

# An introduction to

# HTCondor

## High Throughput Computing

---

---

WRES 3109

Principle of Distributed Systems

# Acknowledgement

- The content of this presentation is adapted from the tutorial and documentation obtained from Condor developer web site @ University of Wisconsin–Madison

<http://research.cs.wisc.edu/htcondor/>

# The Condor Project

- established in 1985
- research and development of distributed high throughput computing



Miron Livny



2012



1988

# Miron's quotes

- the problems we are facing are fundamental and **silver bullet technologies** are pipe-dreams, for computing in general and distributed computing in particular
- Bringing the right data to the processing unit and shipping the results back to the consumer is a fundamental (hard) problem.

I give it all

# Condor is

- High throughput computing, instead of high performance computing
  - less concerned about instantaneous computing power; but the amount of computing that can be harnessed over a month or a year
- all about **match making**
  - bring together the one who requests for computing power with those who offer them
  - “*Condor - a hunter of idle workstations.*” [1988]

# Definitions

- Job
  - the Condor representation of a piece of work
  - Condor's quanta of work
  - Like a Unix process, and can be an element of a workflow
- ClassAd
  - Condor's internal data representation
- Machine or Resource
  - computers that can do the processing
- Pool
  - A collection of computers used by Condor

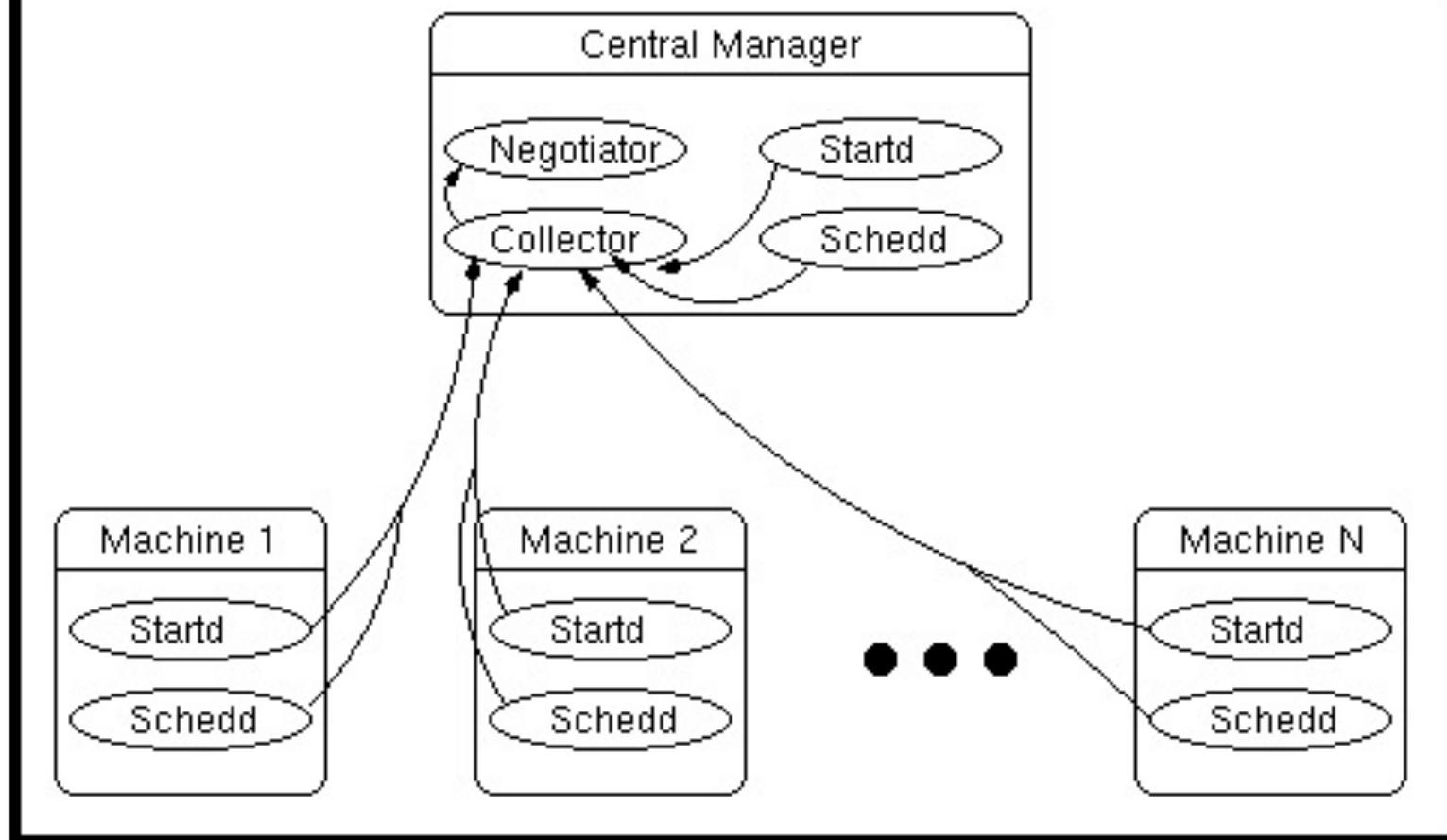
*what you have  
what you want*

# More definitions

- Match Making
  - Associating a job with a machine resource
- Central Manager / Condor Master
  - Central repository for the whole pool
  - Does match making
- Submit Host / Scheduler
  - The computer from which jobs are submitted to Condor
- Execute Host / Executor
  - The computer that runs a job

# Condor daemons

Architecture of a Condor Pool  
(With no jobs running)



Launch 3 instance in your AWS

- ↳ small (1 cpu) submission host      3 machines
- ↳ medium (condor master) (manager)
- ↳ large (executor) (1 cpu) (execution host)

1. aws academy

2. Launch dashboard

3. Module

4. Learner lab

5. Start lab

6. EC2 dashboard

↳ launch instance

1. Condor Manager

- Ubuntu 20.04

- t2. small

- key pair harithu

create security group

Condor\_Pool

## HTCondor

1. Download HTCondor

2. 10.0.0.0

Installation guide

3. Click on linux

4. Ambil command

→ Condor central manager

Launch

2. Submission-host

- Ubuntu 20.04

- t2.small

- same key pair

- same security group

Condor-Pool

3. Execution-Host

- Ubuntu 20.04

- t2.large

- same key pair

- select existing

security group

EC2 instance connect

→ connect

Upkan semua instances

htcondor.readthedocs.io

delete sudo dalam command

↳ tular nama m → CondorManager

↳ password → "password"

Check is condor running

sudo systemctl status condor

cd /etc

cd /hosts

Condor Master



private IPv4  
address

172.31

semua node kena add tiga  $\infty$  hosts

# condor\_status

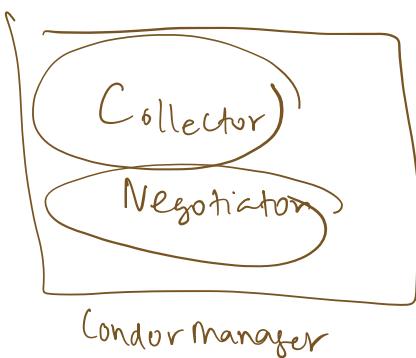
# condor\_query

!! Nama central manager kena sama

1. Central Manager

2 - Execution

3 - Submission



★ Charge security

All TCP From  
condor  
port  
save

Check diorang using

sudo systemctl status condor

U

condor-q → submission\_host

condor-status → executor\_host

cat /etc/condor/config.d/01-executor-

config

manu /etc/condor

/var/log/condor

- condor\_master:
  - This program runs constantly and ensures that all other parts of Condor are running. If they hang or crash, it restarts them.
- condor\_collector:
  - This program is part of the Condor central manager. It collects information about all computers in the pool as well as which users want to run jobs. It is what normally responds to the condor\_status command.
- condor\_negotiator:
  - This program is part of the Condor central manager. It decides what jobs should be run where.
- condor\_startd:
  - If this program is running, it allows jobs to be started up on this computer—that is, your computer is an "execute machine". This advertises your computer to the central manager (more on that later; but in this case it's also your computer) so that it knows about this computer. It will start up the jobs that run.
- condor\_schedd
  - If this program is running, it allows jobs to be submitted from this computer—that is, your computer is a "submit machine". This will advertise jobs to the central manager so that it knows about them. It will contact a condor\_startd on other execute machines for each job that needs to be started.
- condor\_shadow (Not shown above)
  - For each job that has been submitted from this computer, there is one condor\_shadow running. It will watch over the job as it runs remotely. In some cases it will provide some assistance. You may or may not see any condor\_shadow processes running, depending on what is happening on the computer when you try it out.

# Installation

- [follow [http://research.cs.wisc.edu/htcondor/tutorials/hands\\_on\\_2008/](http://research.cs.wisc.edu/htcondor/tutorials/hands_on_2008/)]
- Option 1:
  - Download HTCondor 7.8.8 tarball @ <http://research.cs.wisc.edu/htcondor/downloads/>
  - use condor\_install pearl script to install Condor
- Option 2: (for ubuntu)
  - Use HTCondor Debian Repository @ <http://research.cs.wisc.edu/htcondor/debian/>
  - in su mode: `echo "deb http://research.cs.wisc.edu/htcondor/debian/stable/ squeeze contrib" >> /etc/apt/sources.list`
  - `sudo apt-get update`
  - `sudo apt-get install condor`

# Start Condor

- If install from scratch
  - source condor.sh // set environment variable
  - condor\_master // start condor\_master
- If using apt-get
  - sudo /etc/init.d/condor start
- Check the status using
  - condor\_status

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@ip-10-135-46	LINUX	X86_64	Unclaimed	Idle	0.000	3725	0+00:04:43
slot2@ip-10-135-46	LINUX	X86_64	Unclaimed	Idle	0.000	3725	0+00:05:05
		Total	Owner	Claimed Unclaimed Matched Preempting Backfill			
		X86_64/LINUX	2	0 0	2 0	0 0	0 0
		Total	2	0 0	2 0	0 0	0 0

# Stop Condor

- Just in case you want to know
- If install from scratch
  - `condor_off -master`
- If using apt-get
  - `sudo /etc/init.d/condor stop`

```
ubuntu@ip-10-135-46-41:~$ condor_status
Error: communication error
CEDAR:6001:Failed to connect to <10.135.46.41:9618>
```

# Creating your first job

```
ubuntu@ip-10-135-46-41:~/condor$ cat > simple.c
#include <stdio.h>

main(int argc, char **argv)
{
    int sleep_time;
    int input;
    int failure;

    if (argc != 3) {
        printf("Usage: simple <sleep-time> <integer>\n");
        failure = 1;
    } else {
        sleep_time = atoi(argv[1]);
        input      = atoi(argv[2]);

        printf("Thinking really hard for %d seconds...\n", sleep_time);
        sleep(sleep_time);
        printf("We calculated: %d\n", input * 2);
        failure = 0;
    }
    return failure;
}
```

# Compile and test your program

```
ubuntu@ip-10-135-46-41:~/condor$ gcc -o simple simple.c
```

```
ubuntu@ip-10-135-46-41:~/condor$ ./simple 1 2
```

Thinking really hard for 1 seconds...

We calculated: 4

# Create a job description file

```
ubuntu@ip-10-135-46-41:~/condor$ cat > submit  
Universe      = vanilla  
Executable    = simple  
Arguments     = 4 10  
Log           = simple.log  
Output         = simple.out  
Error          = simple.error  
should_transfer_files   = YES  
when_to_transfer_output = ON_EXIT  
Queue
```

# Submit your first job

```
ubuntu@ip-10-135-46-41:~/condor$ condor_submit submit  
Submitting job(s).  
1 job(s) submitted to cluster 1.
```

```
ubuntu@ip-10-135-46-41:~/condor$ condor_q
```

```
-- Submitter: ip-10-135-46-41.ap-southeast-1.compute.internal : <10.135.46.41:42992> :  
ip-10-135-46-41.ap-southeast-1.compute.internal  


| ID  | OWNER  | SUBMITTED  | RUN_TIME   | ST | PRI | SIZE | CMD         |
|-----|--------|------------|------------|----|-----|------|-------------|
| 1.0 | ubuntu | 4/17 16:56 | 0+00:00:04 | R  | 0   | 0.0  | simple 4 10 |

  
1 jobs; 0 completed, 0 removed, 0 idle, 1 running, 0 held, 0 suspended
```

```
ubuntu@ip-10-135-46-41:~/condor$ condor_status
```

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime	
slot1@ip-10-135-46	LINUX	X86_64	Unclaimed	Idle	0.460	3725	0+00:00:04	
slot2@ip-10-135-46	LINUX	X86_64	Claimed	Busy	0.000	3725	0+00:00:05	
		Total	Owner	Claimed	Unclaimed	Matched	Preempting	Backfill
	X86_64/LINUX	2	0	1	1	0	0	0

# Check the log file

```
ubuntu@ip-10-135-46-41:~/condor$ more simple.log
000 (001.000.000) 04/17 16:56:30 Job submitted from host: <10.135.46.41:42992>
...
001 (001.000.000) 04/17 16:56:31 Job executing on host: <10.135.46.41:48957>
...
006 (001.000.000) 04/17 16:56:35 Image size of job updated: 10
    0 - MemoryUsage of job (MB)
    0 - ResidentSetSize of job (KB)
...
005 (001.000.000) 04/17 16:56:35 Job terminated.
    (1) Normal termination (return value 0)
        Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
        Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
        Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
        Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
    56 - Run Bytes Sent By Job
    8531 - Run Bytes Received By Job
    56 - Total Bytes Sent By Job
    8531 - Total Bytes Received By Job
Partitionable Resources : Usage Request
    Cpus : 1
    Disk (KB) : 17 17
```

# Parameter sweep

```
ubuntu@ip-10-135-46-41:~/condor$ cat > submit-ps
```

```
Universe = vanilla
```

```
Executable = simple
```

```
Arguments = 4 10
```

```
Log = simple.log
```

```
Output = simple.$(Process).out
```

```
Error = simple.$(Process).error
```

```
should_transfer_files = YES
```

```
when_to_transfer_output = ON_EXIT
```

```
Queue
```

```
Arguments = 4 11
```

```
Queue
```

```
Arguments = 4 12
```

```
Queue
```

```
ubuntu@ip-10-135-46-41:~/condor$ condor_q
```

```
-- Submitter: ip-10-135-46-41.ap-southeast-1.compute.internal :  
<10.135.46.41:42992> : ip-10-135-46-41.ap-southeast-1.compute.internal
```

ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD	
2.0	ubuntu	4/17 16:59	0+00:00:00	I	0	0.0	simple	4 10
2.1	ubuntu	4/17 16:59	0+00:00:00	I	0	0.0	simple	4 11
2.2	ubuntu	4/17 16:59	0+00:00:00	I	0	0.0	simple	4 12

```
3 jobs; 0 completed, 0 removed, 3 idle, 0 running, 0 held, 0 suspended
```

```
ubuntu@ip-10-135-46-41:~/condor$ condor_status
```

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@ip-10-135-46	LINUX	X86_64	Claimed	Busy	0.020	3725	0+00:00:04
slot2@ip-10-135-46	LINUX	X86_64	Claimed	Busy	0.000	3725	0+00:00:05
Total Owner Claimed Unclaimed Matched Preempting Backfill							
	X86_64/LINUX	2	0	2	0	0	0
	Total	2	0	2	0	0	0

# Condor DAGMan

- refer to example located at
  - examples (if install from scratch)
  - `/usr/share/doc/condor/examples/dagman/`  
(apt-get)