HTCondor As A ~~Service~~

Batch System

Stefano Dal Pra

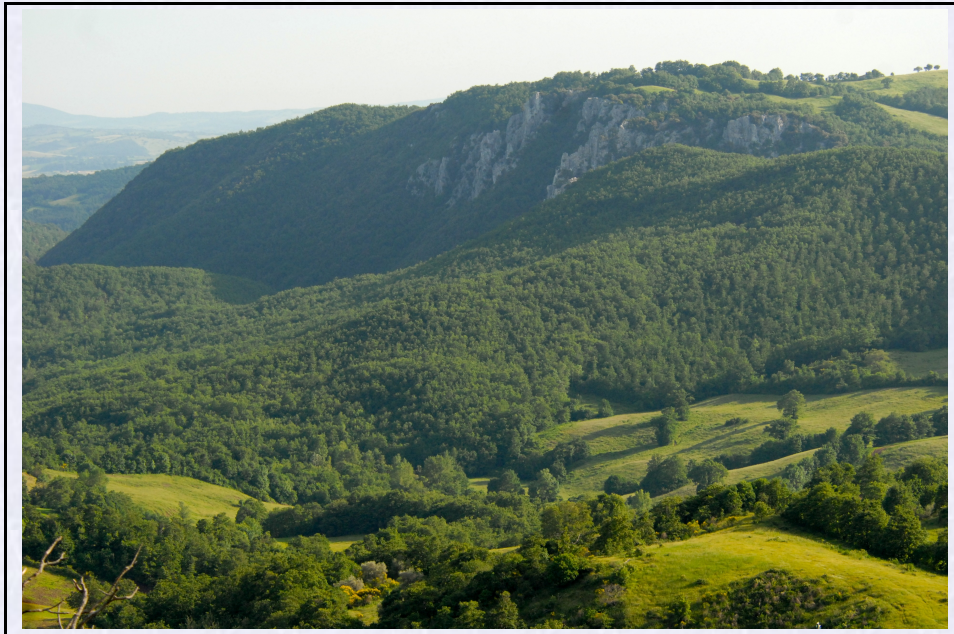
Perugia 28 Nov 2019,

Email: stefano.dalpra@cnaf.infn.it

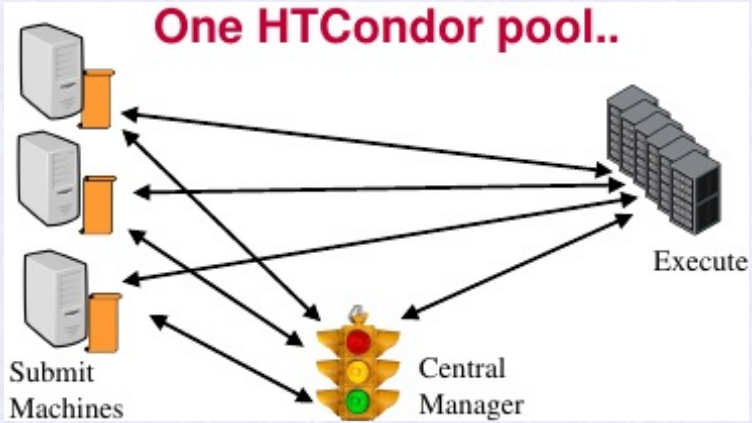
Premessa e disclaimer

- Molte parti di queste slides si rifanno in modo diretto a presentazioni date agli "HTCondor week" o simili eventi, su temi attinenti. Si consiglia di fare riferimento a questi per approfondimenti
- HTC week 2019: https://agenda.hep.wisc.edu/event/1325/ (User oriented)
- HTC week 2018: https://agenda.hep.wisc.edu/event/1201/
- HTC workshop 2019: https://indico.cern.ch/event/817927/ (Admin oriented)
- https://htcondor.readthedocs.io

Conoscenza introduttiva



Panoramica per orientarsi all'ambiente



Panoramica Per HTCondor

Conoscenza dettagliata, “offroad”

L’ambiente, dettagli

```
10/23/19 18:16:15 (D_SECURITY) SECMAN: command 60
19> from TCP port 1747 (non-blocking).
10/23/19 18:16:15 (D_SECURITY) SECMAN: waiting fo
9>.
10/23/19 18:16:15 (D_ALWAYS:2) DaemonKeepAlive: L
10/23/19 18:16:15 (D_ALWAYS:2) SharedPortClient:
.35:9619> for shared port id 20786_3e80
10/23/19 18:16:15 (D_SECURITY) SECMAN: resuming c
.11.35:9619> from TCP port 1747 (non-blocking).
10/23/19 18:16:15 (D_SECURITY) SECMAN: using sess
a6a6 for {<192.135.11.35:9619?addrs=192.135.11.35
10/23/19 18:16:15 (D_SECURITY) SECMAN: successful
10/23/19 18:16:15 (D_SECURITY) SECMAN: startComma
10/23/19 18:16:15 (D_SECURITY) IPVERIFY: Subsyste
10/23/19 18:16:15 (D_SECURITY) IPVERIFY: Permissi
10/23/19 18:16:15 (D_SECURITY) IPVERIFY: Subsyste
10/23/19 18:16:15 (D_SECURITY) IPVERIFY: Permissi
10/23/19 18:16:15 (D_SECURITY) IPVERIFY: allow RE
10/23/19 18:16:15 (D_SECURITY) ipverify: READ opt
10/23/19 18:16:15 (D_SECURITY) IPVERIFY: Subsyste
10/23/19 18:16:15 (D_SECURITY) IPVERIFY: Permissi
```

HTCondor, dettagli

Introduzione. Il problema del calcolo

Devo elaborare dei dati con alcuni software di calcolo.

Approccio ingenuo. Uso ssh:

1. con ssh mi collego a un host remoto, copio dati necessari (input)
2. lancio un programma (executable)
3. seguo l'esecuzione
4. alla fine recupero i risultati (output)

Per poche esecuzioni, diverse tutte diverse può anche andar bene cosí.

- Se devo eseguire molte volte lo stesso task
- Se ci sono molti host disponibili (non sempre liberi)

l'approccio ingenuo diventa presto impraticabile

Serve un tool che esegua i passaggi $1, ..., 4$ al nostro posto

Alcuni dettagli "impliciti":

1. Quale host? È libero?
2. È adatto per quel che devo fare? C'è il necessario per eseguire il mio task?
3. E se qualcosa va storto? Chi preme CTRL-C ? E quando?

Intuizione. Servono:

- un "linguaggio" per descrivere il task
- un "intermediario" che riceve la richiesta
- la accoda con eventuali altre
- la serve "quando possibile" $\rightarrow$ (conosce lo stato degli host disponibili)

Un tool siffatto si chiama "Batch System"

HTCondor è un batch system

- Gestisce ed esegue lavori (**Job**) al nostro posto.
- Li schedula sequenzialmente (per non sovraccaricare) su un singolo computer
- oppure su gruppi di computer (**pool**, in gergo HTCondor) anche non direttamente accessibili all'utente
- Gestisce task sottomessi da piú utenti, da un host detto Submit Node
- Interpreta files di testo che descrivono i task (**submit files**)

Submit file

1.	`executable = my_program` `arguments = inp1.dat inp2.dat result.out`	Eseguibile e argomenti

2.	`should_transfer_files = YES`
	`transfer_input_files = us.dat, wi.dat`
	`when_to_transfer_output = ON_EXIT`
		trasferimento IN e OUT

3.	`log = job.log`
	`output = job.out`
	`error = job.err`
		log, stdout, stderr

4.	`request_cpus = 1`
	request_memory = 20MB
	`request_disk = 20MB`
		Resource requests

5.	`queue 5`	istanzia 5 copie

Sottomettere job, vederne lo stato e lo storico

Da un Submit Node (anche detto schedd)

```
[sdalpra@sn-01 p308]$ condor_submit p308.sub
Submitting job(s)...
5 job(s) submitted to cluster 142390.
```

```
[sdalpra@sn-01 p308]$ condor_q
 -- Schedd: sn-01: <...:9618?...  @ 11/24/19 21:56:28
OWNER BATCH_NAME SUBMITTED DONE RUN IDLE TOTAL JOB_IDS
sdalpra ID: 142390 11/24 21:56     _      5      _      5      142390.0-4
```

```
[sdalpra@sn-01 ~]$ condor_history 142389.
 ID     OWNER          SUBMITTED   RUN_TIME     ST COMPLETED   CMD
142390.5 sdalpra      11/24 21:45  0+00:00:02 C  11/24 21:46 htcp308 0 0 1 1
```

Note nel submit file c'è **`queue 5`**. Il submit risponde con il `ClusterId`. (142390)

Nota2 Avremo cinque `JobId = ClusterId.ProcId`, `ProcId`=$0, \ldots, 4$.

Richiesta risorse

- È importante indicare risorse adeguate per il job.
- specificando limiti troppo bassi il job può essere fermato da HTCondor (HELD)
- Con limiti troppo alti è piú difficile trovare host adeguati

Submit multiplo

```
executable = my_program
arguments = file$(ProcId).in file$(ProcId).out

log = job_$(ClusterId)_$(ProcId).log
output = job_$(ClusterId)_$(ProcId).out
error = job_$(ClusterId)_$(ProcId).err

queue 5
```

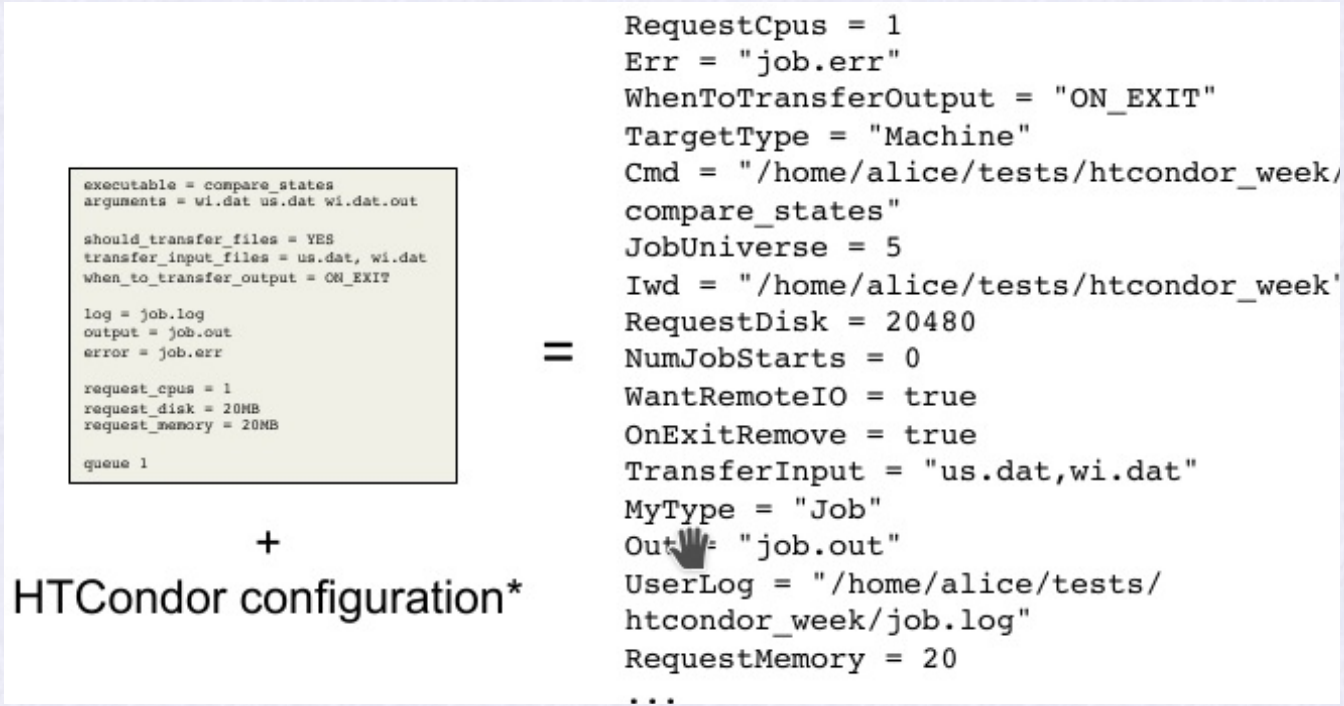
Usando `$(ClusterId)`, `$(ProcId)` si assegnano valori unici per i singoli job

Altri esempi

vedere file `submit_tutorial.txt`

ClassAd

HTC mantiene un elenco di informazioni `Key = Value` di stato per ogni Job e per ogni Nodo, dette **ClassAd** (`condor_q -l <ClusterId>` per vedere quelle di un job)



Host ClassAd

\+

HTCondor configuration

=

```
HasFileTransfer = true
DynamicSlot = true
TotalSlotDisk = 4300218.0
TargetType = "Job"
TotalSlotMemory = 2048
Mips = 17902
Memory = 2048
UtsnameSysname = "Linux"
MAX_PREEMPT = ( 3600 * 72 )
Requirements = ( START ) &&
( IsValidCheckpointPlatform ) &&
( WithinResourceLimits )
OpSysMajorVer = 6
TotalMemory = 9889
HasGluster = true
OpSysName = "SL"
HasDocker = true
```

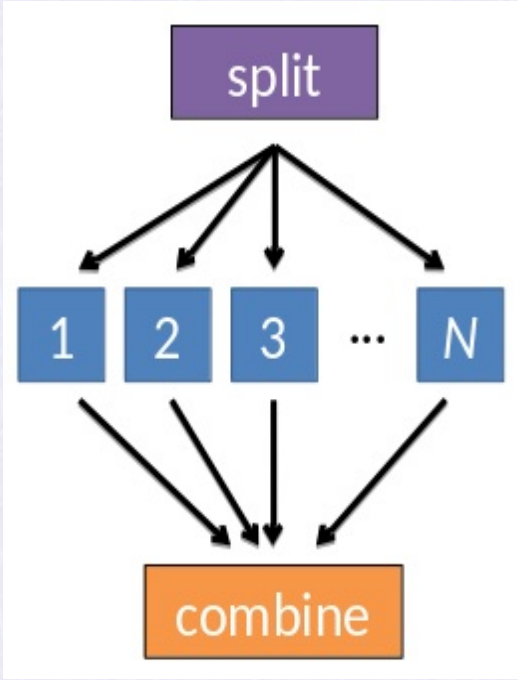
Il **Central Manager** ciclicamente confronta i ClassAd dei Job con quelli degli host. Quando c'è un match il job può essere gestito dal corrispondente host. Le interazioni successive avvengono tra **Submit Node** e **Worker Node**.

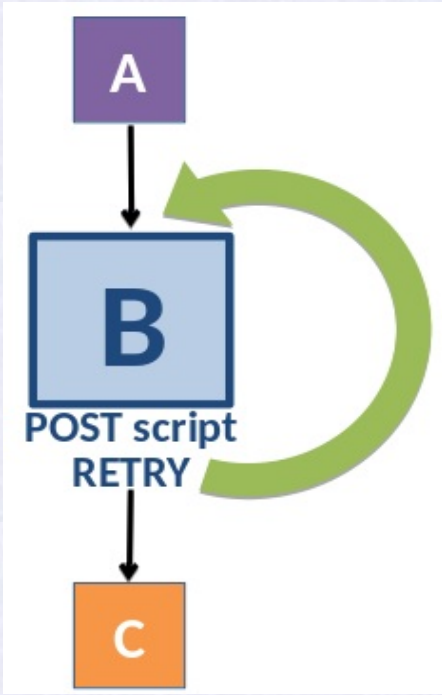
DAG Jobs

Esempio: Un task produce N files
Altri task lavorano sugli n files...
Un ultimo task combina gli N risultati.

DAG = **D**irect **A**cyclic **G**raph

Elementi:
Job Nodes, Parent, Child, Edges
Sono possibili DAG con cicli
Esempi: cfr. submit_tutorial.txt





Backup slides

Load Balancing

Priority Assessment