Università di Perugia
Dipartimento di Matematica e Informatica

Esame HPC

Esercitazione 2 - HTCondor

Professore
Prof. Osvaldo Gervasi
Dott. Damiano Perri

Studente
Nicolò Vescera

Anno Accademico 2021-2022

Indice

1 Obbiettivo

L'obbiettivo di questa esercitazione è quello di realizzare un cluster ad alte prestazioni, cioè un insieme di nodi interconnessi tra loro capaci di distribuirsi il carico di lavoro in maniera autonoma mediante l'uso e la configurazione del software HTCondor.

2 Ambiente di Lavoro

Questa esercitazione è stata svolta all'interno del seguente ambiente di lavoro:

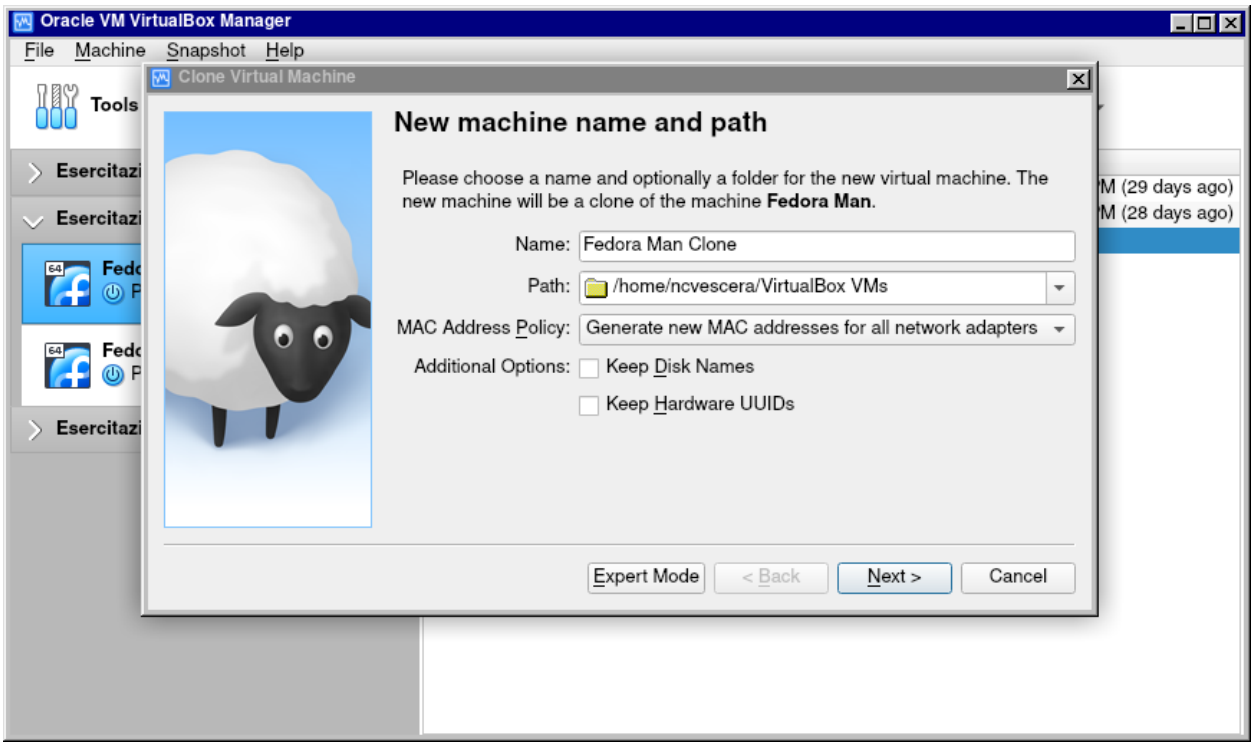
- **Hardware**:
  - **CPU**: AMD Ryzen 9 5900x
  - **RAM**: 32 GB DDR4 @3200 MHz
- **Software**:
  - **Host OS:** Arch Linux
  - **Guest OS**: Ubuntu 20.4 LTS Server
  - **Virtualization Software**: VirtualBox 6.1

3 Configurazione Macchine

Per questa esercitazione sono necessarie 3 macchine virtuali che saranno 1 `master`, il quale avrà il compito di accettare i job e di effettuare lo scheduling di questi e 2 `slave` che avranno il compito di eseguire i vari job assegnategli . Ogni macchina è stata configurata come segue:

- **Cores:** 4 Core
- **RAM:** 4 GB
- **Dischi di archiviazione:**
  - Disco Principale da 10 GB
- **Scheda di Rete:** Scheda di rete con Bridge

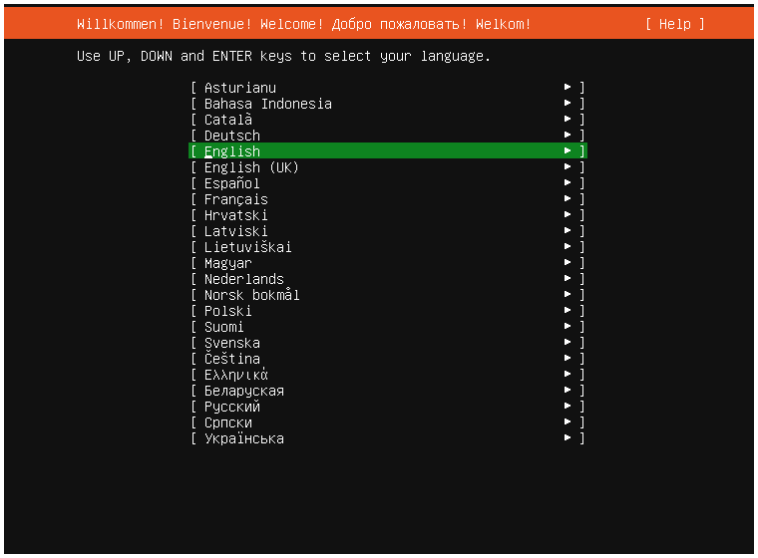
È consigliato configurare una sola macchina, installare e configurare il sistema operativo e i software necessari, per poi utilizzare la funzione 'Clona' di VirtualBox per generare una copia identica senza dover ripetere tali operazioni nuovamente. Durante la fase di clonazione della macchina è necessario selezionare l'opzione "Generare nuovi Mac Address per ogni Network Adapter" come policy per la gestione dei Mac Address.



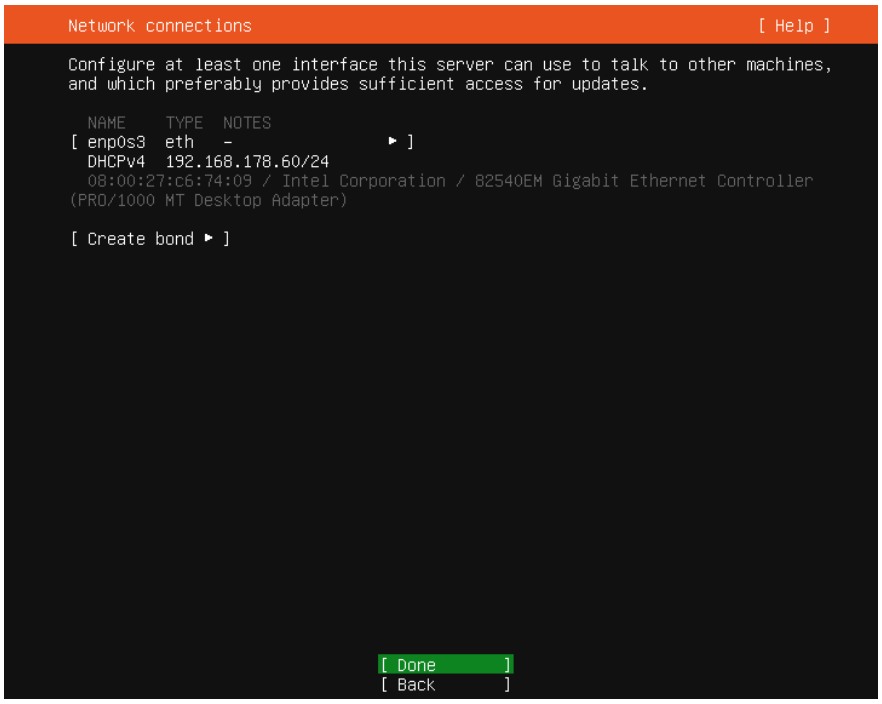
4 Configurazione Software

4.1 Sistema Operativo

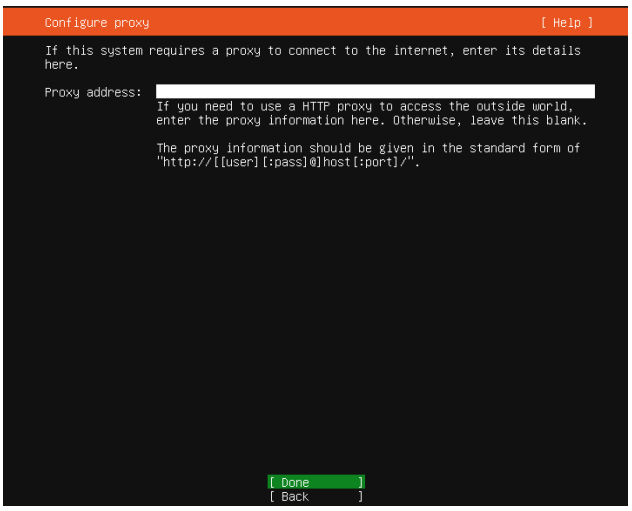
Appena terminata la configurazione della macchina virtuale aggiungo la iso di Ubuntu 20.4 Server e avvio la fase di installazione.
Seleziono la lingua del sistema e il layout della tastiera:

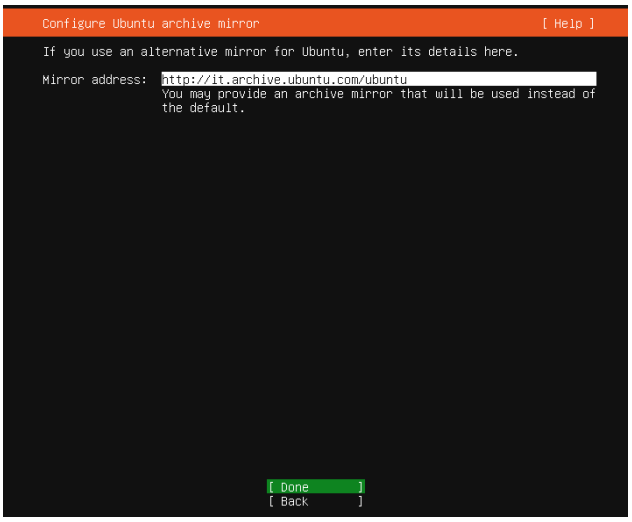


Mi assicuro che alla scheda di rete venga assegnato un IP dal DHCP per poter scaricare il software necessario durante l'installazione:

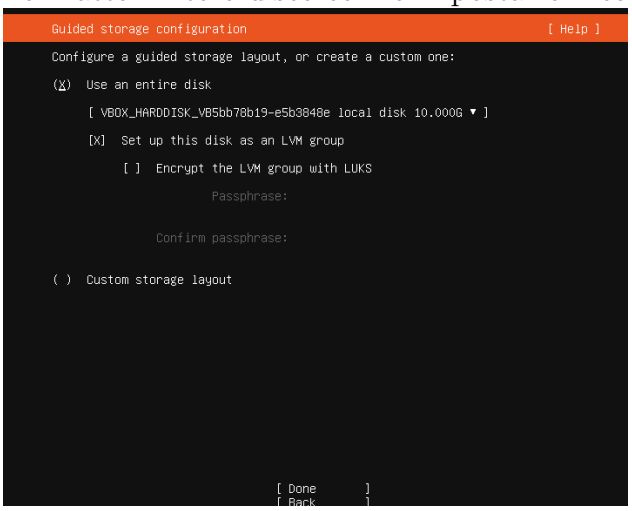


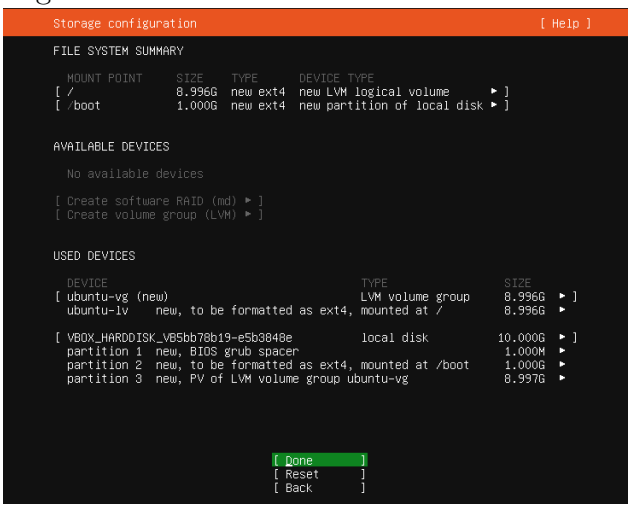
Ignoro la sezione del proxy e scelgo il mirror italiano:



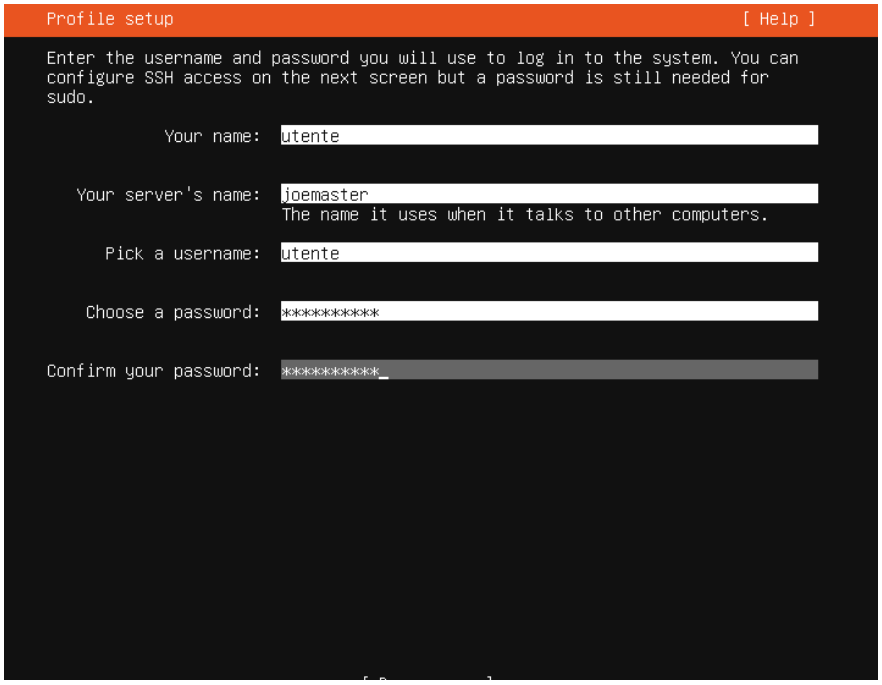


Formatto l'intero disco con le impostazioni consigliate:

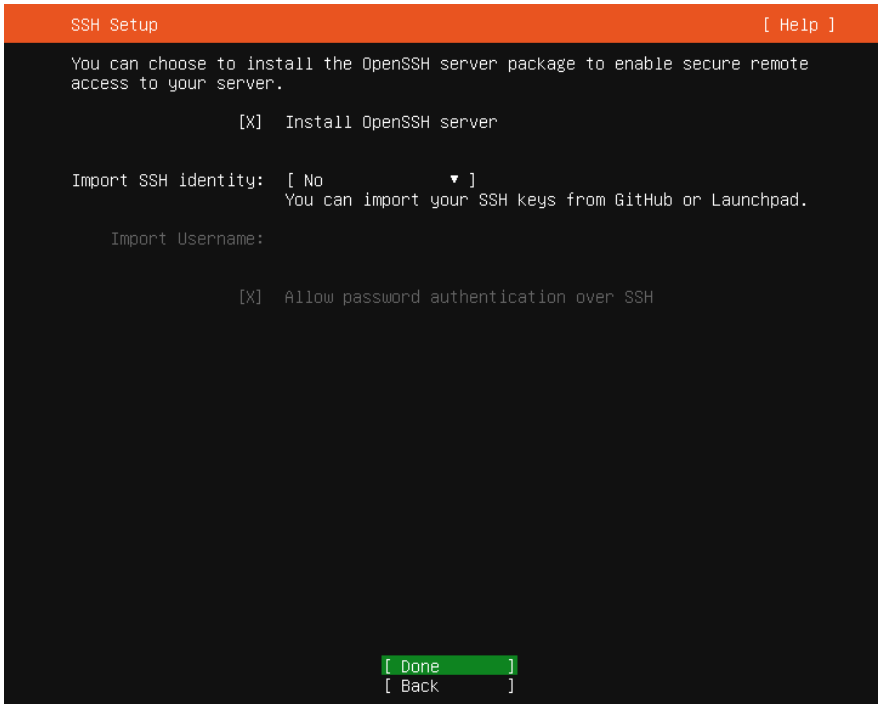




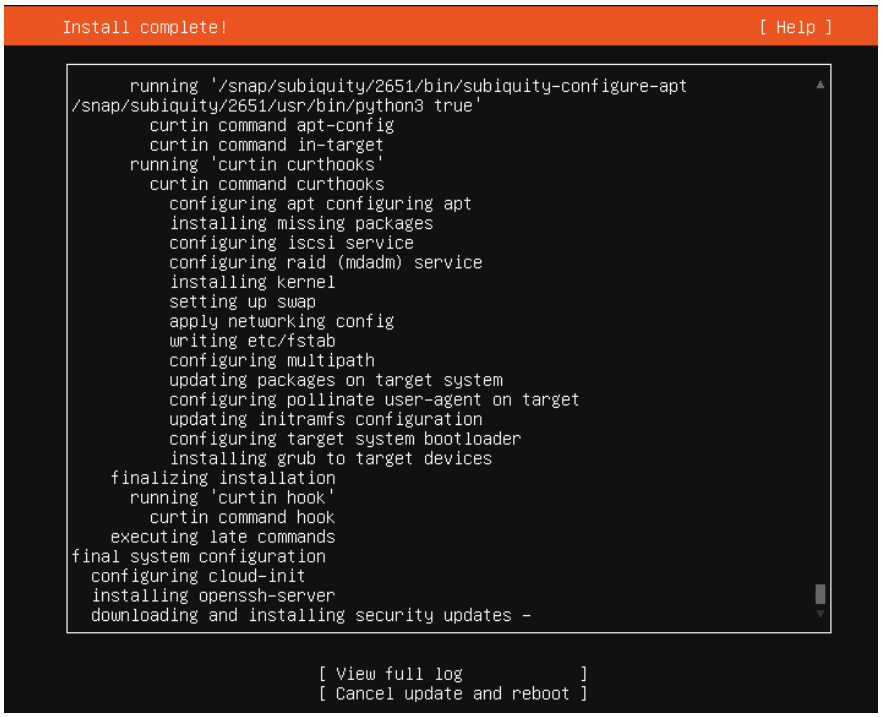
Creo un nuovo utente e scelgo la pasword:



Mi assicuro che il servizio **sshd** venga installato e abilitato;



Completo l'installazione e riavvio la macchina:



4.2 Software Necessario

Appena ho terminato la fase di installazione del SO, ho provveduto ad aggiornarlo con i seguenti comandi, per evitare problemi di compatibilità e software obsoleto:

```
sudo apt update
sudo apt upgrade
```

Poi ho installato i pacchetti necessari (`HTCondor` e `stress-ng`) con:

```
sudo apt install htcondor
sudo apt install stress-ng
```

Durante l'installazione del software `HTCondor` verrà mostrato un menu dove viene chiesto se utilizzare una configurazione già pronta di Condor, io ho scelto `No` in quanto ho preferito configurare tutto a mano per comprendere meglio il file di configurazione ed i vari parametri da modificare.

Il pacchetto `stress-ng` serve per mettere sotto sforzo i vari core della macchina e verrà utilizzato in seguito per testare la configurazione dei due slave.

4.3 IP

Ho assegnato IP statici alle macchine per essere sicuro di poterle sempre raggiungere e che il DHCP del mio router non gli assegni indirizzi diversi col passare del tempo.
Gli indirizzi che ho scelto sono:

- Master: 192.168.178.57
- Slave 1: 192.168.178.54
- Slave 2: 192.168.178.59

Per farlo ho utilizzato la seguente procedura (è uguale per tutte le macchine, basta solo cambiare l'ip):

1. Ho creato il file `99-disable-network-config.cfg` e l'ho popolato con la seguente configurazione:

   ```
   echo "network: {config: disabled}" | sudo tee /etc/cloud/
      cloud.cfg.d/99-disable-network-config.cfg
   ```

2. Ho riscritto il file di configurazione `/etc/netplan/00-installer-config.yaml` con:

   ```
   network:
     version: 2
     renderer: networkd
   ```

```
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.178.57/24
      gateway4: 192.168.178.1
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

3. Ho riavviato la macchina per rendere effettive le modifiche.

Per le altre macchine va modificato il parametro `addresses` con il corrispettivo IP.

Per controllare il corretto funzionamento di questa operazione basta utilizzare il comando `ifconfig enp0s3` e controllare che abbia il giusto indirizzo:

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
inet 192.168.178.57  netmask 255.255.255.0  broadcast
   192.168.178.255
...
```

4.4 hosts & hostname

Ho modificato il file `/etc/hostname` definendo un nome diverso per ogni macchina in modo tale da poterle distinguere più facilmente durante le sessioni SSH:

- Nome Macchina 1 (Master): `joemaster`
- Nome Macchina 2 (Slave1): `joeslave1`
- Nome Macchina 3 (Slave2): `joeslave2`

In tutte le macchine, alla fine del file `/etc/hosts` ho aggiunto le seguenti righe per facilitare poi la configurazione di htcondor:

```
192.168.178.57 nodo1   nodo1.condor
192.168.178.54 nodo2   nodo2.condor
192.168.178.59 nodo3   nodo3.condor
```

4.5 Firewall

Per evitare problemi di comunicazione tra le varie macchine ho deciso di disabilitare il firewall. In Ubuntu 20.4 LTS solitamente il firewall è disabilitato. Ho comunque controllato con il comando:

```
sudo ufw status
```

Se ho questo output vuol dire che il servizio non è attivo:

```
Status: inactive
```

In caso contrario lo andrò a disabilitare con:

```
sudo ufw disable
```

4.6 Configurazione HTCondor

In ogni macchina ho modificato il file di configurazione di HTCondor `/etc/condor/config.d/00personal_condor.config` in modo opportuno in base al compito che dovrà svolgere. Di seguito ho riportato le varie configurazioni per il Master e gli Slave.

4.6.1 JoeMaster

```
CONDOR_HOST = nodo1
COLLECTOR_NAME = Personal Condor at $(FULL_HOSTNAME)

START = TRUE
SUSPEND = FALSE
CONTINUE = TRUE
KILL = FALSE

DAEMON_LIST = COLLECTOR, MASTER, NEGOTIATOR, SCHEDD

HOSTALLOW_WRITE = *.condor
```

Nella lista `DAEMON_LIST` manca `STARTD` in quanto il master non deve eseguire job, ma solo riceverli ed assegnarli ai vari slave.

La variabile `CONDOR_HOST` indica qual è il nodo master, nel mio caso `node1` per via della configurazione in sottosezione 4.4. Questo sarà uguale anche per gli Slave.

Con `HOSTALLOW_WRITE` vado ad autorizzare tutti gli host all'interno del dominio `condor`, definito in sottosezione 4.4.

4.6.2 JoeSlave1/JoeSlave2

```
#Macro(n)
NonCondorLoadAvg = (LoadAvg - CondorLoadAvg)
HighLoad = 0.6
BgndLoad = 0.3
CPU_Busy = ($(NonCondorLoadAvg) >= $(HighLoad))
CPU_Idle = ($(NonCondorLoadAvg) <= $(BgndLoad))
KeyboardBusy = (KeyboardIdle < 10)
MachineBusy = ($(CPU_Busy) || $(KeyboardBusy))
ActivityTimer = (CurrentTime - EnteredCurrentActivity)

CONDOR_HOST = nodo1
COLLECTOR_NAME = Personal Condor at $(FULL_HOSTNAME)

WANT_SUSPEND = True
WANT_VACATE = True

START =  $(CPU_Idle) && !$(KeyboardBusy)
SUSPEND = $(MachineBusy)
CONTINUE = $(CPU_Idle) && KeyboardIdle > 120
PREEMPT = (Activity == "Suspended") && $(ActivityTimer) > 120
KILL = $(ActivityTimer) > 300

DAEMON_LIST = MASTER, STARTD

HOSTALLOW_WRITE = *.condor
```

La configurazione per gli slave è la medesima in quanto avranno tutti e due lo stesso comportamento: dovranno interrompere l'esecuzione del job se viene registrata attività della tastiera (vengono considerate anche le sessioni remote come ssh) o se il carico della CPU supera una certa soglia.
Tramite le variabili `START, SUSPEND e WANT_SUSPEND` e le Macro `CPU_Idel, KeyboardBusy e MachineBusy` riesco ad ottenere il comportamento atteso.

- `WANT_SUSPEND`, se `True` dice a Condor di prendere il considerazione la policy di sospensione all'interno di `SUSPEND`.
- `START`, se `True` vuol dire che il nodo è pronto ad accettare ed eseguire job. Nel mio caso sarà `True` solo quando la CPU sarà sotto il 30% di utilizzo e non verrà rilevata attività della tastiera da più di 10 secondi.
- `SUSPEND`, se `True`, mette in pausa l'esecuzione del job attualmente in lavorazione.

- `CONTINUE` valuta quando riprendere l'esecuzione del job sospeso. Nel mio caso quando la CPU è sotto il 30% di carico e quando non viene registrata attività della tastiera per più di 10 secondi.
- `WANT_VACATE`, se `True`, dice a Condor di valutare la variabile `PREEMPT` per decidere se effettuare la preemption del job oppure no.
- `PREEMPT`, se `True`, effettua la preemption del job, crea un checkpoint del lavoro e riporta il job nella coda dei processi per essere assegnato ad un altro nodo. Nel mio caso avviene quando il job è stato sospeso da più di 120 secondi.
- `KILL` indica quando uccidere immediatamente (senza effettuare preemption) un job. Nel mio caso dopo 300 secondi di inattività.

4.7 Avvio di Condor

Ho abilitato ed avviato il servizio di Condor con:

```
sudo systemctl enable condor.service
sudo systemctl start condor.service
```

Controllo che la configurazione sia andata a buon fine con il comando:

```
condor_status
```

```
Name                      OpSys       Arch   State     Activity LoadAv Mem   ActvtyTime

slot1@joeslave1.fritz.box LINUX       X86_64 Unclaimed Idle      0.310  965  0+00:00:23
slot2@joeslave1.fritz.box LINUX       X86_64 Unclaimed Idle      0.000  965  0+00:00:03
slot3@joeslave1.fritz.box LINUX       X86_64 Unclaimed Idle      0.000  965  0+00:00:23
slot4@joeslave1.fritz.box LINUX       X86_64 Unclaimed Idle      0.000  965  0+00:00:23
slot1@joeslave2.fritz.box LINUX       X86_64 Unclaimed Idle      0.160  965  0+00:00:03
slot2@joeslave2.fritz.box LINUX       X86_64 Unclaimed Idle      0.000  965  0+00:00:19
slot3@joeslave2.fritz.box LINUX       X86_64 Unclaimed Idle      0.000  965  0+00:00:19
slot4@joeslave2.fritz.box LINUX       X86_64 Unclaimed Idle      0.000  965  0+00:00:19

               Total Owner Claimed Unclaimed Matched Preempting Backfill  Drain

X86_64/LINUX       8     0       0         8       0          0        0      0

Total              8     0       0         8       0          0        0      0
```

Vedrò tante righe quanti sono i core che ha ogni macchina. L'output precedente mostra 2 macchine con 4 core ciascuna, indica che la mia configurazione ha funzionato.

4.8 Invio di un Job

Prima di poter inviare un job ho bisogno di un eseguibile, ho perciò deciso di scrivere un programma in C che:

- Accetta 2 argomenti:
  1. int sleep_time: un numero intero che indica i secondi che il programma aspetterà
  2. int input: un numero intero su cui verranno effettuate le operazioni
- Aspetta per `sleep_time` secondi
- Finito il tempo di attesa restituisce `input * 2`

Di seguito riporto il codice che ho compilato con `gcc -o simple simple.c`:

```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;
}
```

Procedo quindi a scrivere il file `jobstarter` per sottomettere il job al master:

```
Universe   = vanilla
Executable = simple
Arguments  = 42 10
Log        = ./logs/simple.log
Output     = ./logs/simple.$(Process).out
Error      = ./logs/simple.$(Process).error
Queue 3
```

- `Executable` indica il file eseguibile che verrà avviato
- `Arguments` è la lista dei parametri da passare la programma
- `Log` indica dove verrà salvato il file di log
- `Output` indica dove verrà salvato il file con al suo interno l'output del programma
- `Error` indica dove verrà salvato il file con possibili errori del programma
- `Queue 3` sottometto 3 volte lo stesso job

Infine mando in esecuzione il job con:

```
condor_submit jobstarter
```

Per controllare in modo efficiente lo stato di avanzamento dei vari job ho scritto il seguente script bash:

```
1 while true
2 do
3     clear
4     date
5     condor_status
6     condor_q -nobatch
7     sleep 2
8 done
```

il quale produrrà un output del tipo:

```
Wed 15 Dec 2021 03:42:25 PM UTC
Name                        OpSys      Arch   State     Activity LoadAv Mem   ActvtyTime

slot1@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.080  965  0+00:00:03
slot2@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:00:20
slot3@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:00:20
slot4@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:00:20
slot1@joeslave2.fritz.box LINUX      X86_64 Claimed   Busy      0.000  965  0+00:00:03
slot2@joeslave2.fritz.box LINUX      X86_64 Claimed   Busy      0.000  965  0+00:00:03
slot3@joeslave2.fritz.box LINUX      X86_64 Claimed   Busy      0.000  965  0+00:00:03
slot4@joeslave2.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:00:20

                 Total Owner Claimed Unclaimed Matched Preempting Backfill  Drain

X86_64/LINUX         8     0       3         5       0          0        0      0

Total                8     0       3         5       0          0        0      0


-- Schedd: joemaster : <192.168.178.57:9618?... @ 12/15/21 15:42:25
ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
28.0    osvaldo         12/15 15:42   0+00:00:06 R  0    0.0 simple 420 10
28.1    osvaldo         12/15 15:42   0+00:00:06 R  0    0.0 simple 420 10
28.2    osvaldo         12/15 15:42   0+00:00:06 R  0    0.0 simple 420 10

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


-- Schedd: joemaster : <192.168.178.57:9618?... @ 12/15/21 15:42:25
OWNER   BATCH_NAME     SUBMITTED   DONE   RUN    IDLE  TOTAL JOB_IDS
osvaldo CMD: simple  12/15 15:42      _      3       _      3 28.0-2

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

Si può notare che sono stati assegnati 3 job alla macchina `JoeSlave2` dato che 3 dei suoi core sono in stato `Claimed/Busy`, ciò vuol dire che stanno performando azioni per condor. Lo stato `Uncleimde/Idle` indica che il nodo è pronto ed in attesa di ricevere un job da eseguire. Questo mi fa capire che il tutto sta funzionando come dovrebbe e che la configurazione è andata a buon fine.

4.9 Test della configurazione

Per testare se la mia configurazione funziona ho utilizzato il tool `stress-ng` per caricare di lavoro i core di una determinata macchina in modo tale da simulare l'attività di un utente. Modifico il job creato in precedenza aumentando il parametro `sleep_time` del mio programma in modo da dare il tempo a condor di valutare tutte le variabili prima che l'esecuzione finisca:

```
  Arguments  = 420 10
```

poi lo avvio con `condor_submit jobstarter` e vado a controllare a quale macchina viene assegnato il lavoro:

```
 Wed 15 Dec 2021 03:42:25 PM UTC
Name                        OpSys      Arch   State     Activity LoadAv Mem   ActvtyTime

slot1@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.080  965  0+00:00:03
slot2@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:00:20
slot3@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:00:20
slot4@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:00:20
slot1@joeslave2.fritz.box LINUX      X86_64 Claimed   Busy      0.000  965  0+00:00:03
slot2@joeslave2.fritz.box LINUX      X86_64 Claimed   Busy      0.000  965  0+00:00:03
slot3@joeslave2.fritz.box LINUX      X86_64 Claimed   Busy      0.000  965  0+00:00:03
slot4@joeslave2.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:00:20

                Total Owner Claimed Unclaimed Matched Preempting Backfill  Drain

X86_64/LINUX        8     0       3         5       0          0        0      0

Total               8     0       3         5       0          0        0      0


-- Schedd: joemaster : <192.168.178.57:9618?... @ 12/15/21 15:42:25
ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
28.0    osvaldo         12/15 15:42   0+00:00:06 R  0    0.0 simple 420 10
28.1    osvaldo         12/15 15:42   0+00:00:06 R  0    0.0 simple 420 10
28.2    osvaldo         12/15 15:42   0+00:00:06 R  0    0.0 simple 420 10

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


-- Schedd: joemaster : <192.168.178.57:9618?... @ 12/15/21 15:42:25
OWNER   BATCH_NAME     SUBMITTED   DONE   RUN    IDLE  TOTAL JOB_IDS
osvaldo CMD: simple  12/15 15:42      _      3       _      3 28.0-2

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

Nel mio caso i 3 job vengono eseguiti dai core `1, 2 e 3` della macchina `JoeSlave2`, quindi su di essa avvio lo stress test con:

```
stress-ng --cpu 4 &
```

Utilizzo la `&` alla fine per eseguire il comando in background e potere avviare `htop` in modo da controllare se effettivamente il carico della CPU aumenta:

```
1  [||||||||||||||||||||||||||||||||||||||||||||||||100.0%]   Tasks: 37, 32 thr; 4 running
2  [||||||||||||||||||||||||||||||||||||||||||||||||100.0%]   Load average: 3.42 1.66 0.68
3  [||||||||||||||||||||||||||||||||||||||||||||||||100.0%]   Uptime: 00:08:38
4  [||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
Mem[||||||||                           197M/3.77G]
Swp[                                      0K/0K]
PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
1228 osvaldo    20   0  101M  7220  3732 R 400.  0.2  1:41.12 stress-ng-cpu
1 root       20   0  163M 11372  8272 S  0.0  0.3  0:00.67 /sbin/init maybe-ubiquity
392 root       19  -1 66816 28596 27576 S  0.0  0.7  0:00.09 /lib/systemd/systemd-journald
422 root       20   0 21388  5436  3896 S  0.0  0.1  0:00.15 /lib/systemd/systemd-udevd
594 root       RT   0  273M 17940  8200 S  0.0  0.5  0:00.00 /sbin/multipathd -d -s
595 root       RT   0  273M 17940  8200 S  0.0  0.5  0:00.00 /sbin/multipathd -d -s
596 root       RT   0  273M 17940  8200 S  0.0  0.5  0:00.00 /sbin/multipathd -d -s
597 root       RT   0  273M 17940  8200 S  0.0  0.5  0:00.01 /sbin/multipathd -d -s
598 root       RT   0  273M 17940  8200 S  0.0  0.5  0:00.00 /sbin/multipathd -d -s
599 root       RT   0  273M 17940  8200 S  0.0  0.5  0:00.00 /sbin/multipathd -d -s
593 root       RT   0  273M 17940  8200 S  0.0  0.5  0:00.04 /sbin/multipathd -d -s
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Da questo output posso constatare che l'utilizzo della CPU di questa macchina è arrivato al `100%`, quindi aspetto qualche istante e controllo che la macchina raggiunga lo stato di `Claimed/Suspended`:

```
Wed 15 Dec 2021 04:13:54 PM UTC
Name                      OpSys      Arch   State     Activity  LoadAv Mem   ActvtyTime

slot1@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle       0.000  965  0+00:29:46
slot2@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle       0.000  965  0+00:30:03
slot3@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle       0.000  965  0+00:30:03
slot4@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle       0.000  965  0+00:30:03
slot1@joeslave2.fritz.box LINUX      X86_64 Claimed   Suspended  0.620  965  0+00:00:03
slot2@joeslave2.fritz.box LINUX      X86_64 Claimed   Suspended  0.770  965  0+00:00:03
slot3@joeslave2.fritz.box LINUX      X86_64 Claimed   Suspended  0.650  965  0+00:00:03
slot4@joeslave2.fritz.box LINUX      X86_64 Owner     Idle       0.320  965  0+00:00:03

                 Total Owner Claimed Unclaimed Matched Preempting Backfill  Drain

X86_64/LINUX     8     1       3         4       0          0        0      0

Total            8     1       3         4       0          0        0      0


-- Schedd: joemaster : <192.168.178.57:9618?... @ 12/15/21 16:13:55
ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
30.0    osvaldo         12/15 16:10   0+00:02:58 S  0    0.0 simple 420 10
30.1    osvaldo         12/15 16:10   0+00:02:58 S  0    0.0 simple 420 10
30.2    osvaldo         12/15 16:10   0+00:02:58 S  0    0.0 simple 420 10

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


-- Schedd: joemaster : <192.168.178.57:9618?... @ 12/15/21 16:13:55
OWNER   BATCH_NAME     SUBMITTED   DONE   RUN    IDLE  TOTAL JOB_IDS
osvaldo CMD: simple  12/15 16:10      _      _      3      3 30.0-2

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

Ora attendo che la condizione nella variabile `PREEMPT` definita in sottosottosezione 4.6.2 raggiunga il valore `True` in modo tale da riportare i job nella coda e riassegnarli ad una macchina in stato `Unclaimed/Idle`. La macchina sotto stress avrà lo stato di `Owner/Idle`.

```
Wed 15 Dec 2021 04:16:03 PM UTC
Name                      OpSys      Arch   State     Activity LoadAv Mem   ActvtyTime

slot1@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:29:46
slot2@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:30:03
slot3@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:30:03
slot4@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:30:03
slot1@joeslave2.fritz.box LINUX      X86_64 Owner     Idle      0.950  965  0+00:00:03
slot2@joeslave2.fritz.box LINUX      X86_64 Owner     Idle      1.000  965  0+00:00:03
slot3@joeslave2.fritz.box LINUX      X86_64 Owner     Idle      1.000  965  0+00:00:03
slot4@joeslave2.fritz.box LINUX      X86_64 Owner     Idle      0.320  965  0+00:00:03

                Total Owner Claimed Unclaimed Matched Preempting Backfill  Drain

X86_64/LINUX    8     4       0         4       0          0        0      0

Total           8     4       0         4       0          0        0      0
```

```
-- Schedd: joemaster : <192.168.178.57:9618?... @ 12/15/21 16:16:03
ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
30.0   osvaldo         12/15 16:10   0+00:04:55 I  0    0.0 simple 420 10
30.1   osvaldo         12/15 16:10   0+00:03:10 I  0    0.0 simple 420 10
30.2   osvaldo         12/15 16:10   0+00:03:40 I  0    0.0 simple 420 10

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


-- Schedd: joemaster : <192.168.178.57:9618?... @ 12/15/21 16:16:03
OWNER   BATCH_NAME     SUBMITTED   DONE   RUN    IDLE  TOTAL JOB_IDS
osvaldo CMD: simple  12/15 16:10      _      _      3      3 30.0-2

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

Dal seguente output si nota che i core della macchina `JoeSlave2` sono tutti in stato `Owner/Idle` (sono in uso dall'utente e quindi non possono eseguire job) e che i job in esecuzione su di essa sono stati assegnati alla macchina `JoeSlave1` in quanto si trova nello stato `Claimed/Busy`.

```
Name                      OpSys      Arch   State     Activity LoadAv Mem   ActvtyTime

slot1@joeslave1.fritz.box LINUX      X86_64 Claimed   Busy      0.000  965  0+00:00:03
slot2@joeslave1.fritz.box LINUX      X86_64 Claimed   Busy      0.000  965  0+00:00:03
slot3@joeslave1.fritz.box LINUX      X86_64 Claimed   Busy      0.000  965  0+00:00:03
slot4@joeslave1.fritz.box LINUX      X86_64 Unclaimed Idle      0.000  965  0+00:35:03
slot1@joeslave2.fritz.box LINUX      X86_64 Owner     Idle      0.970  965  0+00:00:36
slot2@joeslave2.fritz.box LINUX      X86_64 Owner     Idle      1.000  965  0+00:02:21
slot3@joeslave2.fritz.box LINUX      X86_64 Owner     Idle      1.000  965  0+00:01:51
slot4@joeslave2.fritz.box LINUX      X86_64 Owner     Idle      1.000  965  0+00:04:56

                Total Owner Claimed Unclaimed Matched Preempting Backfill  Drain

X86_64/LINUX        8     4       3         1       0          0        0      0

Total               8     4       3         1       0          0        0      0


-- Schedd: joemaster : <192.168.178.57:9618?... @ 12/15/21 16:17:25
ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
30.0   osvaldo         12/15 16:10   0+00:05:23 R  0    0.0 simple 420 10
30.1   osvaldo         12/15 16:10   0+00:03:38 R  0    0.0 simple 420 10
30.2   osvaldo         12/15 16:10   0+00:04:08 R  0    0.0 simple 420 10

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


-- Schedd: joemaster : <192.168.178.57:9618?... @ 12/15/21 16:17:25
OWNER   BATCH_NAME     SUBMITTED   DONE   RUN    IDLE  TOTAL JOB_IDS
osvaldo CMD: simple  12/15 16:10      _      3      _      3 30.0-2

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