookie@fermicloud054 test5]$ cat out.txt  
Fri Jan 30 01:06:01 CST 2015  
Fri Jan 30 01:06:06 CST 2015  
[wookie@fermicloud054 test5]$ █
```

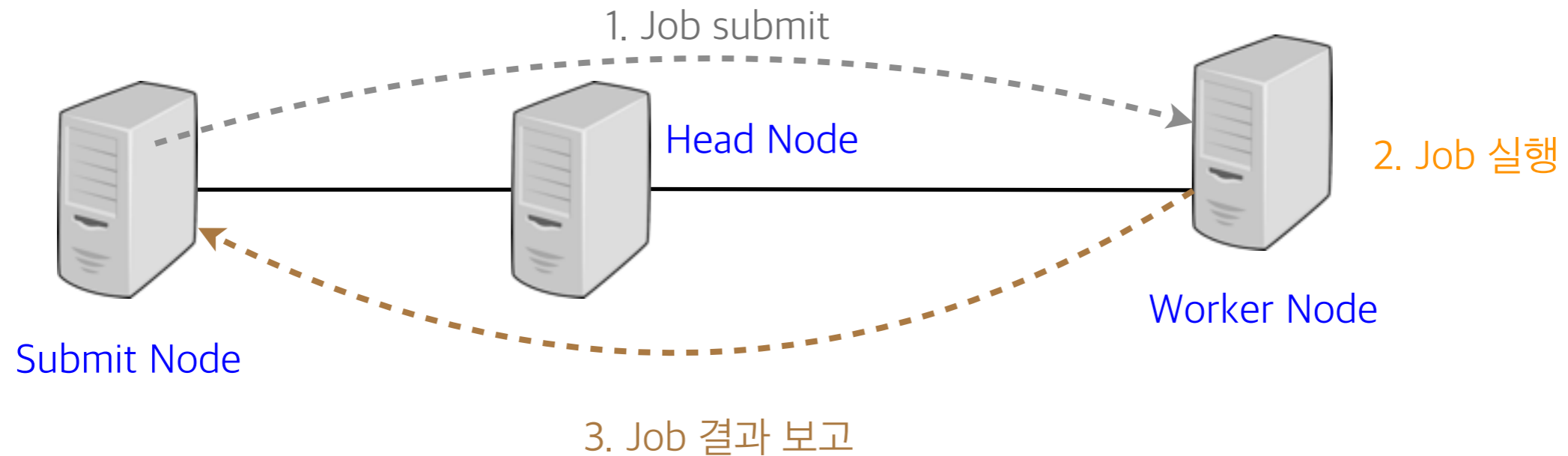
예제 (2) #1

Hello, HTCondor 출력

- **#cat hello.out**

```
[wookie@fermicloud054 ~]$ cat hello.out  
hello, HTCondor
```

- 환경구성



예제 (2) #2

Hello, HTCondor 출력

- **#condor_status**

```
[root@fermicloud054 test1]# condor_status
Name                OpSys      Arch      State      Activity LoadAv Mem      ActvtyTime
fermicloud050.fnal  LINUX     X86_64    Unclaimed Idle       0.000 1877  0+00:00:04
fermicloud051.fnal  LINUX     X86_64    Unclaimed Idle       0.000 1877  0+00:04:37
      Total Owner Claimed Unclaimed Matched Preempting Backfill
      X86_64/LINUX    2     0     0     2     0     0     0     0
      Total          2     0     0     2     0     0     0     0
[root@fermicloud054 test1]#
```

예제 (2) #3

Hello, HTCondor 출력

- `#mkdir /home/XXX/test2`
- `#cd /home/XXX/test2`
- `#vim hello.c`

```
#include <stdio.h>

int main(void)
{
    printf("hello, HTCondor\n");
    return 0;
}
~
~
```

- `#gcc -c hello.c -o hello.o`
- `#condor_compile gcc hello.o -o hello`

예제 (2) #4

Hello, HTCondor 출력

- **#vim submit.hello**

```
#####  
#Submit description file for hello program#  
#####  
Executable      = hello  
Universe        = standard  
Output          = hello.out  
Log             = hello.log  
Queue  
~
```

예제 (2) #5

Hello, HTCondor 출력

- **#condor_status**

```
[wookie@fermicloud054 test1]$ condor_status
Name                OpSys      Arch  State      Activity LoadAv Mem  ActvtyTime
fermicloud050.fnal  LINUX     X86_64 Claimed    Busy      0.000 1877 0+00:00:04
fermicloud051.fnal  LINUX     X86_64 Unclaimed Idle       0.000 1877 0+00:08:04
Total Owner Claimed Unclaimed Matched Preempting Backfill
X86_64/LINUX      2      0      1      1      0      0      0
Total            2      0      1      1      0      0      0
[wookie@fermicloud054 test1]$
```

- **#cat hello.out**

```
[wookie@fermicloud054 test1]$ cat hello.out
hello, HTCondor
```

실습



실습 예제 (1) #1

Argument를 갖는 JoB
(두 연산의 합)

- #vim count.sh

```
#!/bin/bash

NUM_START=$1
NUM_END=$2
LOOP_COUNT=$NUM_START

SUM=0
echo "Sum from $NUM_START to $NUM_END"
while [ $LOOP_COUNT -le $NUM_END ]
do
    SUM=`expr $SUM + $LOOP_COUNT`
    LOOP_COUNT=`expr $LOOP_COUNT + 1`
done
echo "Total Sum = $SUM"
```

- #vim argument.jds

```
executable = count.sh
universe = vanilla
arguments = 1 10
output = out.txt
log = log.txt

should_transfer_files = YES
when_to_transfer_output = ON_EXIT
transfer_input_files = count.sh

queue
```

실습 예제 (1) #2

동일한 실행 프로그램을 여러번 수행 (Single Output)

- #vim count.sh

```
#!/bin/bash

NUM_START=$1
NUM_END=$2
LOOP_COUNT=$NUM_START

SUM=0
echo "Sum from $NUM_START to $NUM_END"
while [ $LOOP_COUNT -le $NUM_END ]
do
    SUM=`expr $SUM + $LOOP_COUNT`
    LOOP_COUNT=`expr $LOOP_COUNT + 1`
done
echo "Total Sum = $SUM"
```

- #vim argument.jds

```
executable = count.sh
universe = vanilla
arguments = 1 10
output = out.txt
log = log.txt

should_transfer_files = YES
when_to_transfer_output = ON_EXIT
transfer_input_files = count.sh
```

```
queue 10
```

```
~
```

```
~
```

실습 예제 (1) #3

동일한 실행 프로그램을 여러번 수행 (Multi Output)

- #vim count.sh

```
#!/bin/bash
NUM_START=$1
NUM_END=$2
LOOP_COUNT=$NUM_START
SUM=0
echo "Sum from $NUM_START to $NUM_END"
while [ $LOOP_COUNT -le $NUM_END ]
do
    SUM=`expr $SUM + $LOOP_COUNT`
    LOOP_COUNT=`expr $LOOP_COUNT + 1`
done
echo "Total Sum = $SUM"
```

- #vim argument.jds

```
executable = count.sh
universe = vanilla
arguments = 1 10
output = out.$(Process).txt
error = error.$(Process).txt
log = log.$(Process).txt

should_transfer_files = YES
when_to_transfer_output = ON_EXIT
transfer_input_files = count.sh

queue 10
```


Reference

[1] **HTCondor 소개 및 HTC@PLSI 구축 사례 , KISTI 박주원**

[2] **<https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallCondor>**

[3] **Condor by Example, Douglas Than**

[4] **Installation Guide for CMS Tier 3 Farm, KISTI**



감사합니다.