

KIAF User Guide

v0.3_en

GSDC

August 9, 2016

Contents

1 KIAF Basics	1
1.1 User Account Subscription	1
1.1.1 User Account Subscription Form	1
1.2 How to Connect to KIAF	2
1.3 Loading The ALICE Environment	2
2 Dataset	4
2.1 Datasets Request	4
2.2 Datasets List	4
3 HTCondor Guide	5
3.1 Condor Cluster Status	5
3.2 Condor Job Queue Status	5
3.3 Condor Job Removal	6
4 Simulation Tutorial(To be completed)	7
5 Analysis Tutorial	8
5.1 Example File	8
5.1.1 Ex 1. Batch_test	8
5.1.2 Ex 2. Multi_output	9
5.2 Application of User's Analysis Macro	10
5.2.1 Macro 1. Batch_test	10
5.2.2 Macro 2. Multi_output	11
6 Additional Option	12
6.1 Alien Token	12
6.1.1 Certification Copy	12
6.1.2 Issue Alien Token	12
6.1.3 Issues Alien Token to Condor Worker Node	12
6.2 PROOF Guide	13
6.2.1 PROOF-Lite Guide	13
6.2.2 PoD Guide	13
Reference	16
A GSDC User Account Subscription Form	17
B KIAF Architecture	19

List of Figures

1	GSDC User Account Subscription Form	1
2	KIAF Architecture	19

1 KIAF Basics

In this section, we introduce how to connect to KIAF. First of all you are required to have a KIAF account.

1.1 User Account Subscription

In order to use the KIAF, you need to create KIAF account. Please refer to the following sections.

1.1.1 User Account Subscription Form

Fill the user account subscription form at below figure 1, original size is available at page 18.

And send the filled form to the following address : kiaf-support@kisti.re.kr

(User subscription form will be immediately discarded after account creation for privacy protection.)

GSDC User Account Subscription Form			
User's Information			
Account Name		VO (Research Group)	
First Name (Native)		Last Name (Native)	
First Name (English)		Last Name (English)	
Position		Country	
Organization (Full Name)	Native		
	English		
Email		Phone	
Supervisor's Information			
First Name (Native)		Last Name (Native)	
First Name (English)		Last Name (English)	
Organization		Position	
Email		Phone	
* Please, acquire Supervisor hand-written signature.			
Subscriber's Signature : _____ (date : . . .)			
Supervisor's Signature : _____ (date : . . .)			
GSDC Director's Signature : _____ (date : . . .)			

Figure 1: GSDC User Account Subscription Form

Account Name should be a user ID, and be written in legible writing¹. After account creation, the administrator will send a confirmation via email with temporary password.

¹For accuracy, please specify your preferred Account Name in email when you send the subscription form to the KIAF administrator.

1.2 How to Connect to KIAF

Once you obtain an account with temporary password, you are ready to connect to KIAF(kiaf.sdfarm.kr) via SSH. Note that SSH for KIAF is 4280.

```
ssh -p 4280 kong91@kiaf.sdfarm.kr
```

Since you are the first time to connect to KIAF, you will be asked to change the password. The new password has to satisfy the password policy. **Once you completed password change, you will be lost the connection.(Don't be panic)** Just try again to connect to KIAF with the new password then you will be succeeded. Your home directory is `/alice/home/KIAF.ID`.

1.3 Loading The ALICE Environment

You can user `kiafenv` command to load the ALICE environment for KIAF.

```
[kong91@kiaf-test-01 ~]$ kiafenv
Load the ALICE environment
```

In `$HOME/.alice_env/alice_env.conf`, the version of AliRoot or AliPhysics to be ready by `kiafenv` command is defined. (You do not mix AliRoot and AliPhysics variables)

```
[kong91@kiaf-test-01 ~]$ cat $HOME/.alice_env/alice_env.conf
export ALIPHYSICS_VERSION="vAN-20160417-1"
```

If you want to use AliRoot, then follow the example below.

```
[kong91@kiaf-test-01 ~]$ cat $HOME/.alice_env/alice_env.conf
export ALICE_ROOT_VERSION="v5-07-20-4"
```

The available version of AliRoot and AliPhysics in KIAF can check at <http://alimonitor.cern.ch/packages>. **Please check the CVMFS status of version is Available.** If that is unavailable, AliRoot is not working and makes error.

`kiafenv` command has options `-t` or `--token` and `-p` or `--pod`. The `t` option creates AliEn session token, the `p` option loads the PoD envitonmet. The option `-h` or `--help` displays help. **All options can be used at same time.** Refer to section 6.1 and 6.2 for more details about AliEn token and PoD.

```
[kong91@kiaf-test-01 ~]$ kiafenv -h
This command load the ALICE environment for KIAF
options:
-p, --pod      : Load the PoD environment
```

```
-t, --token : Create alien session token  
-h, --help  : Display help
```

2 Dataset

In this section, we introduce how to request and check the dataset.

2.1 Dateset Request

In KIAF, datasets are copied to local disk from the Grid. Therefore you need to request dateset to be staged for the analysis. In order to request, send the dateset information to KIAF administrator via email.(kiaf-data-request@kisti.re.kr) Please use this email only for dateset request, Any other inquiries should be sent to kiaf-support@kisti.re.kr Dateset has following info, Data or Sim, Period, Run Number, Variant, Pass, etc. The location of dataset after staging is under `/xrootfs/alice`.

(Ex) - `/xrootdfs/alice/alice/sim/2013/LHC13d3/195675/AOD159/0001/root_archive.zip`

2.2 Dateset List

The date set stored in KIAF is located under the `/xrootdfs/alice`. You can browse datasets directly. The list of dataset is stored as a file, `type.period.run.pass.variant.txt` in `/pool/datalist`.

(Ex) - `sim.LHC13d3.195675.AOD159.txt`

`showdataset` command shows the status of dataset staging. That command can use with a value that be a keyword in the dataset list. The command without the value, shows all dataset list perhaps it needs to take amount of time.

```
[kong91@kiaf-test-01 ~]$ showdatalist LHC12i
```

Dataset	Queued	Downloading	Success	Fail
data/2012/LHC12i/000192762/ESDs/muon_calor_pass2	0	0	220	0
data/2012/LHC12i/000192765/ESDs/muon_calor_pass2	0	0	480	0
data/2012/LHC12i/000192772/ESDs/muon_calor_pass2	0	0	1920	0

3 HTCondor Guide

In this section, we introduce how to use HTCondor.

3.1 Condor Cluster Status

`condor_status` command shows the status of condor slots.

```
[kong91@kiaf-test-01 ~]$ condor_status
```

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@kiaf-test-02	LINUX	X86_64	Unclaimed	Idle	0.110	129013	18+11:37:38
slot1@kiaf-test-03	LINUX	X86_64	Unclaimed	Idle	0.080	129013	18+11:37:29
slot1@kiaf-test-04	LINUX	X86_64	Unclaimed	Idle	0.070	129013	18+11:37:24
slot1@kiaf-test-05	LINUX	X86_64	Unclaimed	Idle	0.090	129013	18+11:37:27
Machines Owner Claimed Unclaimed Matched Preempting							
X86_64/LINUX		4	0	0	4	0	0
Total		4	0	0	4	0	0

3.2 Condor Job Queue Status

`condor_q` command shows the all queue in farm. If job status changes to I or H after run, then check the job log and correct the code. This command shows not only job status but also job ID, job OWNER, etc.

```
[kong91@kiaf-test-01 multi_output]$ condor_q
```

-- Schedd: kiaf-test-01.sdfarm.kr : <134.75.125.139:14337?...>								
ID	OWNER	SUBMITTED		RUN_TIME		ST	PRI	SIZE CMD
9618.0	kong91	6/13	16:58	0+00:00:08	R	0	0.3	condor_dagman -p 0
9619.0	kong91	6/13	16:59	0+00:00:00	I	0	0.0	job_wrap.sh AliPhy
9620.0	kong91	6/13	16:59	0+00:00:00	I	0	0.0	job_wrap.sh AliPhy
9621.0	kong91	6/13	16:59	0+00:00:00	I	0	0.0	job_wrap.sh AliPhy
9622.0	kong91	6/13	16:59	0+00:00:00	I	0	0.0	job_wrap.sh AliPhy
9623.0	kong91	6/13	16:59	0+00:00:00	I	0	0.0	job_wrap.sh AliPhy
6 jobs; 0 completed, 0 removed, 5 idle, 1 running, 0 held, 0 suspended								

3.3 Condor Job Removal

If job stopped cause of some reason, this command can manually remove job. Command is `condor_rm ID`.

```
[kong91@kiaf-test-01 final_test]$ condor_rm 9629  
All jobs in cluster 9629 have been marked for removal
```

4 Simulation Tutorial(To be completed)

This section has not yet been written.

5 Analysis Tutorial

The KIAF uses HTCondor as resource management system. In this section, we introduce how to perform analysis with user analysis macro.

5.1 Example File

5.1.1 Ex 1. Batch_test

An example is located under /tutorial. You can copy `batch_test` directory or copy `batch_test.tar` and extract it.

```
cp -r /tutorial/batch_test $HOME/batch_test
```

or

```
cp -r /tutorial/batch_test.tar $HOME/batch_test.tar
tar -xvf batch_test.tar
```

The list of files in `batch_test` is following.

```
[kong91@kiaf-test-01 batch_test]$ ls -l
total 28
-rw-r--r--. 1 kong91 kong91  776 Jun  7 10:06 AddMyTask.C
-rw-r--r--. 1 kong91 kong91 2036 Jun  7 10:06 AliAnalysisTaskMyTask.cxx
-rw-r--r--. 1 kong91 kong91  895 Jun  7 10:06 AliAnalysisTaskMyTask.h
-rw-r--r--. 1 kong91 kong91 1120 Jun  7 10:06 runAnalysis.C
-rwxr--r--. 1 kong91 kong91 3546 Jun  7 12:56 runCondor.sh
-rw-rw-r--. 1 kong91 kong91 5325 Jun  7 10:06 sim_LHC13d3_195675_AOD159.txt
```

You can see analysis macro, condor run macro and dataset list in the list. Now just run `runCondor.sh` for testing. The example will successfully work without any errors.

Note that, `runCondor.sh` macro splits input files to small sized chunks and run in parallel. In other words, if `runAnalysis.C` macro has errors, `runCondor.sh` macro also will have error. Please make sure that `runAnalysis.C` works well locally first.

Run `runCondor.sh`, then you can see the result as following.

```
[kong91@kiaf-test-01 batch_test]$ ./runCondor.sh
Renaming rescue DAGs newer than number 0
```

```

File for submitting this DAG to Condor : 160612173845/macro/condor.dag.condor.sub
Log of DAGMan debugging messages      : 160612173845/macro/condor.dag.dagman.out
Log of Condor library output          : 160612173845/macro/condor.dag.lib.out
Log of Condor library error messages  : 160612173845/macro/condor.dag.lib.err
Log of the life of condor_dagman itself : 160612173845/macro/condor.dag.dagman.log

Submitting job(s).
1 job(s) submitted to cluster 9244.
-----

```

The working directory, 160612173845, is the timestamp of which you start the job. The history of each job will be saved in `logs` directory under the working directory. The final result file will be saved as `merge_AnalysisResults.root` in `batch_test`.

Working job queue can be checked by `condor_q` command.

```

[kong91@kiaf-test-01 batch_test]$ condor_q

-- Schedd: kiaf-test-01.sdfarm.kr : <134.75.125.139:14337?...

ID      OWNER      SUBMITTED      RUN_TIME ST PRI  SIZE CMD
9317.0   kong91          6/13 00:15     0+00:00:08 R  0   0.3  condor_dagman -p 0
9318.0   kong91          6/13 00:15     0+00:00:01 R  0   0.0  job_wrapper.sh Ali
9319.0   kong91          6/13 00:15     0+00:00:01 R  0   0.0  job_wrapper.sh Ali
9320.0   kong91          6/13 00:15     0+00:00:01 R  0   0.0  job_wrapper.sh Ali
9321.0   kong91          6/13 00:15     0+00:00:01 R  0   0.0  job_wrapper.sh Ali
9322.0   kong91          6/13 00:15     0+00:00:01 R  0   0.0  job_wrapper.sh Ali

```

5.1.2 Ex 2. Multi_output

This example is for the analysis having multi output.

AN example file is located `/tutorial`. You can copy `multi_output` directory or copy `multi_output.tar` and extract it. The structure of macro is similar to Ex 1. Run `runCondor.sh`, then you can see two output files as following.

```

[kong91@kiaf-test-01 multi_output]$ ls -ls *root
26 -rw-r--r--. 1 kong91 kong91 4265 Jun 13 16:34 merge0_AnalysisResults.root
26 -rw-r--r--. 1 kong91 kong91 4265 Jun 13 16:34 merge1_AnalysisResults.root

```

5.2 Application of User's Analysis Macro

In this section we introduce how to make run script with user analysis macro based on the tutorial. Please read the previous examples before proceeding.

5.2.1 Macro 1. Batch_test

Please make sure that your macro runs properly. And after files(.C, .cxx and .h) are located at the same location. prepare the user macro finished test. We need the run macro that is `runCondor.sh` from example 1 at 5.1.1.

The first part of `runCondor.sh` is following.

```
#!/bin/bash
condor_slot=40
nfile=10

work_dir='date +%y%m%d%H%M%S'
run_macro="runAnalysis.C"
data_file="sim_LHC13d3_195675_AOD159.txt"
input_files="AddMyTask.C,AliAnalysisTaskMyTask.cxx,AliAnalysisTaskMyTask.h"
out_file="AnalysisResults.root"
```

- **condor_slot** Maximum number of running slots at same time. Currently KIAF has 160 slots. To facilitate the use of others, please keep the value less than 40.
- **nfile** The number of running data files in each slot. It works similar to `SetSplitMaxInputFileNumber` for Grid job submission. You can refer to the number of files by running the follow command `cat data_file | wc -l`.
- **work_dir** This is working directory name, which you can set as you want.
- **run_macro** This value should be the name of working macro with aliroot.
- **data_file** This is the name of dataset list. The list of dataset is located at `/pool/datalist`, copy the list from the location or create your own. The dataset list should be located in the submission directory with `runCondor.sh`. If not, you need to set an absolute path of dataset location.
- **input_files** Include all files required during analysis such as C, cxx, h, etc. You may need to write in absolute path, if it is not collocated with `runCondor.sh`. Mutiple files can be separated by comma(,).
- **out_file** The name of output of the analysis. It is saved in same directory with `runCondor.sh`.

5.2.2 Macro 2. Multi.output

This macro is for multiple outputs as analysis. Please prepare like Macro 1. You need to modify **runCondor.sh** from example 2 at 5.1.2.

The first part of runCondor.sh is following.

```
condor_slot=40
nfile=10

work_dir='date +%y%m%d%H%M%S'
run_macro="runAnalysis.C"
data_file="sim_LHC13d3_195675_AOD159.txt"
input_files="AddMyTask.C, AliAnalysisTaskMyTask.cxx, AliAnalysisTaskMyTask.h"
out_files="AnalysisResults.root, AnalysisResults.root"
```

Just **out_files** is added to Macro 1. If you write output files name separated by comma(,), the output files are saved in same directory with **runCondor.sh** named **merge#-<out_files>**.²

²#is merge number, it is increase from 0 as writing order.

6 Additional Option

6.1 Alien Token

6.1.1 Certification Copy

Grid User certificates, as `usercert.pem`, `userkey.pem`, should be copied to `$HOME/.globus`. For coping, using the follow command `scp`. This command help to copy the files from local machine to KIAF UI.

```
kong91:~/> scp -r -P 4280 $HOME/.globus kong91@kiaf.sdfarm.kr:/alice/home/kong91/.
kong91@kiaf.sdfarm.kr's password:
usercert.pem                100% 3334      3.3KB/s   00:00
userkey.pem                 100% 1907      1.9KB/s   00:00
```

6.1.2 Issue Alien Token

`alien-token-init` command issues Alien token. If your KIAF ID is not same as CERN ID, `alien-token-init` command needs a value of your CERN ID like `alien-token-init <CERN_ID>`

And added one line, `export alien_API_USER=CERN_ID`, to `$HOME/.alice_env/alice.env.conf` for issue the token by `kiafenv -t`.

6.1.3 Issues Alien Token to Condor Worker Node

The case when you need the token during analysis, condor worker node also need the token. In this case, you have to use `kiafenv -t` command instead of `alien-token-init` command. This command needs a special setting, please connect to administrator before use that command(just once at first time).

6.2 PROOF Guide

6.2.1 PROOF-Lite Guide

PROOF-Lite works by AliRoot command without special configuration. (PROOF-Lite use all CPU in KIAF UI, not a all farm.) Start AliRoot and open the proof session.

```
[kong91@kiaf-test-01 ~]$ aliroot -l
root [0] TProof::Open("")
+++ Starting PROOF-Lite with 40 workers +++
Opening connections to workers: OK (40 workers)
Setting up worker servers: OK (40 workers)
PROOF set to parallel mode (40 workers)
(class TProof*)0x2d20420
root [1]
```

PROOF-Lite uses all cpu in KIAF UI, therefore 40 workers ready for PROOF-Lite.

6.2.2 PoD Guide

PoD is application that set a PROOF environment as user requirement. You can use all resources in KIAF Farm with PoD.

PoD configuration

`kiafenv -p` command loads the PoD configuration. Last two lines in kernel display at first time. (If you run `kiafenv` more than twice, you can see the warning during start AliRoot.)

```
[kong91@kiaf-test-01 ~]$ kiafenv -p
Load the ALICE environment
Set the AliPhysics::vAN-20160417-1 environment
Load PoD environment
Generating a default PoD configuration file...
Generating a default PoD configuration file - DONE.
```

After running the command, `.PoD` directory is created on your home directory. `PoD.cfg` in `.PoD` directory sets a detail information of PROOF. After modify `PoD.cfg`, need to load again the PoD environment. Then use `source /pool/PoD/3.16/PoD_env.sh` command.

Start PoD

For using PoD, need some progress after setting. First, start the PoD server with `pod-server start` command.


```
[kong91@kiaf-test-01 ~]$ pod-server start
Starting PoD server...
updating xproofd configuration file...
starting xproofd...
starting PoD agent...
preparing PoD worker package...
select user defined environment script to be added to worker package...
selecting pre-compiled bins to be added to worker package...
PoD worker package: /share/kong91/.PoD/wrk/PoDWorker.sh
-----
XPROOFD [2860349] port: 21002
PoD agent [2860374] port: 22002
PROOF connection string: kong91@kiaf-test-01.sdfarm.kr:21002
-----
```

After start PoD server, load the PROOF worker node as you want. This time we use the `pod-submit -r condor -n # command`.

```
[kong91@kiaf-test-01 ~]$ pod-submit -r condor -n 10
Job ID: 17606
```

`-r` option sets the resource management system. PoD supported condor, ge, glite, loadleveler, lsf, panda, pbs, slurm. But KIAF supports just condor, therefore you can't change this value. `-n` option desired number of PROOF workers. In KIAF maximum value is 160. And PFOOF worker nodes are running by condor therefore each node assigned with condor job ID.

`pod-info -n` shows the number of prepared worker node. If this number is not matched with your desired number than wait a little moment for prepared worker node by condor.

```
[kong91@kiaf-test-01 ~]$ pod-info -n
10
```

`condor_q`, the command of condor, also shows the worker status. Condor status R means can work pod worker.

```
[kong91@kiaf-test-01 ~]$ condor_q

-- Schedd: kiaf-test-01.sdfarm.kr : <134.75.125.139:12198?...

```

ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD
17606.0	kong91	6/23 13:37	0+00:10:12	R	0	2.0	PoDWorker.sh
17606.1	kong91	6/23 13:37	0+00:10:12	R	0	2.0	PoDWorker.sh
17606.2	kong91	6/23 13:37	0+00:10:12	R	0	2.0	PoDWorker.sh
17606.3	kong91	6/23 13:37	0+00:10:12	R	0	2.0	PoDWorker.sh
17606.4	kong91	6/23 13:37	0+00:10:12	R	0	2.0	PoDWorker.sh
17606.5	kong91	6/23 13:37	0+00:10:12	R	0	2.0	PoDWorker.sh
17606.6	kong91	6/23 13:37	0+00:10:12	R	0	2.0	PoDWorker.sh
17606.7	kong91	6/23 13:37	0+00:10:12	R	0	2.0	PoDWorker.sh
17606.8	kong91	6/23 13:37	0+00:10:12	R	0	2.0	PoDWorker.sh
17606.9	kong91	6/23 13:37	0+00:10:12	R	0	2.0	PoDWorker.sh

Finished all progress, connect to PoD server in Aliroot.

```
[kong91@kiaf-test-01 ~]$ aliroot -l
root [0] TProof::Open("pod://","masteronly")
Starting master: opening connection ...
Starting master: OK
Note: File "iostream" already loaded
PROOF set to sequential mode
(class TProof*)0x31e6f80
root [1]
```

Reference

- [1] Introduction to HTCondor (2016 GSDC School of Data Grid Computing). https://indico.cern.ch/event/476290/contributions/1154545/attachments/1221990/1786975/HTCondor_....pdf.
- [2] PROOF tutorial. <https://root.cern.ch/root/html/doc/tutorials/proof/index.html>.
- [3] PoD User's Manual. <http://pod.gsi.de/doc/3.16/PoD.pdf>.

A GSDC User Account Subscription Form

Print the form on next page and fill below informations.

1. User's Information

- **Account Name** The ID is issued to the user.
We recommend write two or more to make preparations for already using case.
(Attempts to issue in accordance with the written order.)
- **VO (Research Group)** The experimental group
ALICE is the experimental group that uses the KIAF.
- **Position** The user's status
(Master students, PhD students, post-doc, researchers, etc.)
- **Country** The countries where the Institute is located
- **Organization** Your home institution
- **Organization Unit** The department of institute (like dept. Physics)
- **Email** The contact available Email
- **Phone** The contact available phone number

2. Supervisor's Information

- **Organization** Supervisor's home institution
- **Position** The title of your supervisor (professor, etc.)
- **Email** The contact available Email
- **Phone** The contact available phone number

Do your signature and get a signature from your supervisor. After scan the form, please sending to us via email.(kiaf-support@kisti.re.kr)

GSDC User Account Subscription Form

User's Information

Account Name		VO (Research Group)	
First Name (Native)		Last Name (Native)	
First Name (English)		Last Name (English)	
Position		Country	
Organization (Full Name)	Native		
	English		
Email		Phone	

Supervisor's Information

First Name (Native)		Last Name (Native)	
First Name (English)		Last Name (English)	
Organization		Position	
Email		Phone	

* Please, acquire Supervisor hand-written signature.

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

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

GSDC Director's Signature : _____ (date : . . .)

B KIAF Architecture

1. Computing resource

- **Server** 5 Machines
- **CPU** Each server 20 cores, total 100 physical cores
(200 cores with Hyper-Threading)
- **RAM** Each server 128GB
- **Storage** NAS 200TB
- **Network** 10Gbps

2. Software

- **OS** Scientific Linux 6
- **HTCondor** v8.4.7
- **PoD** v3.16
- **XRootD** v4.3.0

3. **Condor construction** UI / CE - 1 machine, WN - 4 machines

4. **User authorization** LDAP

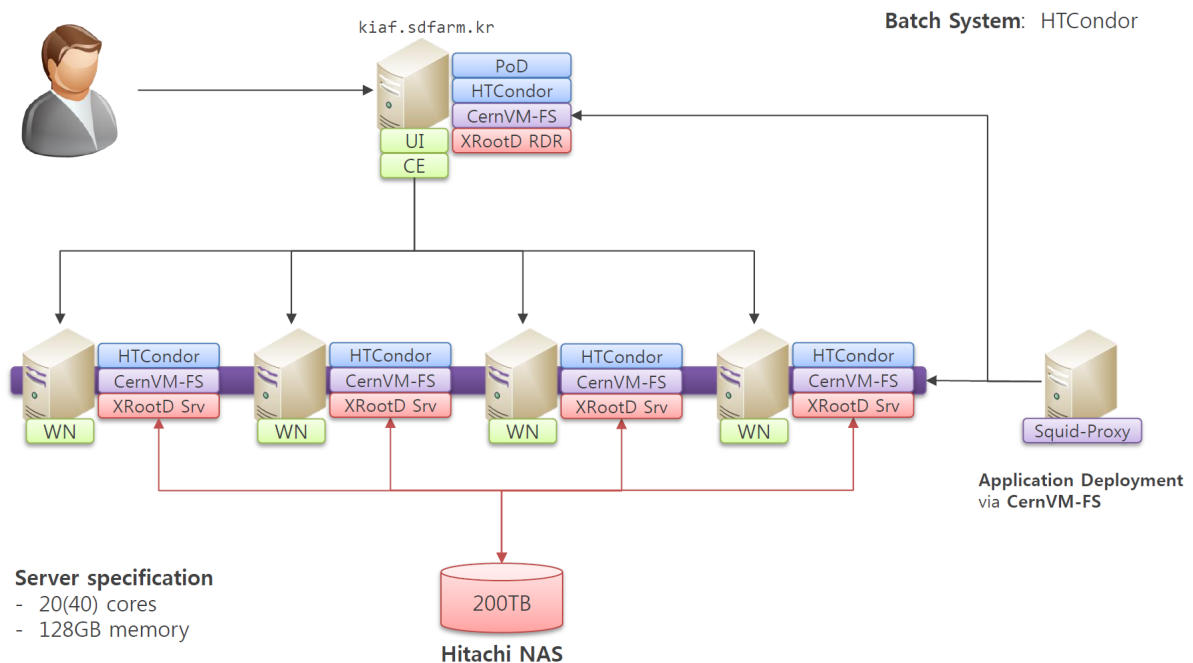


Figure 2: KIAF Architecture