

CINECA-HPC

T. Boccali, S. Dal Pra, D. Spiga
Special Thanks: M. Morgotti

Scopo del lavoro

- Accedere a risorse HPC attraverso gli ordinari canali di sottomissione via Grid
- Use case: Grant di 30M-ore su cluster di nodi:
 - INTEL KNL, 68 core x 4 thread = 272 “slot”
 - 96GB RAM, (10 riservati al SO)
 - Batch System: Slurm
 - Outgoing connectivity solo verso CERN e CNAF
- **NB:** il taglio minimo è “whole node” (Slurm assegna nodi interi ai job)
 - Un comune job singlecore di 1h costa 68h !

| | |
|----------------|---------------------|
| Project name | LHC@HPC |
| Research field | High Energy Physics |

Principal Investigator (PI)

| | |
|--------------------------|--|
| Title (Dr., Prof., etc.) | Dr |
| Last name | Boccali |
| First name | Tommaso |
| Organisation name* | Istituto Nazionale di Fisica Nucleare (INFN) |
| Department* | Sezione di Pisa |
| Group* | Scientific Computing, CMS |
| Country | Italy |

Cosa serve?

- Un modo per far girare i jobs di LHC li'
 - Requisiti minimi di sistema, rete, servizi, ...

- Un modo per far atterrare i jobs di LHC li'
 - Idea: utilizzare come estensione elastica del CNAF, far vedere la cosa SLURM locale attraverso un HTCondorCE/Slurm

Requisiti minimi

- CVMFS, Singularity, external networking verso CNAF (storage) e CERN (calibrazioni): installati su nostra richiesta
- Utenti locali (poolaccount, 6 x VO) creati manualmente, su richiesta
- Per accesso via Grid un host locale al CINECA configurato come HTCondor-CE a nostra cura.
 - Problema: solo accesso utente al CE; tutte le necessarie operazioni privilegiate eseguite dall'admin locale (M. Morgotti) su nostra indicazione
 - Privilegi sudo per alcune operazioni (edit conf, restart ce, view logfile)



Install e setup di HTCondor-CE per Slurm

- Si segue la documentazione di OSG e alcune note locali per il setup degli HTC-CE del cnafr
(http://wiki.infn.it/progetti/htcondor-tf/htcondor-ce_setup)
- Nota: a differenza dei siti WLCG, i comuni tool di configurazione non sono presenti
- Tutti i setup sono stati fatti (e rifatti...) a mano, sempre su indicazioni date all'admin
- La parte più ostica: GSI (voms, argus, poolaccount)



Setup specifici e workaround

- I nodi di cluster HPC hanno minore spazio HDD locale insufficiente → **la home viene riconfigurata su area GPFS agendo sul “prolog” di Slurm**
- Un bug in `slurm_submit.sh` converte `#cores` richiesti in `#nodi` richiesti → **applicata patch, in attesa del fix**
- Un problema di configurazione JumboFrame in un router GARR causa timeout nei trasferimenti da storage CNAF ai nodi Marconi → **In attesa di fix**
- Per il resto, tutta la filiera di sottomissione funziona correttamente



Lato esperimenti - checks, status etc

- **2018 (grant request):** ATLAS, CMS e LHCb effettuato tests minimali (tipicamente workflows interattivi).
- **2019 (grant utilization):** configurazione cambiata sui nodi per renderla di produzione, si ricomincia quasi da zero. ALICE si unisce.

Stato dell'arte (comandi etc) mantenuto [qui](#)

Integrazione con CMS: in sintesi

NOTA: Le risorse del CINECA sono integrate in CMS come un'estensione di T1_IT_CNAF e non come un nuovo sito

- Di fatto per CMS significa no storage element, no SiteDB/CRIC entry etc... ma solo CPU aggiuntiva
 - Per l'output facciamo riferimento allo Storage del T1
 - Per l'input usiamo XRootD proxy cache @ CNAF (vede tutta la federazione, nessuna limitazione vera)
- In sostanza quindi l'integrazione del CINECA con CMS consiste nell'aggiungere un nuovo CE nella Factory di GlideIn WMS
 - E dire alla Factory che quel CE è parte del pool dei CE del T1_IT_CNAF ma che dietro ha risorse "diverse" (vedi dopo)

Pre-requisiti per il runtime env.

- CVMFS & Singularity
 - Ok dopo qualche iterazione (versione buggata etc...)
- Squid for CMS Frontier
 - Usa gli stessi squid CINECA-operated (6 nodi?); squid CNAF come backup
- CMS SITECONF (Trivial File Catalog)
 - Repository per i file di configurazione del sito, contengono informazioni sul sito CMS direttive I/O, squid.. Versione specifica CINECA che punta al CNAF in output e all'Xrootd Proxy in input
- Più una serie di variabili di ambiente necessarie per la corretta gestione di:
 - Certificati, BIND-PATH per singularity, runtime PATH..

Questo ha richiesto debugging fatto eseguendo glidein wrapper script manualmente sul nodo.

Setup della Factory

CMS ha tre istanze **ITB-Dev, ITB e Prod**. Noi siamo partiti da ITB-Dev per i primi test e ora siamo migrati allo step successivo (ITB) che è il pre-prod. Obiettivo è Prod

- ITB-Dev : usato per la prima integrazione del nuovo CE (avvenuta con successo) e per i test manuali con sleep-job per il fine tuning di parametri:
 - Siccome usiamo *wholenode*, per non sprecare risorse è necessario fine tuning di “required cores” vs “declared cores” etc
 - Required Operating System, Etc...
- ITB: lo stiamo iniziando ad usare per l'esecuzione di job di CMS (tipo HammerCloud)

SAM Tests

L'integrazione standard di un CE in CMS implica strettamente l'esecuzione di SAM Test

Noi non vogliamo eseguire questi test al CINECA e per ora abbiamo “bannato” lcgadmin..

- questa non è la soluzione per lo scenario finale (il CE sarebbe marcato come NON OK)
 - Va capito bene come poter dire che i SAM che girano al CNAF non devono andare su quel CE, è possibile farlo in modo “pulito”... work in progress
 - A lungo termine la validazione andrà fatta anche per questo tipo di risorse ma questo è un discorso CMS

Gestione dell'I/O

Il setup prevede che l'output generato dai jobs in esecuzione al CINECA venga scritto al T1-CNAF : **storm-fe.cr.cnaf.infn.it**

E che l'input venga letto dalla proxy-cache di XRootD: **ds-110.cr.cnaf.infn.it**

Al momento questo NON funziona perchè c'è un problema di MTU tra CNAF e CINECA

- **Risolto MTU tutto il giro è funzionante**
- **Per esempio per input da CERN (che non ha 9000) funziona già' tutto**

Prossimi passi (breve termine)

Dobbiamo gestire il “*penalty factor*” di KNL per evitare che tutti i jobs di CMS muoiano prima della fine dell’esecuzione

- Questo si fa tunando opportunamente i parametri di match-making (ClassAD) anche se altre esperienze (NERSC) ci dicono che può non essere sufficiente

- Un errore di pochi % (10-20) nel jobSplitting puo’ far si che si vada oltre le 48 ore

Dobbiamo tunare bene, a livello factory, la gestione del wholenode. Sfruttare o no l’HyperTreading dipende dalla RAM...

- Se dichiariamo e.g 250 cores ma li usiamo e.g 100.. La factory abbassa la pressione e non riceviamo più jobs

In sintesi

- L'integrazione con CMS è a buon punto
- Risolto il problema MTU siamo in grado di confermare che i test end-to-end con job CMS di HammerCloud funzionano correttamente
- Prima di passare ai job di produzione dobbiamo
 - Chiarire come gestire opportunamente i SAM tests (problema + politico che pratico...)
 - Finalizzare il setup della factory per la gestione migliore del wholenode
 - Capire bene come comportarci con il penalty factor di KNL
- **Nessuno di questi problemi pare insormontabile al momento ...**