Build your own cluster: Session 4

Farrukh Khan
Computational Science and AI Directorate (CSAID)
Fermi National Accelerator Laboratory
May 21, 2024

Recap

- We learned a little about system management
- We learned how operating system is essentially the core of any system
- We had a little hands-on exercise with Alma Linux 9
- We saw how different systems work together to form a facility
- We looked at compute systems, or more specifically HTCondor
- We learned how to install and configure HTCondor to create a little cluster of our own

HTCondor Software Suite (HTCSS) – Hands-on exercise

- Install HTCondor inside containers
- Go to /etc/condor/config.d/ and create a 99-local.conf file
- Inside that configuration file:
  - set DAEMON_LIST to include the relevant daemons
  - set COLLECTOR_HOST to point to the central manager
  - set ALLOW_READ, ALLOW_WRITE and ALLOW_DAEMON knobs to set authorization level
    - These are generally set as <identity>/<hostname>. For ease of exercise, you can set them as wild cards */*
- Configure appropriate authentication mechanism
- Start condor_master

HTCondor Software Suite (HTCSS) – Hands-on exercise

- Install HTCondor:

```
[root@fkhan interactive]# cd /packages/
[root@fkhan packages]# dnf install -y ./condor-23.0.10-1.el9.x86_64.rpm
Last metadata expiration check: 4:21:31 ago on Thu May 16 17:46:41 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture            Version                 Repository
================================================================================
Installing:
```

- Go to /etc/condor/config.d/ and create a 99-local.conf file

```
[root@fkhan packages]# cd /etc/condor/config.d/
[root@fkhan config.d]# touch 99-local.conf
[root@fkhan config.d]# ls
00-htcondor-9.0.config  10-stash-plugin.conf  99-local.conf
```

HTCondor Software Suite (HTCSS) – Hands-on exercise

- Configure DAEMON_LIST, COLLECTOR_HOST, ALLOW_WRITE, ALLOW_READ and ALLOW_DAEMON
  - Central Manager:

```
[root@central config.d]# cat 99-local.conf
DAEMON_LIST = COLLECTOR, NEGOTIATOR
COLLECTOR_HOST = central
ALLOW_READ = */*
ALLOW_WRITE = */*
ALLOW_DAEMON = */*
```

HTCondor Software Suite (HTCSS) – Hands-on exercise

- Configure DAEMON_LIST, COLLECTOR_HOST, ALLOW_WRITE, ALLOW_READ and ALLOW_DAEMON
  - Access point:

```
[root@access config.d]# cat 99-local.conf
DAEMON_LIST = SCHEDD
COLLECTOR_HOST = central

ALLOW_READ = */*
ALLOW_WRITE = */*
ALLOW_DAEMON = */*
```

  - Execution point:

```
[root@exec config.d]# cat 99-local.conf
DAEMON_LIST = STARTD
COLLECTOR_HOST = central

ALLOW_READ = */*
ALLOW_WRITE = */*
ALLOW_DAEMON = */*
```

HTCondor Software Suite (HTCSS) – Hands-on exercise

- Configure appropriate authentication mechanism: THE EASY (DEPRECATED) WAY
- HTCondor comes with directives to do the heavy lifting for you. Modify the 00-htcondor-9.0.config and switch from recommended_v9_0 security to host_based security as shown below:

```
[root@access config.d]# cat 00-htcondor-9.0.config
##
## Default security settings
##
## Host-based security was the default for the 8.8 series (and earlier).
##
## Host-based security assumes that all users on a machine are trusted.
## For example, if host-based security trusts that a given machine can
```

```
##

use security:host_based
# use security:recommended_v9_0
```

HTCondor Software Suite (HTCSS) – Hands-on exercise

- Configure appropriate authentication mechanism: THE HARD (PRACTICAL) WAY
- Make sure that the security is set to recommended_v9_0 security in the 00-htcondor-9.0.config file:

```
# use security:host_based
use security:recommended_v9_0
```

HTCondor Software Suite (HTCSS) – Hands-on exercise

- Central Manager:
  - In both authentication cases, you can just go ahead and start condor_master on the central manager

```
[root@central config.d]# condor_master
[root@central config.d]# ps faux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          31  0.0  0.0   5100  4236 pts/1    Ss   21:06   0:00 /bin/bash
root         397  0.0  0.0   7560  3320 pts/1    R+   22:39   0:00  \_ ps faux
root           1  0.0  0.0   4996  3976 pts/0    Ss+  21:06   0:00 /bin/bash
condor       360  0.0  0.0  24672 13432 ?        Ss   22:39   0:00 condor_master
root         391  0.0  0.0   8096  3164 ?        S    22:39   0:00  \_ condor_procd -A /var/run/condor/procd_pipe -L /var/log
condor       392  0.5  0.0  19948 14540 ?        Ss   22:39   0:00  \_ condor_shared_port
condor       393  1.0  0.0  20704 15388 ?        Ss   22:39   0:00  \_ condor_collector
condor       394  0.5  0.0  21288 15912 ?        Ss   22:39   0:00  \_ condor_negotiator
```

HTCondor Software Suite (HTCSS) – Hands-on exercise

- Access point and Execution point:
  - In both authentication cases, you can start condor_master
  - However, there is an additional step involved in case of IDTokens
  - At daemon start up, both condor_schedd and condor_startd will attempt to get a IDTokens from the central manager
  - These requests have to be manually approved using condor_token_request_list and condor_token_request_approve
  - You can check the respective log file under /var/log/condor to confirm that the daemon is indeed making a request
  - If the daemon is not making a request on its own, you can create the tokens manually on the central manager and place them under /etc/condor/tokens.d
    - I'll show you how to do the setup by creating a token manually

HTCondor Software Suite (HTCSS) – Hands-on exercise

- On the central manager create IDTokens for access points and execution points with the following authorizations:

```
[root@central config.d]# condor_token_create -authz ADVERTISE_MASTER -authz ADVERTISE_SCHEDD -authz READ
-authz WRITE -authz DAEMON -identity access
eyJhbGciOiJIUzI1NiIsImtpZCI6IlBPT0wifQ.eyJpYXQiOjE3MTU5MDE0NjMsImlzcyI6ImNlbnRyYWwiLCJqdGkiOiJhMzAxZDlhMj
U1YTU2YWFmNTdhM2Q2MjE4MzU5NDEyMiIsInNjb3BlIjoiY29uZG9yOlwvQURWRVJUSVNFX01BU1RFUiBjb25kb3I6XC9BRFZFUlRJU0V
fU0NIRUREIGNvbmRvcjpcL1JFQUQgY29uZG9yOlwvV1JJVEUgY29uZG9yOlwvREFFTU9OIiwic3ViIjoiYWNjZXNzIn0.Lp8pm_-k0ZQK
jHOJ_IUxDFyWJhmiioABZCp_ObRMP2U
[root@central config.d]# condor_token_create -authz ADVERTISE_MASTER -authz ADVERTISE_STARTD -authz READ
-authz WRITE -authz DAEMON -identity exec
eyJhbGciOiJIUzI1NiIsImtpZCI6IlBPT0wifQ.eyJpYXQiOjE3MTU5MDE1MTIsImlzcyI6ImNlbnRyYWwiLCJqdGkiOiIxZDIzZDVmNm
VlODBmZDRmMzYzMDg5MjEzYjViOWFjYyIsInNjb3BlIjoiY29uZG9yOlwvQURWRVJUSVNFX01BU1RFUiBjb25kb3I6XC9BRFZFUlRJU0V
fU1RBUlREIGNvbmRvcjpcL1JFQUQgY29uZG9yOlwvV1JJVEUgY29uZG9yOlwvREFFTU9OIiwic3ViIjoiZXhlYyJ9.QP3aSZmfZtiDwBv
p0h4sLMQqQ0KOXxsd9qGRTk5GbW0
```

HTCondor Software Suite (HTCSS) – Hands-on exercise

- Now create a file under /etc/condor/tokens.d on the access point and the execution point with the content set to the hash generated by the central manager. Access point as an example:

```
[root@access tokens.d]# ls /etc/condor/tokens.d/
token
[root@access tokens.d]# condor_token_list
Header: {"alg":"HS256","kid":"POOL"} Payload: {"iat":1715901463,"iss":"central","jti":"a301d9a255a56aaf57
a3d62183594122","scope":"condor:\/ADVERTISE_MASTER condor:\/ADVERTISE_SCHEDD condor:\/READ condor:\/WRITE
 condor:\/DAEMON","sub":"access"} File: /etc/condor/tokens.d/token
```

- The name doesn't matter. The identity matters for traceability and for finer grained control (recall the ALLOW_* configuration params!)
- After creating the token files, run condor_reconfig for the HTCondor to pick up the new IDToken
- The verification commands as shown in session 3 slides should work now

HTCondor Software Suite (HTCSS) – Submitting a job

- With great power comes great responsibility!
- Now that you have root access, you have the potential to mimic any user on the system. All system managers have this capability but doing so without explicit user permission is unethical
- Now let's submit a job on your access point. You should feel to mimic access, exec or central user for your job submission. The example below mimics central:

```
[root@access tokens.d]# sudo -u central /bin/bash
sudo: unable to send audit message: Operation not permitted
bash-5.1$ cd /home/interactive/
bash-5.1$ mkdir job
```

HTCondor Software Suite (HTCSS) – Submitting a job

- A simple sleep script and a simple sleep job descriptive file:

```
bash-5.1$ cd job/
bash-5.1$ cat sleep.sh
#!/bin/bash

sleep 600
bash-5.1$ cat sleep.jdl
Universe   = vanilla
Executable = sleep.sh
Output     = sleep.out.$(Cluster).$(Process)
Error      = sleep.err.$(Cluster).$(Process)
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
RequestCpus = 1
RequestMemory = 1024
Queue 3
```

HTCondor Software Suite (HTCSS) – Submitting a job

- Job submission and queue overview:

```
bash-5.1$ condor_submit sleep.jdl
Submitting job(s)...
3 job(s) submitted to cluster 2.
bash-5.1$ condor_q -nobatch


-- Schedd: access : <10.89.0.42:9618?... @ 05/16/24 23:33:50
 ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
   1.0   central         5/16 23:30   0+00:00:00 I  0    0.0 sleep.sh
   1.1   central         5/16 23:30   0+00:00:00 I  0    0.0 sleep.sh
   1.2   central         5/16 23:30   0+00:00:00 I  0    0.0 sleep.sh
   2.0   central         5/16 23:33   0+00:00:00 I  0    0.0 sleep.sh
   2.1   central         5/16 23:33   0+00:00:00 I  0    0.0 sleep.sh
   2.2   central         5/16 23:33   0+00:00:00 I  0    0.0 sleep.sh

Total for query: 6 jobs; 0 completed, 0 removed, 6 idle, 0 running, 0 held, 0 suspended
Total for central: 6 jobs; 0 completed, 0 removed, 6 idle, 0 running, 0 held, 0 suspended
Total for all users: 6 jobs; 0 completed, 0 removed, 6 idle, 0 running, 0 held, 0 suspended
```

HTCondor Software Suite (HTCSS) – Submitting a job

- Running jobs!

```
bash-5.1$ condor_q -nobatch -run


-- Schedd: access : <10.89.0.42:9618?... @ 05/16/24 23:42:57
 ID      OWNER            SUBMITTED     RUN_TIME HOST(S)
   1.0   central         5/16 23:30   0+00:00:22 slot1_1@exec
   1.1   central         5/16 23:30   0+00:00:22 slot1_2@exec
   1.2   central         5/16 23:30   0+00:00:22 slot1_3@exec
   2.0   central         5/16 23:33   0+00:00:22 slot1_4@exec
   2.1   central         5/16 23:33   0+00:00:22 slot1_5@exec
   2.2   central         5/16 23:33   0+00:00:22 slot1_6@exec
```

HTCondor Software Suite (HTCSS) – Optional exercise

- OPTIONAL live exercise if time permits (or to do by yourself if you're curious to learn more):
  - PROBLEM: your user pool has three different users. User A belongs to the physics department, user B belongs to chemistry department and user C belongs to mathematics. Department of physics pays extra money to get preferential access to the resources. Department of mathematics does not pay anything but contributes manpower to facility operations in hopes of retaining access to the pool. Department of chemistry has negotiated an opportunistic access to the pool
  - TASK: you, as a pool administrator, are being asked to come up with an accounting policy in your pool that gives preferential access to department of physics users, followed by department of mathematics and finally department of chemistry
  - CONCERN: the user from department of physics submits bad jobs from time to time. They generally ask for a single core and end up spawning 8 threads inside their jobs. Since they get preferential access to your pool, you want to ensure this doesn't mean broken or down worker nodes

HTCondor Software Suite (HTCSS) – Optional exercise

- CENTRAL MANAGER:
  - Set up accounting groups for each department
  - Come up with appropriate priorities or quotas for the three departments
- ACCESS POINT:
  - Make sure appropriate user jobs are tagged for their correct accounting groups
  - Make sure a user cannot modify their accounting group (i.e. the classAd should be immutable)
- EXECUTION POINT:
  - Due to CPU load concerns, hide 2 CPUs and 2 GB of RAM from users to allow your system to function adequately

HTCondor Software Suite (HTCSS) – Conclusion

- That's all! This brings our hands-on exercises to a close
- I hope you were able to get something useful out of these sessions and experience what it feels like to setup systems and get them to work!