Introduction

In this page I will show you how to make use of IPAS Condor pool.

This means you can learn how to do the following actions

- Job submission: condor_submit
- Job listing: condor_q
- Job removal: condor_rm

Job submission

Before submitting you should know how to describe it.

It means you must write a JDL(Job Description Language).

Below I will show you several examples how to do it.

Helloworld One

This is a minimal JDL you can use to submit jobs.

- Helloworld1.jdl

```
universe   = vanilla
executable = ./Helloworld.sh
queue 1
```

Description

- **universe = vanilla** means this is a local submission.
- **executable = ./Helloworld.sh** means which file is your initial script.
- **queue 1** means you submit one job.

This is the initial script

- Helloworld.sh

```
#!/bin/sh

date
hostname
pwd
printenv | sort
exit 0
```

JDL and initial script are ready, you can start to submit the job

- Submit this job

```
\> condor_submit Helloworld1.jdl
Submitting job(s).
1 job(s) submitted to cluster 1193.
```

As I say this is a minimal JDL. No submission log, standard output and error.

Condor will launch your job in the same directory you submit this job.

If you create some files in your script you will see them in the local directory.

Below is the standard output of Helloworld.sh

```
Tue Dec 19 12:04:33 CST 2006
hep022
/home/tsan/HelloWorld.test
PWD=/home/tsan/HelloWorld.test
SHLVL=1
_=/usr/bin/printenv
_CONDOR_ANCESTOR_15248=15250:1166501073:3696742912
_CONDOR_ANCESTOR_23205=15248:1166501072:1507036043
_CONDOR_ANCESTOR_2522=23205:1162878898:2770834733
_CONDOR_SCRATCH_DIR=/condor_local/execute/dir_15248
```

Helloworld Two

It's better you ask condor to log when you submit.

Also ask condor to log standard output and standard error.

- Helloworld2.jdl

```
universe                = vanilla
executable              = ./Helloworld.sh

transfer_output         = true
transfer_error          = true

log                     = $(Cluster).condor.log
output                  = $(Cluster).$(Process).stdout
error                   = $(Cluster).$(Process).stderr

queue 1
```

Description

- **transfer_output = true** and **transfer_error = true** tell Condor transfer

standard output and error.

- **log = XXX** tells Condor log job submission into this file.
- **output = XXX** and **error = XXX** tell Condor log standard output and error into these two files.
- **$(Cluster)** is the Condor job id
- **$(Process)** is the section id of this job, in this case it's **0**

You can use this JDL to submit

- Submit this job

```
\> condor_submit Helloworld2.jdl
Submitting job(s).
1 job(s) submitted to cluster 1198.
```

Under the submission directory you can see these three logs: 1198.condor.log, 1198.0.stdout and 1198.0.stderr

It's the same as **Helloworld One** your jobs start in the same directory you submit.

- 1198.0.stdout

```
Tue Dec 19 12:08:52 CST 2006
hep022
/home/tsan/HelloWorld.test
PWD=/home/tsan/HelloWorld.test
SHLVL=1
_=/usr/bin/printenv
_CONDOR_ANCESTOR_15265=15266:1166501332:3455885056
_CONDOR_ANCESTOR_23205=15265:1166501332:1507036045
_CONDOR_ANCESTOR_2522=23205:1162878898:2770834733
_CONDOR_SCRATCH_DIR=/condor_local/execute/dir_15265
```

Helloworld Three

Execute your jobs in the same directory you submit is not appropriate. It's not safe, too.

In fact it's useful when you want to see up-to-date outputs.

For production I do not suggest this way.

You should write your outputs to local disk in worker nodes.

- Helloworld3.jdl

```
universe                = vanilla
executable              = ./HelloWorld.sh

transfer_output         = true
transfer_error          = true
```

```
transfer_executable             = true
should_transfer_files           = true

when_to_transfer_output         = ON_EXIT

log                             = $(Cluster).condor.log
output                           = $(Cluster).$(Process).stdout
error                           = $(Cluster).$(Process).stderr

queue 1
```

Description

- **transfer_executable = true** tells Codnor transfer initial script.
- **should_transfer_files = true** tells Condor transfer files into Condor Execute directory.
- **when_to_transfer_output = ON_EXIT** tells Condor transfer files back when done.

You can use this JDL to submit

- Submit this job

```
\> condor_submit Helloworld3.jdl
Submitting job(s).
1 job(s) submitted to cluster 1199.
```

The same as **Helloworld Two** you can see these three logs: 1199.condor.log, 1199.0.stdout and 1199.0.stderr

The difference is your working directory changes to Condor scratch directory.

- 1199.0.stdout

```
Tue Dec 19 13:53:57 CST 2006
hep037
/condor_local/execute/dir_23687
PWD=/condor_local/execute/dir_23687
SHLVL=1
_=/usr/bin/printenv
_CONDOR_ANCESTOR_14941=23687:1166507636:3264454673
_CONDOR_ANCESTOR_23687=23689:1166507637:3466335488
_CONDOR_ANCESTOR_2500=14941:1162878900:2350999085
_CONDOR_SCRATCH_DIR=/condor_local/execute/dir_23687
```

One more thing need to know, Condor will copy all files back to submission directory you create. But this does not include folders, just **files**

Add-on

Number of jobs

If you want to submit jobs more than one. You can do in this way

- JDL

```
...

queue 3
```

It means Condor will create three jobs.

In **Helloworld Two** and **Helloworld Three**, $(Process) will be from 0 to 2

Script arguments

If you want to pass arguments to your initial script, you can do in this way

- JDL

```
arguments = Argv1 $(Cluster) Argv2 $(Process)
```

Thus your initial script will get **Argv1 1220 Argv2 0** as its arguments if Condor job id is **1220** and section id is **0**

If the number of jobs are 3, these three jobs' arguments will be

- **Argv1 1220 Argv2 0**
- **Argv1 1220 Argv2 1**
- **Argv1 1220 Argv2 2**

Get env vars into JDL

You can obtain env vars in JDL. It's convenient to submit jobs if you use loop.

Here I take **arguments** as an example. You can use it anywhere in JDL

- JDL

```
arguments = Argv1 $(Cluster) $ENV(SEED) $(Process)
```

Your initial script will get **Argv1 $1220 31 0** if env var, SEED, is equal to **31**

Set env vars in JDL

You can ask Condor to set up some env vars before yours jobs start.

- JDL

```
environment = C=3;aa=$ENV(aa);P=$(Process)
```

Separate them by **;**

I show you all possible ways:

- Set C to be **3**
- Set aa according to env var aa
- Set P from $(Process)

Turn to test mode

As I show in **Helloworld Three**, your working directory changes to Condor scratch directory.

If you want to test your jobs, see the up-to-date outputs you can change the value of should_transfer_files

- JDL

```
should_transfer_files = if_needed
```

BE CAREFUL. YOUR WORKING DIRECTORY IS THE SAME AS YOU SUBMIT JOBS

Mail notification

If you wan to receive mails when your jobs are done or have problems, add these into JDL

- JDL

```
notification            = ALWAYS
notify_user             = tsan@fnal.gov
```

Otherwise set **notification** to **NEVER**.

Inputs and outputs

In the above examples I do not mention that how to transfer inputs and outputs explicitly.

By default Condor will not copy files into Condor scratch directory, except initial script.

By default Condor will copy all files(not include folders) back to submission directory.

If you want to transfer files in you want, assign them in JDL

- JDL

```
transfer_input_files     = ./Helloworld.dat,./seed.dat
```

Separate them by **,**

If you want transfer files secure, assign them in JDL

- JDL

```
dont_encrypt_input_files = ./seed.dat
encrypt_input_files      = ./Helloworld.dat
```

Make sure Condor can read these files. It means the path to these files is reachable under submission directory.

These files will be transfered into Condor scratch directory, no directory structure.

If you want to transfer files back, assign them in JDL

- JDL

```
transfer_output_files = my.$(Cluster).$(Process).log,my.$(Cluster).$(Process).dat
```

Separate them by **,**

If you want transfer files secure, assign them in JDL

- JDL

```
dont_encrypt_output_files      = my.$(Cluster).$(Process).log
encrypt_output_files           = my.$(Cluster).$(Process).dat
```

Make sure Condor can read these files under Condor scratch directory.

It means the path to these files is reachable in worker nodes.

Hint: you can tar your inputs and outputs into one file respectively.

Job listing

Once you succeed to submit jobs, you can issue condor_q to list your jobs

```
\> condor_q
-- Submitter: ipas004.phys.sinica.edu.tw : <140.109.102.24:32797> : ipas004.phys.sinica.edu.tw
 ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
 535.0   tsan          12/12 01:02   7+15:43:52 R  0   625.3 XXX
...
1154.0   tsan          12/19 08:43   0+08:03:27 R  0   158.1 XXX
...
```

More specific you can do this to list your jobs only.

```
\> condor_q tsan
-- Submitter: ipas004.phys.sinica.edu.tw : <140.109.102.24:32797> : ipas004.phys.sinica.edu.tw
 ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
 535.0   tsan          12/12 01:02   7+15:43:52 R  0   625.3 XXX
...
```

You can do this to get more details of your jobs.

```
\> condor_q -l 535.0
-- Submitter: ipas004.phys.sinica.edu.tw : <140.109.102.24:32797> : ipas004.phys.sinica.edu.tw
MyType = "Job"
TargetType = "Machine"
ClusterId = 535
...
```

Job removal

If you think the job you submitted is wrong or the job seems go wrong.

You can remove it by

```
\> condor_rm 535.0
```

Or you can remove all of your jobs

```
\> condor_rm tsan
```

Use it carefully.

Suggestion

Here I want to show you the suggested JDL you use in IPAS Condor.

This is to let users use IPAS Condor more effective.

There are some principles, please keep them in mind

- Do not store data in your home directory, it's not storage area.
- Do not read/write online from/to home directory, also in storage area.

Thus

- For inputs: let Condor to transfer your inputs to worker nodes if possible.
- For outputs: let Condor to transfer your outputs from worker nodes if possible.
- When your jobs are running, they should write their outputs to local disk, not storage area, not home directory.
- When your jobs are done, you can copy them to storage area in worker nodes.

Job behavior

You should let your jobs write their outputs to anywhere you want.

It means you can let it write to Condor scratch directory.

So you will not write online to storage area.

If one of your inputs is large, impossible to transfer into worker nodes.

It's fine you can read it online, but not write online.

JDL for production

Here is a template you can use.

- production.jdl

```
universe                = vanilla
executable              = ./Helloworld.sh

transfer_output         = true
transfer_error          = true
transfer_executable     = true
should_transfer_files   = true

when_to_transfer_output = ON_EXIT

log                     = $(Cluster).condor.log
output                  = $(Cluster).$(Process).stdout
error                   = $(Cluster).$(Process).stderr

notification            = NEVER

queue 1
```

Add other setting if you need it

JDL for test mode

You can use the production.jdl as you JDL templae.

Change **should_transfer_files** to **if_needed**

You can see up-to-date outputs.

WARNING: The working directory will be the same as you submit

IpasCC CondorHandBook (上次是 TsanLung 在 2006-12-20 10:31:15 編輯的)