

CNAF Reloaded Task 2.4

HTC-HPC Opportunistico

Stato e Prossimi passi

Stefano Dal Pra (CNAF), Daniele Spiga (INFN-PG)

17.06.2022

Crediti e contributi : Tommaso Boccali, Lorenzo Rinaldi, ...



Task2.4: Contesto

Oltre alle risorse condivise da Leonardo che saranno utilizzate come pledge CNAF si prevede di poter accedere ulteriori risorse (Leonardo) in una modalità opportunistica

- I.e. utilizzare ulteriori Nodi @ Leonardo quando disponibili; risorse fornite al CNAF via grant o accordi di tipo diverso

Molte delle soluzioni tecniche sviluppate per l'integrazione di risorse pledged e opportunistiche saranno sinergiche, tuttavia ci aspettiamo di dover gestire specificità

- **Policy di utilizzo delle risorse** (QoS potrebbe essere diverso tra pledged e non)
- **HW** (la partizioni opportunistica avrà GPU rispetto alla pledged)
- **Network** (speriamo sia il più possibile uguale alle condizioni dei nodi pledged)
- **Accesso allo storage** (come sopra ma non è ovvio... quindi ci prepariamo al “peggio”)
- **Runtime environment** (prevediamo di avere meno gradi di libertà rispetto alla configurazione dei nodi pledged)
- ...



Background e nuove opportunità (di test)

Non è ancora stata totalmente definita la strategia di integrazione delle risorse opportunistiche e dipenderà dalle condizioni al contorno ancora in fase di definizione. Però **abbiamo iniziato a fare del lavoro di integrazione con il batch per effettuare test di fattibilità** (vedi dopo):

NOTA:

- Ci basiamo sulle (tante) lezioni imparate durante l'integrazione di Marconi A2 (Knl)
 - Grant LHC xxxx
- Stiamo sfruttando l'opportunità di usare M100 per studiare la configurazione del batch, del runtime environment etc.
 - M100 tra l'altro è una "bestia" piuttosto complicata e questo ci aiuta ad identificare possibili problemi
 - Ottima opportunità per provare (vedi dopo)



Federazione di risorse opportunistiche @ Tier1

Lo scenario che vogliamo abilitare è quello di una **federazione di risorse opportunistiche con cui il batch del Tier1 si estende trasparentemente all'utente e/o esperimento.**

- Stefano “controlla” le risorse che si federano al pool estendendolo dinamicamente
- L'esperimento accede alle risorse opportunistiche attraverso lo stesso Gateway (CE) con cui accede alle risorse pledged
 - Eventualmente, in prospettiva, l'esperimento/l'utente può definire quando/se usarle attraverso definizione di ClassAdd. Per questo è necessario definire una convenzione (o standard)
 - Questo ha particolarmente senso, per esempio, se le risorse hanno HW specializzato (i.e. GPUs)



Dove siamo e cosa stiamo facendo

Abbiamo esteso il pool HTCondor del Tier1 ai nodi di M100 rendendoli accessibili attraverso uno dei CE di produzione.

- Modello scatola cinese ovvero da una parte si fa provisioning di job slot (worker node) in modo indipendente
 - vengono sottomessi job via slurm al Cineca.
- Dall'altra vengono incluse le slot nel condor pool del Tier1 (in modo authN/Z)
 - I job slurm avviano demoni startd di HTCondor che sono pre-configurati per autenticarsi con il pool HTCondor del Tier1.

Abbiamo abilitato la possibilità di sottomettere ed eseguire jobs su M100 sottomettendo ai CE di produzione del Tier1

- Attualmente ci sono delle route definite per “fare match” ma questo è uno dei gradi di libertà che abbiamo per gestire le esigenze di utenti/esperimenti



Dove siamo... il runtime a M100

Riprendendo quanto fatto nell'integrazione CMS, abbiamo preparato qualche script per gestire l'ambiente del WN di M100 e garantire la compatibilità coi WN grid

- L'ambiente è gestito attraverso immagine singularity (slc7) e CVMFS.
 - Singularity lo usiamo per garantire voms e gfal/xrootd etc su WN
 - Check in parallelo per rimuovere l'esigenza di girare dentro singularity. Questo è importante perchè facilita l'uso di singularity per l'esecuzione del payload
- HTCondor build per PPC

Primi test a vario livello (ma ovviamente non scale test) sono stati fatti da ATLAS e CMS.

- Tests e feedback sono essenziali!



Accesso alle GPU

Seppure in maniera preliminare **abbiamo verificato che riusciamo ad accedere alle GPU** anche attraverso la catena descritta

- Test con eseguibili “Hello World” e con workflow di CMS (HLT)
- Riusciamo a gestire l’accesso alle GPU da singularity

Nessun problema tecnico noto

Anche qui → Test necessari e benvenuti

```
(base) [dspiga00@r224n05 ~]$ nvidia-smi
Fri Apr 22 11:41:30 2022
-----
NVIDIA-SMI 450.51.06      Driver Version: 450.51.06      CUDA Version: 11.0
-----+-----
GPU   Name               Persistence-M  Bus-Id        Disp.A    Volatile Uncorr. ECC
Fan   Temp    Perf    Pwr:Usage/Cap      Memory-Usage  GPU-Util  Compute M.
                                           Memory-Usage  GPU-Util  Compute M.
-----+-----+-----+-----+-----+-----+-----+-----
  0   Tesla V100-SXM2...  On           00000004:04:00:0  Off          1%      Default
N/A   38C    P0     57W / 300W        1253MiB / 16160MiB                          MIG M.
                                           N/A
-----+-----+-----+-----+-----+-----+-----
  1   Tesla V100-SXM2...  On           00000004:05:00:0  Off          9%      Default
N/A   41C    P0     60W / 300W        1271MiB / 16160MiB                          MIG M.
                                           N/A
-----+-----+-----+-----+-----+-----+-----
  2   Tesla V100-SXM2...  On           00000035:03:00:0  Off          0%      Default
N/A   35C    P0     56W / 300W        305MiB / 16160MiB                           MIG M.
                                           N/A
-----+-----+-----+-----+-----+-----+-----
  3   Tesla V100-SXM2...  On           00000035:04:00:0  Off         10%      Default
N/A   42C    P0     61W / 300W        1261MiB / 16160MiB                          MIG M.
                                           N/A
-----+-----+-----+-----+-----+-----+-----

Processes:
GPU   CI   CI   PID  Type  Process name      GPU Memory
ID   ID   ID                                Usage
-----+-----+-----+-----+-----+-----+-----
  0   N/A N/A   56840  C    cmsRun            1251MiB
  1   N/A N/A   58058  C    cmsRun            1269MiB
  2   N/A N/A   59841  C    cmsRun            303MiB
  3   N/A N/A   53622  C    cmsRun            1259MiB
-----+-----+-----+-----+-----+-----+-----
```



NOTA

Quello che stiamo facendo adesso è sfruttare M100 come “testbed” in preparazione all’integrazione di Leonardo. Quindi:

- Le “limitazioni” di Power9 sono relative a M100 e non Leonardo
- Le “limitazioni” di networking di M100 non saranno così stringenti anche con Leonardo opportunistico
- L’accesso allo storage dai WN di Leonardo opportunistico ci aspettiamo che potrebbe essere più comodo di come è oggi M100
- D’altro canto però
 - l’accesso alle GPUs di M100 ci serve per capire come meglio gestire risorse (massive) di questo tipo con Leonardo
 - L’uso dei nodi slurm per estendere il condor pool del Tier1 sarà sostanzialmente lo stesso

Al netto di queste considerazioni **è molto utile fare test lato esperimento così con i feedback si può disegnare/configurare l’integrazione al meglio** e questo ci interessa perchè



M100 continuerà ad essere disponibile per l’INFN fino a decommissioning (?!?)
Oltre il pledge, già nel 2023 è previsto che ci saranno ore di “opportunistico” su Leonardo



Prossime top - tre priorità

Integrazione del **sistema che genera pressione sul batch slurm**

- I.e. in modo tale che prenda in considerazione la pressione sullo schedd del pool del T1.

Fine tuning dei meccanismi di match-making e/o routing

- Qui è importante acquisire esperienza e informazioni con gli esperimenti per capire quali sono i requirements

Definizione delle policies di accesso / utilizzo dei nodi opportunistici di Leonardo

- Quanto durano i lease
- Quanto possiamo customizzare /configurare i WN ? (WN “general purpose” vs “experiment specific”)
- Accesso storage, Networking