Batch Processing

Client Installation (Windows)

1. Launch MSI installer from **\\HTCONDOR-CM\INSTALL**
2. Choose to **Join an existing HTCondor Pool**
   - Hostname of Central Manager: **HTCONDOR-CM**
3. Check **Submit jobs to HTCondor Pool** if you want to submit jobs from this machine
4. Select **Always run jobs and never suspend them** if you want this machine to execute jobs
5. Set Accounting Domain to **keenswh.com**
6. Leave Email Settings and Java Settings blank
7. Set Host Permission Settings in the following way:
   - Host with Read Access: *****
   - Host with Write Access: **$(FULL_HOSTNAME), HTCONDOR-CM.keenswh.com**
   - Host with Administrator Access: **$(FULL_HOSTNAME), HTCONDOR-CM.keenswh.com**
8. Leave VM Universe Settings disabled
9. Set Destination Folder to **C:\condor**
10. Finish the installation but don't restart the computer yet
11. If this machine is going to execute jobs, set content of **C:\condor\condor_config.local** in the following way:
```
use feature : GPUs
GPU_DISCOVERY_EXTRA = -extra
SLOT_TYPE_1 = cpus=100%,GPUs=100%
NUM_SLOTS_TYPE_1 = 1
```
12. Now restart your computer
13. Allow all the firewall exception request for HTCondor services

Client Installation (Linux)

1. Guide was tested on CentOS 7, installed with Base Environment set to GNOME Desktop, including the following Add-ons: GNOME Applications, Development Tools. Should probably work also on RHEL and Fedora.
2. Install RPM package
```
cd /etc/yum.repos.d
sudo wget http://research.cs.wisc.edu/htcondor/yum/repo.d/htcondor-stable-rhel7.repo
cd ~
wget http://research.cs.wisc.edu/htcondor/yum/RPM-GPG-KEY-HTCondor
sudo rpm --import RPM-GPG-KEY-HTCondor
rm RPM-GPG-KEY-HTCondor
sudo yum install condor.x86_64
```
3. Append the following into HTCondor configuration file (`sudo gedit /etc/condor/condor_config`)
```
CONDOR_HOST = htcondor-cm-centos
UID_DOMAIN = keenswh.com
CONDOR_ADMIN =
SMTP_SERVER =
ALLOW_READ = *
ALLOW_WRITE = $(FULL_HOSTNAME), htcondor-cm-centos.keenswh.com
ALLOW_ADMINISTRATOR = $(FULL_HOSTNAME), htcondor-cm-centos.keenswh.com
SEC_DEFAULT_AUTHENTICATION_METHODS = CLAIMTOBE
use POLICY : ALWAYS_RUN_JOBS
WANT_VACATE = FALSE
WANT_SUSPEND = TRUE
DAEMON_LIST = MASTER SCHEDD STARTD
```
4. In case you want to just submit jobs from this machine, remove `STARTD` daemon from the `DAEMON_LIST` above. Similarly, remove `SCHEDD` daemon to just execute jobs.

5. Put the following into HTCondor local configuration file (sudo gedit /etc/condor/condor_config.local)
   ```
   SLOT_TYPE_1 = cpus=100%
   NUM_SLOTS_TYPE_1 = 1
   ```
6. Open all firewall ports for HTCondor daemons and jobs
   ```
   sudo firewall-cmd --permanent --zone=public --add-port=1025-65535/udp
   sudo firewall-cmd --permanent --zone=public --add-port=1025-65535/tcp
   ```
7. Put HTCondor into startup script (`sudo gedit /etc/rc.local`)
   ```
   service condor start
   ```
8. Ensure the startup script gets executed after start
   ```
   sudo chmod +x /etc/rc.d/rc.local
   ```
9. Restart the computer

Machine Utilization

- It is preferred to not enabling detection of machine idle state (client installation step 4.), because our work stations are almost permanently used during the day and switched off during the night.
- Job owner has an option to limit the set of worker machines to be used via enumerating them in the job script. Preferably you should ask the machine owners whether you can use their machine for the job.
- Resources currently consumed by condor jobs can be viewed in *Task Manager - Show processes from all users*. Look for processes with "condor_" prefix in their name.
- Machine owner can temporarily remove his machine from the cluster via `condor_off` and `condor_on` commands.
- If it is inconvenient, machine owner can kill any job currently running on his machine via `condor_vacate -fast` command. Don't be afraid of using this command, the job will be started again on different machine.

Known Issues

- **Avast DeepScreen/CyberCapture** causes sandbox scan of every HTCondor executable during first run. Disable it through Avast GUI in *Settings - General*.
- **VirtualBox Host-only Adapter** causes wrong IP to be reported to Central Manager. Delete it from *Control Panel - Network and Sharing Center - Change adapter settings*.
- If you are unable to restart condor via `condor_restart`, kill the `condor_master` process in the *Task Manager - Show processes from all users* and then start the `condor` service again via *Control Panel - Administrative Tools - Services*.

Workflow Example

1. Checkout https://github.com/KeenSoftwareHouse/BrainSimInternal and open the solution.
2. Familiarize with an interface provided by **Core/Execution/MyProjectRunner.cs**.
3. Familiarize with an example in **CoreRunner/ExampleExperiment.cs** adapt it for your experiment.
4. Copy **\\HTCONDOR-CM\EXAMPLE** to your local file system.
5. Familiarize with an example and comments in **EXAMPLE/MyExampleJob/MyJobExample.condor**, adapt the script for your experiment.
6. Test syntax correctness of your script via dry-running the job without actually submitting it:
   ```
   condor_submit -name HTCONDOR-CM -dry-run DryRun.log MyJobExample.condor
   ```
7. Submit the job to the central manager, all binaries and input files will be copied there as well:
   ```
   condor_submit -name HTCONDOR-CM -spool MyJobExample.condor
   ```
8. Monitor the execution status of your job and notice the ID of your job cluster (let's say it is 38):
   ```
   condor_q -name HTCONDOR-CM
   ```
9. Wait until all processes in your job cluster are completed and then fetch the results from central manager:
   ```
   condor_transfer_data -name HTCONDOR-CM 38
   ```
10. Clean after yourself by telling central manager to delete any information stored there for your job cluster:
    ```
    condor_rm -name HTCONDOR-CM 38
    ```
11. Now you can review the results of your batch job (e.g. in **MyJobExample/Results**).

Useful Commands

- Synchronously wait on the submit machine for a completion of a particular job process (e.g. 38.2) or entire job cluster (e.g. 38). Could be useful for scripting execution of multiple job clusters where input of some clusters depends on the output of others (e.g. genetic algorithms).
  ```
  condor_wait -status 38
  ```

- Temporarily suspend execution of a particular job process (e.g. 38.2) or entire job cluster (e.g. 38). Computation devices on the affected worker machines will stopped to be utilized, process state is kept in memory to be able to continue where it left off.
  ```
  condor_suspend -name HTCONDOR-CM 38
  condor_continue -name HTCONDOR-CM 38
  ```
- Temporarily hold a particular job process (e.g. 38.2) or entire job cluster (e.g. 38) in the job queue on the central manager. Job in a hold state will not be assigned to any worker machine. If the job is already being calculated by some worker machine, it is killed and returned to the queue so it can be restarted later.
  ```
  condor_hold -name HTCONDOR-CM 38
  condor_release -name HTCONDOR-CM 38
  ```
- Kill a particular job process (e.g. 38.2) or entire job cluster (e.g. 38). If the command is not followed by condor_rm to remove the job/cluster from the queue, it will be run from the start on different set of machines. It can take a few minutes before the affected machines vacate the job and become idle.
  ```
  condor_vacate_job -name HTCONDOR-CM -fast 38
  ```
- Kill any job process that is currently being calculated by your local machine. It will be run from the start on different machine.
  ```
  condor_vacate -fast
  ```
- Find out why the requirements of your job are not being matched with any worker machine.
  ```
  condor_q -name HTCONDOR-CM -analyze
  condor_q -name HTCONDOR-CM -analyze:reverse
  condor_q -name HTCONDOR-CM -better-analyze
  condor_q -name HTCONDOR-CM -better-analyze:reverse
  ```
- Start/stop/restart condor services on the local machine. Condor have to be restarted whenever there is any change in local configuration files to reflect those changes (you can also use condor_reconfig, but some settings need complete restart).
  ```
  condor_on
  condor_off
  condor_restart
  ```
- Notify the central manager from your local worker machine about its availability. This is done automatically every 5 minutes by condor services, so this command might be useful only after the central manager machine is freshly started (e.g. after applying updates) and you don't want to wait.
  ```
  condor_reschedule -name HTCONDOR-CM
  ```
- Find out the IP addresses and ports of condor services running on the local machine.
  ```
  condor_who -verbose
  ```
- Find out whether you have read/write access towards central manager (when invoked from worker/submit machine) or whether you have administrator access towards worker machine (when invoked from the central manager).
  ```
  condor_ping -name HTCONDOR-CM -table READ WRITE
  condor_ping -name USER1-PC -table READ WRITE ADMINISTRATOR
  ```
- User priority is dynamically adjusted by condor depending on how many resources you are consuming relative to other users. You can override/reset automatic adjustments if necessary to run the jobs sooner.
  ```
  condor_userprio -resetusage USERNAME
  condor_userprio -setprio USERNAME 500
  ```
- See the current utilization and state of worker machines. Can be invoked from any machine.
  ```
  condor_status
  ```
- See detailed characteristics of available worker machines. Can be invoked from any machine.
  ```
  condor_status -verbose
  ```
- See the recent job history. Can be invoked only from the central manager.
  ```
  condor_history -limit 10
  ```

Server Installation (Windows)

1. Prepare a dedicated (virtual) machine to host the central manager
   - Machine should have enough disk space to store data for scheduled jobs and results from finished jobs
   - Virtual machine shall use bridged network adapter so it is visible by other physical machines on the network
   - Join the machine to the company domain (e.g. **keenswh.com**) and name it appropriately (e.g. **HTCONDOR-CM**)
   - Choose the machine name wisely to avoid possible future naming conflict
   - Joining the domain should not be necessary if worker machines will neither be in the domain
2. Create **INSTALL** and **EXAMPLE** shared directories somewhere in the server file system and set them to be **readable by everyone**

- Put stable MSI installer downloaded from https://research.cs.wisc.edu/htcondor/downloads/ to INSTALL directory
- Put unpacked content of attached EXAMPLE.zip into EXAMPLE directory

3. Launch downloaded MSI installer
4. Choose to **Create a new HTCondor Pool**
   - Name of New Pool shall be different than machine host name (e.g. **GoodAI**)
5. Leave **Submit jobs to HTCondor Pool** unchecked
6. Leave **Do not run jobs on this machine** selected
7. Set Accounting Domain to the domain name of the machine from the first installation step (possibly empty)
8. Leave Email Settings and Java Settings blank
9. Set Host Permission Settings in the following way:
   - Host with Read Access: *****
   - Host with Write Access: *****
   - Host with Administrator Access: **$(FULL_HOSTNAME)**
10. Leave VM Universe Settings disabled
11. Set Destination Folder to **C:\condor**
12. Finish the installation and restart the computer
13. Allow all the firewall exception request for HTCondor services

Server Installation (Linux)

1. Guide was tested on CentOS 7, installed with Base Environment set to GNOME Desktop, including the following Add-ons: GNOME Applications, Development Tools. Should probably work also on RHEL and Fedora.
2. Prepare a dedicated (virtual) machine to host the central manager
   - Machine should have enough disk space to store data for scheduled jobs and results from finished jobs
   - Virtual machine shall use bridged network adapter so it is visible by other physical machines on the network
   - Set machine domain to match the network domain (e.g. **keenswh.com**) and name it appropriately (e.g. **htcondor-cm-centos**)
   - Choose the machine name wisely to avoid possible future naming conflict
3. Install RPM package

```
cd /etc/yum.repos.d
sudo wget http://research.cs.wisc.edu/htcondor/yum/repo.d/htcondor-stable-rhel7.repo
cd ~
wget http://research.cs.wisc.edu/htcondor/yum/RPM-GPG-KEY-HTCondor
sudo rpm --import RPM-GPG-KEY-HTCondor
rm RPM-GPG-KEY-HTCondor
sudo yum install condor.x86_64
```

4. Alternatively, you can also just download the RPM package and install it later offline

```
yum install condor.x86_64 --downloadonly --downloaddir=.
sudo rpm -ivh condor-8.4.8-1.rhel7.i386.rpm
```

5. Append the following into HTCondor configuration file (`sudo gedit /etc/condor/condor_config`)

```
CONDOR_HOST = $(FULL_HOSTNAME)
COLLECTOR_NAME = GoodAI
UID_DOMAIN = keenswh.com
CONDOR_ADMIN =
SMTP_SERVER =
ALLOW_READ = *
ALLOW_WRITE = *
ALLOW_ADMINISTRATOR = $(FULL_HOSTNAME)
SEC_DEFAULT_AUTHENTICATION_METHODS = CLAIMTOBE
START = FALSE
WANT_VACATE = FALSE
WANT_SUSPEND = TRUE
DAEMON_LIST = MASTER SCHEDD COLLECTOR NEGOTIATOR
```

6. Open all firewall ports for HTCondor daemons and jobs

```
sudo firewall-cmd --permanent --zone=public --add-port=1025-65535/udp
sudo firewall-cmd --permanent --zone=public --add-port=1025-65535/tcp
```

7. Put HTCondor into startup script (`sudo gedit /etc/rc.local`)
   `service condor start`
8. Ensure the startup script gets executed after start
   `sudo chmod +x /etc/rc.d/rc.local`
9. Restart the computer

Further Information

- HTCondor Homepage (http://research.cs.wisc.edu/htcondor/)
- HTCondor Tutorial (http://research.cs.wisc.edu/htcondor/tutorials/scotland-admin-tutorial-2003-10-23/)
- HTCondor Manual (http://research.cs.wisc.edu/htcondor/manual/v8.4/index.html)
- HTCondor Commands (http://research.cs.wisc.edu/htcondor/manual/v8.4/11_Command_Reference.html)
- HTCondor FAQ (https://htcondor-wiki.cs.wisc.edu/index.cgi/wiki?p=HowToAdminRecipes)
- HTCondor Wiki (https://htcondor-wiki.cs.wisc.edu/index.cgi/wiki)