# Considerazioni per 2.1.1


Rimini Sep 14–16 2022

*Email:* dalpra@infn.it

## Premesse

- HTCondor fa il "batch system", *whatever resources are available*. Quindi la valutazione si riduce a vedere cosa cambia (piú o meno semplice, piú o meno efficiente) a seconda di come si presentano le risorse di calcolo passate a HTC.
- WLCG/HEP: "flat budget" e "potenza crescente".
- Rischio "potenza scarsa" $\longrightarrow$ Goal:
- Massimizzare HS06/KW erogati a parità di HW.

## Confronto performances in letteratura

- Bare-metal, virtual machines and containers in OpenStack (IEEE, Mar. 2017)
  **«As expected, bare-metal outperforms all other options in our tests. As such it is recommended for high intensity workloads»**

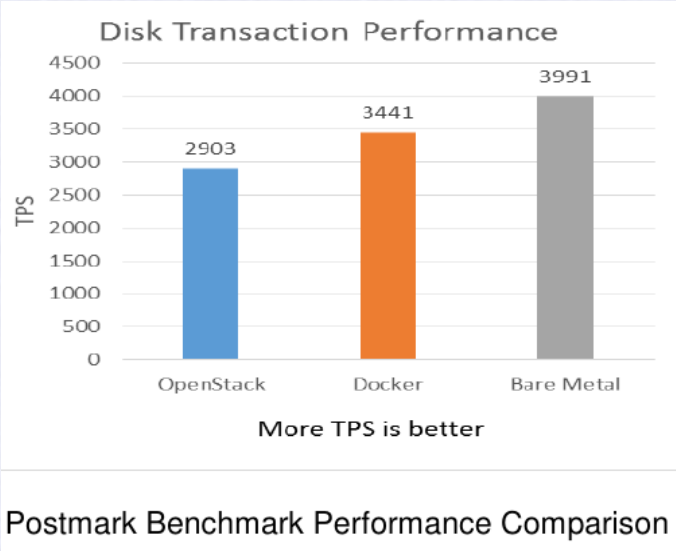
- Performance overhead comparison between hypervisor and container based virtualization (IEEE, Ago. 2017)

  «our evaluation result largely confirms: The container's average performance is generally better than the VM»

- An Empirical Performance Evaluation of Docker Container, Openstack Virtual Machine and Bare Metal Server (IJEECS, Lug. 2017)

  «Docker containers exceed or equal the performance of Openstack KVM virtual instance in every test»

# Alcuni estratti...




Postmark Benchmark Performance Comparison

| Resource | | 8 vCPU | |
|---|---|---|---|
| | time [s] | std | % |
| BM. | 109.5 | 1.0 | -16% |
| Comp. H | 131.4 | 2.1 | 0% |
| Docker | 131.2 | 1.4 | 0% |
| VM | 141.1 | 1.0 | +7% |

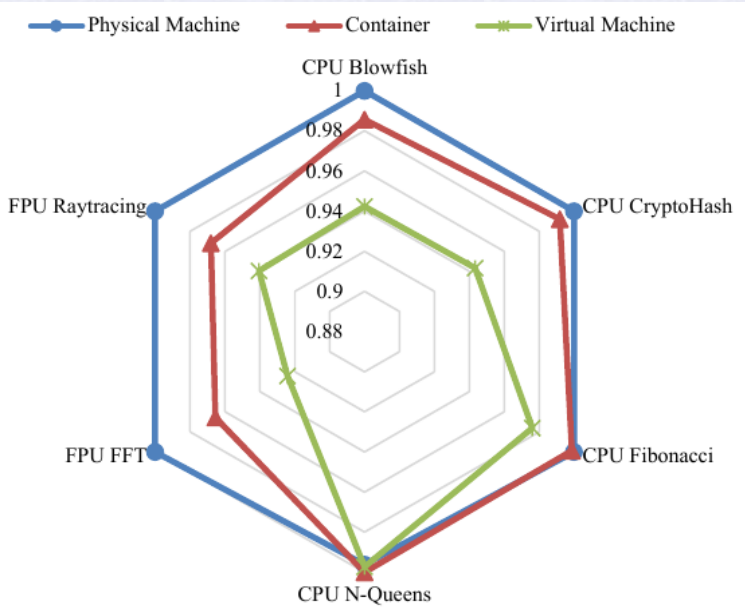



Figure 3. Computation benchmarking results by using HardInfo.

## Confronto HS06 Bare Metal vs Container

Un WN (AMD EPYC 7282 16-Core) misura $14$HS06/core.

Lo "overhead di paravirtualizzazione" costa $27.6$HS06 ($3.2\%$, cioè 2 core).

| **WN 64 core** | BM | docker | $2 \times$ docker |
|---|---|---|---|
| HS06 | $891.25$ | $863.64$ | $438.84 \times 2$ |
| HS06/core | $\sim 14$ | $13.5$ | $13.7$ |
| Diff. | - | $-27.6$ | $-13.6$ |

## Punti

- WN come container su Bare Metal: OK. **I vantaggi compensano il calo HS06?**

- Un container per CPU fisica migliora HS06 complessivo (ma raddoppia numero WN visti da HTCondor)
- BM Provisioning via Openstack: Ironic. **Come interagisce con foreman? Overlap?**

## Todo/Altri punti

- Eseguire un mix di "job reali", non di un unico exp
- testare management container (immagini) via puppet
- Si aggiunge un layer di management: **la variazione di complessità è vantaggiosa?**

| Prima | Dopo |
|---|---|
| HTC | HTC |
| - | container management |
| WN management | BM management |
| BM WN provisioning | BM provisioning |

## Osservazioni su passi del TDR

- 6.2.1.1.1 «In order to enable seamless and uniform access to the computing resources of both partitions through a common access point, containers will be adopted.» In che senso "uniform access? Da parte di chi?" L'accesso alle risorse batch rimane "seamless" e "uniform" anche se si fa provisioning di risorse BM (via Ironic)

- 6.2.1.1.1.1 «single access point to all the HTC/HPC facilities». Attualmente non è piú possibile usare uno stesso HTC-CE per mandare job a Slurm e HTCondor. Possiamo insistere con i developer ma non sembra tra i loro main interest.

## Piano per migrazione HTCondor pool

– Senza interruzione di servizio

– Alla fine un transitorio (1 o 2 giorni) eventualmente con servizi accessori non disponibili (es. dashb monitoring/accounting)

**Prima di cominciare si assume che**

- rete, storage, provisioning (puppet/foreman) già disponibili
- coesistenza in prod. di T1 e TP

**Durante transitorio:** HTcondor funziona con:

- WN al T1, al TP e al Cineca (come da vari anni).
- No upgrade / riconfigurazioni
- Accounting / monitoring "differiti" (si recuperano a fine tr.)
- Eventuale High Availability limitata (per Central Manager)

**Migrazione pool**

Assumendo networking "trasparente" (come tra T1 e WN del cineca)

- Si attivano WN al T.P. Lavorano con i server al T1

- Si spegne il CM in standby al T1 e lo si riattiva al TP (VM migration)
- Si Spegne un CE al T1 e lo si migra al TP (VM [live] migration). Per CE bare metal: possibile trasferire "lo stato" da CE T1 a CE "clone" al TP.
- Si spegne il CM al T1 e lo si migra al TP (VM [live] migration)

## Il transitorio: SharedFS (aka farm-nas)

HTCondor tiene su shared-FS configurazioni per il pool, script di servizio (pochi MB) e joblog di accounting (diversi GB).

- Si copiano le conf e gli script localmente a tutti i nodi del cluster: fatto questo si puo' smontare lo shared fs.
- I dati di accounting si accumulano localmente al WN
- si migrano i dati di farm-nas (o si migra farm-nas)
- si riattiva lo shared fs.

## Accounting DB (dbfarm)

- si spegne dbfarm di standby al T1
- si installa dbfarm al TP, si fa la replica (`pg_basebackup`) si attiva replica al TP
- si invertono ruoli (Master al TP, replica al T1)
- si installa secondo dbfarm al TP, lo si attiva come replica.