



Federica Fanzago
INFN Padova

Batch system elastici su cloud

Premessa

- In cloud i progetti possono utilizzare risorse fino ad un massimo definito dalla quota a loro associata
 - quota statica relativa a CPU, spazio disco, RAM, Volume.
- Le VM vengono istanziate con caratteristiche di calcolo ben definite dai flavor
 - preferibile avere flavor di piccole dimensioni per miglior utilizzo hypervisor e maggior possibilità creazione VM.
- Le risorse di un progetto sono condivise tra gli utenti che ne fanno parte
 - le VM devono essere terminate quando non utilizzate.
- Creazione - distruzione manuale delle VM
 - difficile gestione se il carico di lavoro non è costante nel tempo.

Queste premesse non sempre soddisfano le esigenze di calcolo dei progetti, in particolare per quelli il cui carico di lavoro varia nel tempo e dal tipo di processo che deve essere eseguito.

Esempio: progetto CMS supportato dalla cloud padovana (Compact Muon Solenoid, esperimento di fisica delle alte energie all'acceleratore LHC del CERN).

Caso d'uso: progetto CMS

Le analisi e le produzioni dati della comunita' locale prevedono l'utilizzo di risorse sia in modalita' interattiva che batch.

I processi sono single core e non parallelizzabili. La configurazione standard richiesta e' 2GB RAM per CPU.

Il numero di processi di una produzione o analisi non e' costante ma varia a seconda del tipo di studio.

Un task puo' anche comprendere migliaia di job della durata di alcune ore.

Per l'utente l'ideale sarebbe quindi poter avere in cloud:

- VM personale con flavor "enorme" per poter girare sulla stessa macchina piu' processi possibili → non fattibile
- un cluster di VM gestite da batch-system dove sottomettere i processi.

L'utilizzo statico delle risorse cloud non risponde pienamente alle esigenze degli utenti.

Batch-system elastico in cloud

Per sfruttare al meglio le risorse del progetto si e' applicato il concetto di cluster elastico:

- un pool di risorse non preassegnate in modo statico ma allocate e rilasciate dinamicamente in base al “carico” di lavoro del singolo utente
 - i processi degli utenti vengono sottomessi dal nodo che agisce come master per il batch-system del pool
 - il batch-system garantisce l'utilizzo equo delle risorse condivise tra utenti
 - i nodi slave vengono creati e terminati a seconda del numero di processi in coda nel master.

Implementazione

Come batch-system e' stato utilizzato HTCondor (<http://research.cs.wisc.edu/htcondor/>) che permettere di collegare runtime i nodi slave al cluster.

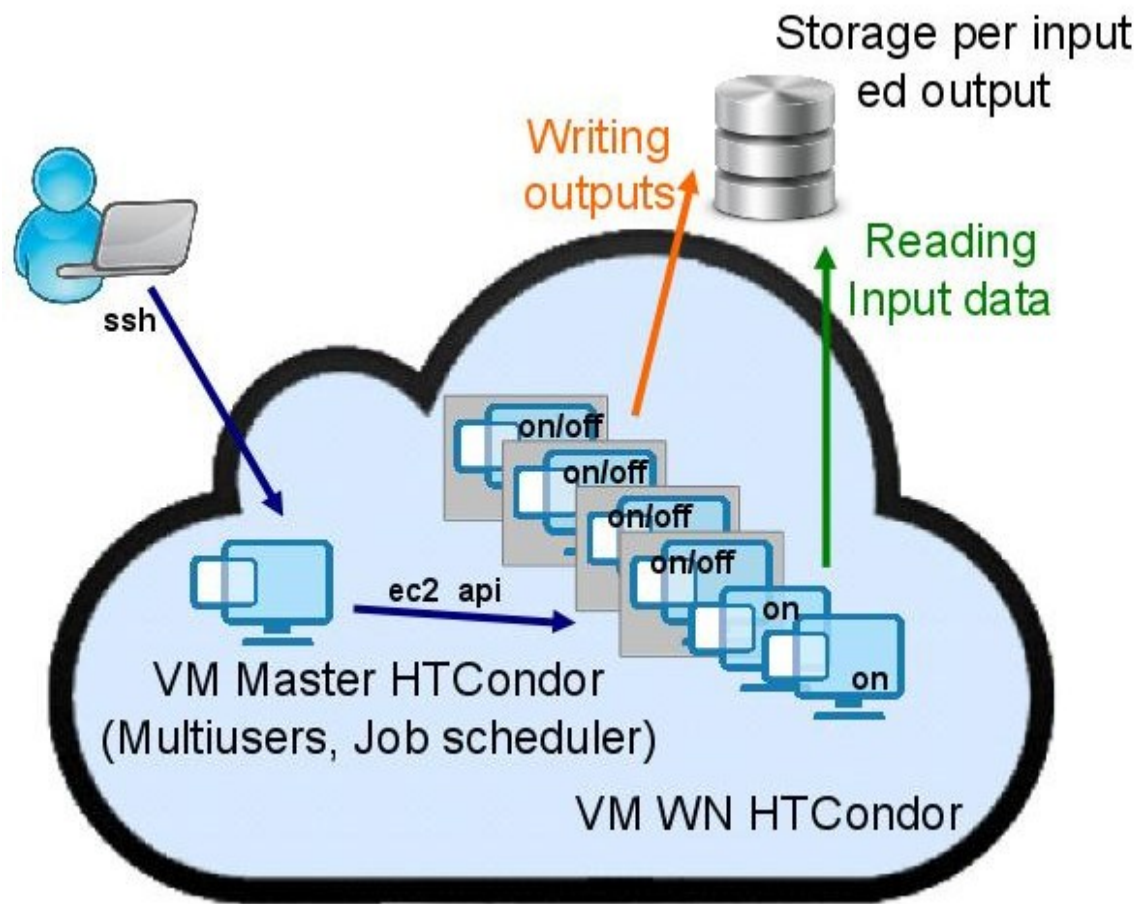
L'applicativo python elastiq (<https://github.com/dberzano/elastiq>), installato ed attivo sul nodo master del cluster, gestisce la scalabilita' dei nodi permettendo:

- allocazione e rilascio delle risorse attraverso api ec2, compatibili con diverse infrastrutture cloud
- configurazione del tempo di inattivita' delle VM prima della loro distruzione e del tempo di attesa dei job in coda prima della creazione di nuove VM
- configurazione del numero minimo e massimo di slave attivi.

Elastiq parla con il batch-system attraverso apposita plugin.

Per CMS vengono utilizzate immagini basate sulla distribuzione Linux SL6, HTCondor 8.4 ed elastiq 0.9.10.

Schema implementativo



I parametri di configurazione dei nodi slave, come immagine e flavor, vengono passati in fase di creazione del cluster via `user_data` file ad `elastiq` che li utilizzerà per attivare i nodi.

```
Elastiq.conf
[elastiq]
check_queue_every_s = 240
idle_for_time_s = 600
waiting_jobs_time_s = 600
n_jobs_per_vm = 6
batch_plugin = htcondor
estimated_vm_deploy_time_s = 420
check_vms_every_s = 300
[ec2]
flavour = CMS_6core
api_url = https://cloud-areapd....
user_data_b64 = lyERQVz1.....
image_id = ami-be72cea8
aws_secret_access_key = xxx
aws_access_key_id = yyy
key_name = zzz
[quota]
max_vms = 165
min_vms = 1
```

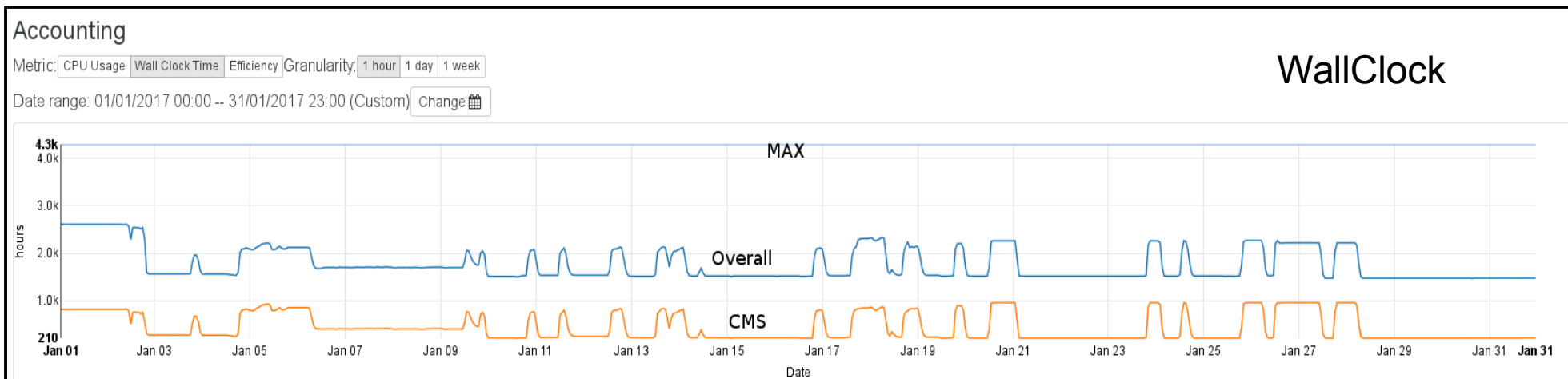
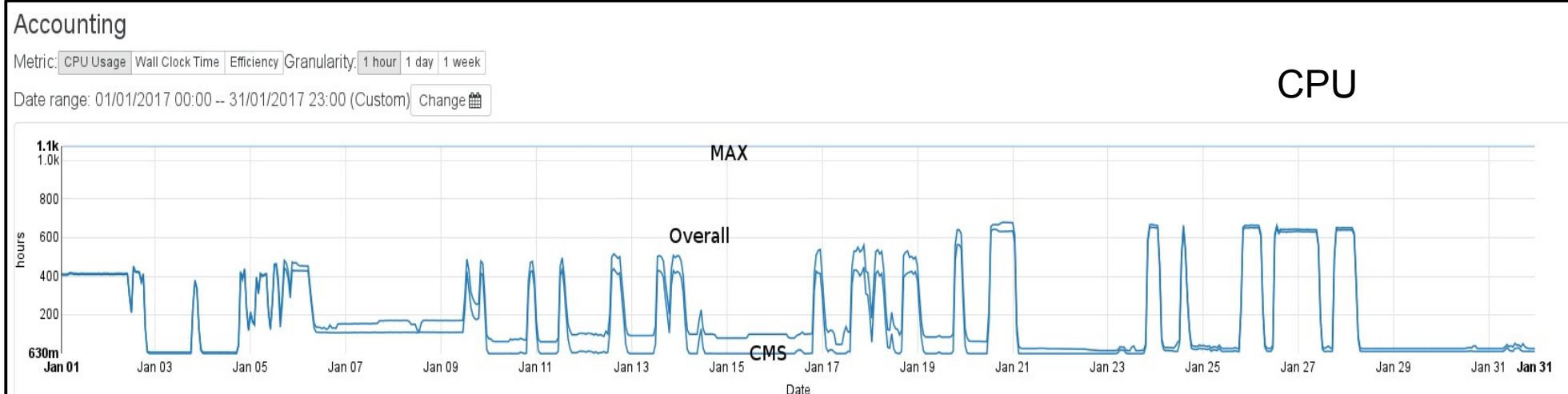
Automazione creazione cluster elastico caso generale

Per facilitare la creazione di cluster elastici con diverse distribuzioni linux, agli utenti della cloud padovana viene fornito l'applicativo python ElastiQ Cluster Maker (ECM) che

- permette di ottenere lo user_data file da usare come script di post-creation al momento dell'istanziamento del master
 - contiene lo script di installazione e configurazione dei nodi del cluster
- utilizza diverse immagini basate sulle diverse distribuzioni
 - SL6, CentOS7, Ubuntu14.04.

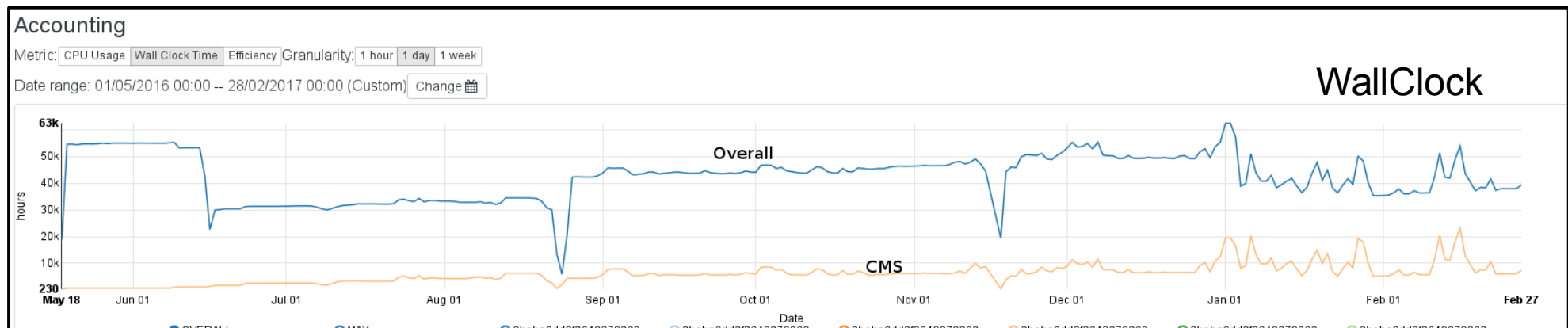
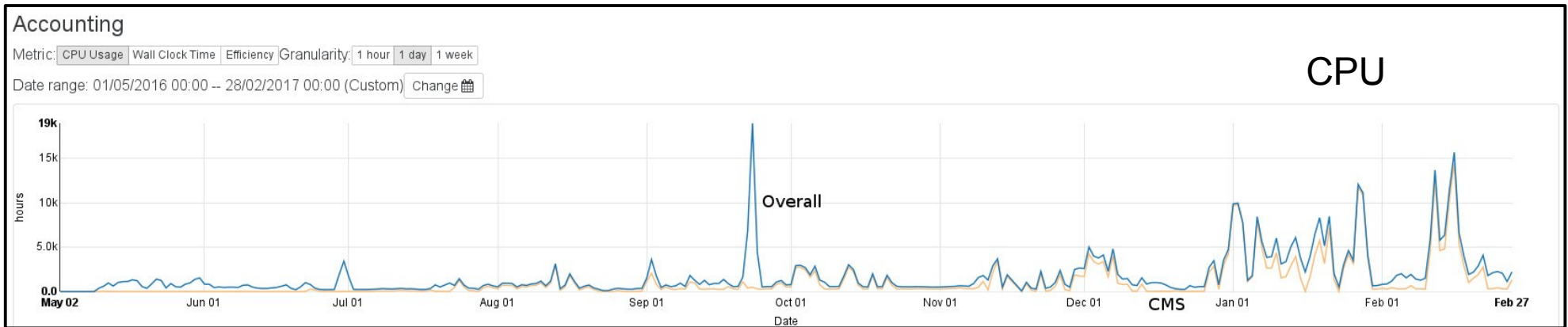
E' stato inoltre creato un repository locale di HTCondor ed elastiQ per le diverse distribuzioni.

Dati utilizzo risorse cloud gennaio 2017



In un mese sono stati eseguiti ~50000 processi con picchi di 750 processi simultanei (125 VM da 6 VCPU)

Dati utilizzo risorse cloud 05/16-02/17



L'utilizzo di cluster elastici a partire da settembre 2016 ha permesso un miglior sfruttamento delle risorse disponibili rispetto a quello ottenuto con VM statiche

THE END

Backup Slides

Alcune configurazioni

Elastiq.conf

[elastiq]

check_queue_every_s = 240

idle_for_time_s = 600

waiting_jobs_time_s = 600

n_jobs_per_vm = 6

batch_plugin = htcondor

estimated_vm_deploy_time_s = 420

check_vms_every_s = 300

#log_level = 20

[ec2]

flavour = CMS_6core

api_url = <https://cloud-areapd.pd.infn.it:8788/services/Cloud>

user_data_b64 = lyERQVz1.....

image_id = ami-be72cea8

aws_secret_access_key = xxx

aws_access_key_id = yyy

key_name = zzz

[quota]

max_vms = 165

min_vms = 1

[root@htcondor-master ~]# condor_status

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@10-64-22-230	LINUX	X86_64	Unclaimed Idle	0.010	1985	3+02:27:00	
slot2@10-64-22-230	LINUX	X86_64	Unclaimed Idle	0.000	1985	3+02:25:57	
slot3@10-64-22-230	LINUX	X86_64	Unclaimed Idle	0.000	1985	3+00:57:44	
slot4@10-64-22-230	LINUX	X86_64	Unclaimed Idle	0.000	1985	3+01:06:10	
slot5@10-64-22-230	LINUX	X86_64	Unclaimed Idle	0.000	1985	2+23:45:28	
slot6@10-64-22-230	LINUX	X86_64	Unclaimed Idle	0.000	1985	2+21:45:55	
slot1@10-64-22-31.	LINUX	X86_64	Unclaimed Idle	0.000	1985	3+00:39:41	
slot2@10-64-22-31.	LINUX	X86_64	Unclaimed Idle	0.000	1985	3+02:23:12	
slot3@10-64-22-31.	LINUX	X86_64	Unclaimed Idle	0.000	1985	3+01:36:41	
slot4@10-64-22-31.	LINUX	X86_64	Unclaimed Idle	0.000	1985	2+23:55:06	
slot5@10-64-22-31.	LINUX	X86_64	Unclaimed Idle	0.000	1985	3+00:39:30	
slot6@10-64-22-31.	LINUX	X86_64	Unclaimed Idle	0.000	1985	3+02:49:13	

Machines Owner Claimed Unclaimed Matched Preempting

X86_64/LINUX	12	0	0	12	0	0
--------------	----	---	---	----	---	---

Total	12	0	0	12	0	0
-------	----	---	---	----	---	---

Setup CMS

- CMS che utilizza la cloud per riprodurre su macchine virtuali l'infrastruttura locale finora utilizzata composta da userinterface che permettono non solo di sottomettere processi di analisi nell'ambiente distribuito grid ma anche di girare processi di analisi e produzione dati in modo interattivo.
- Dati di input vengono letti dal dcache storage del tier2 di Legnaro e i risultati vengono poi scritti in uno storage server con filesystem LUSTRE.
- Il codice di esperimento e' accessibile via CernVM-FS e il sistema di autorizzazione degli utenti e' gestito dall'ldap
- Queste macchine permettono pure l'accesso ai nodi del tier2 di Padova e Legnaro tramite batch system Isf.
- Questo ambiente cosi' articolato e' stato correttamente riprodotto in cloud e viene quotidianamente utilizzato dagli utenti locali di CMS sia con VM interattive che con pool batch