

INFN-CLOUD as an analysis facility

Meeting annuale ATLAS Italia Computing - Milano

30.XI – 01.XII 2022

Caterina Marcon – INFN Milano



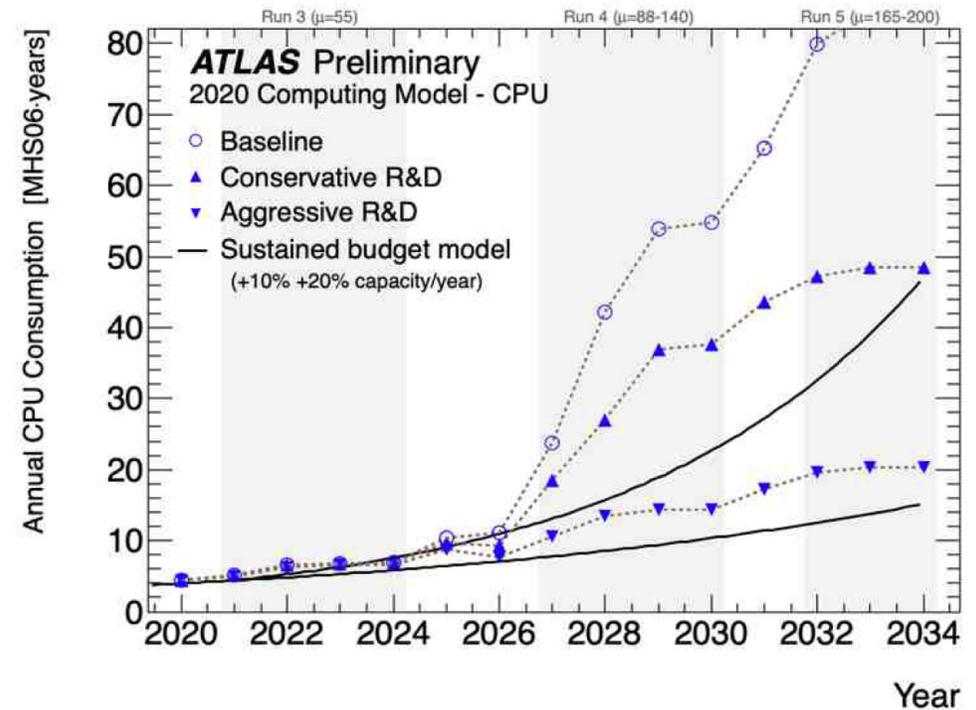
Outline

- Utilizzare risorse di Cloud Computing per ottimizzare il workflow di analisi dati di ATLAS;
- INFN CLOUD infrastructure;
- Stato di avanzamento del progetto:
 - Utenti & workflow di analisi;
 - Un ambiente **containerizzato** come possibile soluzione;
- Verso un progetto "ATLAS" in collaborazione con INFN CLOUD.

Utilizzare risorse di Cloud Computing per
ottimizzare
il workflow di analisi dati di ATLAS

Elaborazione e produzione dei dati in ATLAS

- Gli esperimenti a LHC utilizzano un'**infrastruttura di calcolo complessa e distribuita**:
 - quasi 1 milione di core di calcolo;
 - un exabyte di storage interconnessi tramite reti ad alta velocità.
- Attualmente, la maggior parte dell'elaborazione e produzione dei dati per l'utente finale viene eseguita attraverso il **Worldwide LHC Computing Grid (WLCG)** costituita da centinaia di singoli siti in tutto il mondo presso università e laboratori nazionali;
- **Le esigenze di calcolo di ATLAS nell'era HL-LHC** (a partire dal 2027) **non saranno soddisfatte** dall'infrastruttura esistente (anche tenendo conto dell'evoluzione prevista della tecnologia hardware).



Elaborazione e produzione dei dati in ATLAS

- Si sta quindi **esplorando l'uso di nuove tecnologie** e nuove strutture informatiche che giocheranno un ruolo chiave nell'affrontare il problema della carenza di risorse;
- Fino ad ora, diverse **risorse di calcolo alternative** (HPC, cloud computing e altre risorse opportunistiche) **sono state integrate** nel sistema di calcolo distribuito;
- Siccome queste risorse sono volatili, ATLAS ha utilizzato tali risorse **quasi esclusivamente per scopi di simulazione Monte Carlo** evitando carico associato a workflow di analisi o altri flussi di lavoro più complessi;
- Attualmente studi/**progetti in corso in ATLAS per utilizzare le risorse di cloud privati** (Amazon e Google) -> la sfida principale del cloud computing è l'accesso e la gestione dei dati.

Il progetto

E' indetto un concorso pubblico per **titoli ed esame colloquio** a n. 1 assegno Junior Fascia 1 per la collaborazione ad attività di ricerca scientifica, da usufruire presso la Sezione di Milano dell'I.N.F.N. sul seguente tema di ricerca:

“ Ottimizzazione del workflow di analisi dati dell’esperimento ATLAS su risorse di cloud computing. ”

- Pur non specificato esplicitamente dal bando, è venuto “naturale” pensare di sfruttare le risorse messe a disposizione da **INFN CLOUD**.

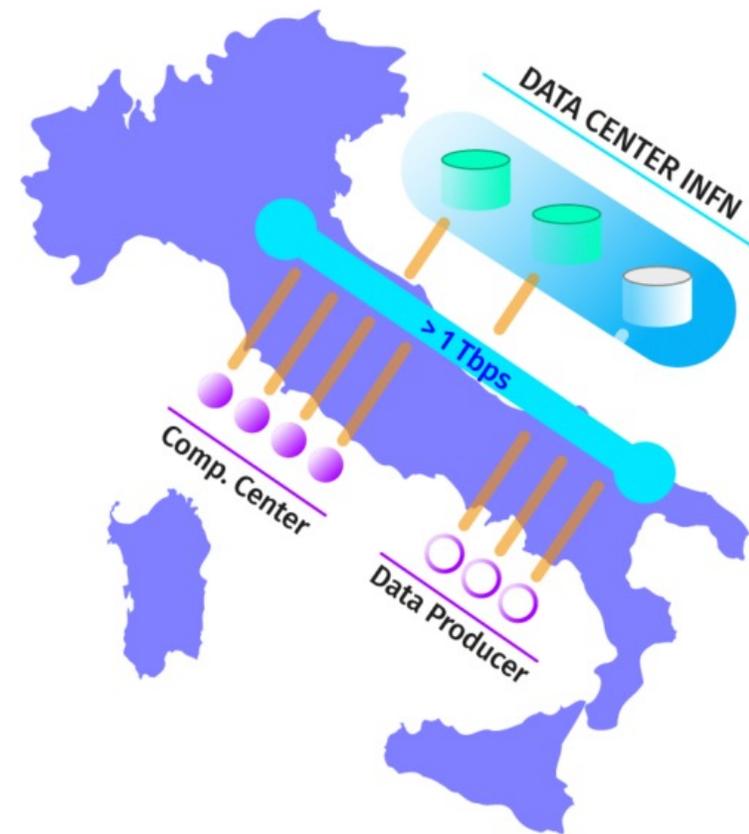
INFN CLOUD infrastructure



INFN Cloud



- Da Marzo 2021, INFN offre ai propri utenti un set di servizi Cloud attraverso la **propria infrastruttura INFN CLOUD dedicata**;
- L'infrastruttura INFN CLOUD si basa su un **core backbone** che collega i grandi data center di **CNAF e Bari**, e su diversi siti federati che si collegano al backbone;
- Le risorse sono orchestrate da **OpenStack**;
- Tutti i siti che offrono risorse sono federati e disponibili agli utenti INFN;
- A seguito della nomina di "amministratore", un utente può, a sua volta, erogare sotto-servizi.
- La tipologia di servizi offerti si colloca nei paradigmi **PaaS e SaaS** (Platform e Software as-a-service, rispettivamente)





INFN Cloud



- L'accesso ai servizi cloud avviene per mezzo di un **pannello di controllo che raccoglie alcune funzionalità templatizzate** -> ad es. per creare delle VMs;
- I template predisposti sono conformi allo standard open TOSCA;
- Da dashboard, i template che si possono richiedere sono diversificati ma comunque limitati (ad es. le VMs riservate possono avere 4 vCPU, 8 GB di RAM e 100+30 GB di archiviazione).

Virtual machine



Docker-compose



Run docker



Elasticsearch and Kibana



Kubernetes cluster



Spark + Jupyter cluster



Jupyter with persist
Notebooks

Sync&Share aaS

Avanzamento del progetto

Un progetto *in fieri*

Individuare gli aspetti basilari per lo sviluppo di un'unità di base

Sviluppo di un'unità di base

Individuare altri aspetti caratterizzanti

Prototipo di prodotto funzionante

Prodotto finito + Utenti

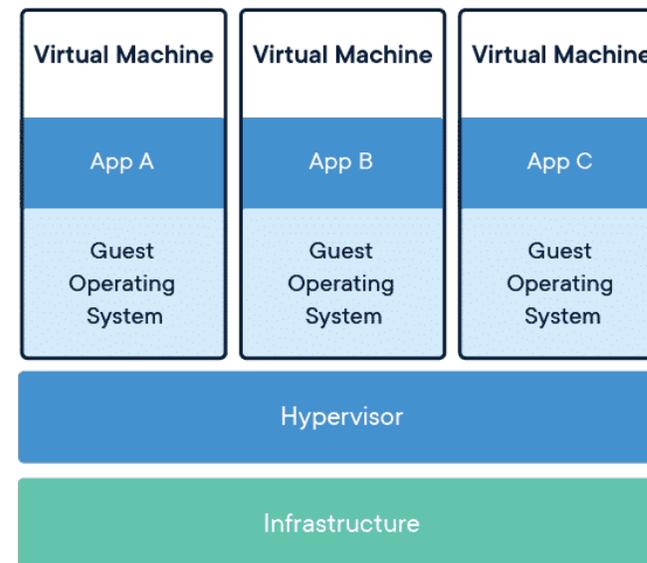
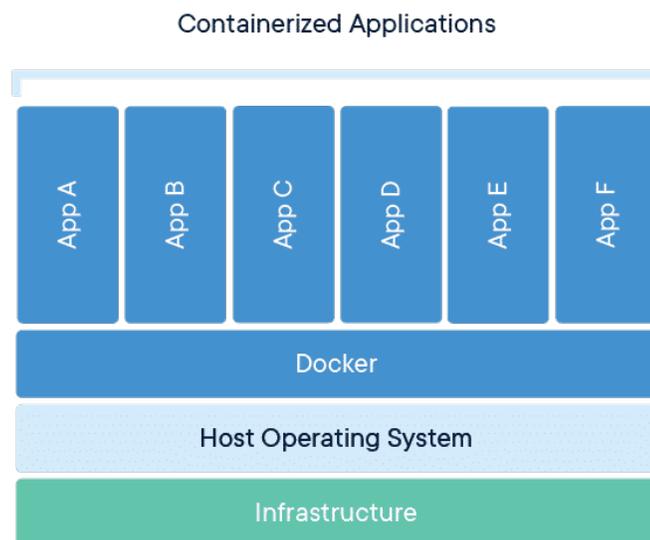


Utenti & workflow di analisi

- Per delineare un primo **set di caratteristiche da implementare**, si è ipotizzato, in questa fase iniziale, per non perdere in generalità, di considerare come utenti “in potenza”:
 - chiunque sia dotato di un account CERN;
 - chiunque abbia un’affiliazione all’esperimento.
- In ATLAS convivono **workflow di analisi eterogenei** -> non si sono posti, al momento, particolari constraints sul tipo di workflow di analisi da sviluppare;
- Si sono individuati due principali approcci (**non ortogonali**) con cui strutturare l’implementazione:
 - **Approccio batch like** (mira a gestire/trasformare ntuple di dati ~GB o decine di GB);
 - **Approccio interattivo** (come ad esempio generazione di plot, tabelle, systematics and limit testing, neural network training).

Un ambiente containerizzato come possibile soluzione

- Per ottenere tale flessibilità si è pensato di usare il **container come unità "atomica"** per lo sviluppo;
- Un container è un'**unità applicativa isolata**;
- A differenza di una macchina virtuale, **non porta con sé indicazioni relative all'hardware** su cui è in esecuzione;
- Il container "**vede**" **tutte le risorse disponibili sulla macchina host** e, in caso di container multipli, le condivide (mantenendo la segregazione logica dei dati)



Un ambiente containerizzato come possibile soluzione

- Per tale scopo si è pensata una soluzione **Platform-as-a-Service (PaaS)**, basata su un'immagine **Docker Centos 7**;
- L'immagine è responsabile del **provisioning dei filesystem condivisi CVMFS ed EOS del CERN**, da cui è possibile caricare un ambiente ATLAS standardizzato;
- L'unica responsabilità dell'utente finale è quella di fornire un'applicazione funzionante in grado di **recuperare e analizzare i dati e di esportare i risultati in una memoria persistente**;
- Il codice di analisi può essere scaricato da EOS o **da repository git remoti**.

```
lsetup
lsetup asetup
lsetup astyle
lsetup atlantis
lsetup eiclient
lsetup emi
lsetup ganga
lsetup lcgenv
lsetup panda
lsetup pyami
lsetup root
lsetup rucio
lsetup views
lsetup xcache
lsetup xrootd
advancedTools
diagnostics
helpMe
printMenu
showVersions
lsetup <tool1> [ <tool2> ...] (see lsetup -h):
(or asetup) to setup an Athena release
ATLAS style macros
Atlantis: event display
Event Index
EMI: grid middleware user interface
Ganga: job definition and management client
lcgenv: setup tools from cvmfs SFT repository
Panda: Production ANd Distributed Analysis
pyAMI: ATLAS Metadata Interface python client
ROOT data processing framework
distributed data management system client
Set up a full LCG release
XRootD local proxy cache
XRootD data access
advanced tools menu
diagnostic tools menu
more help
show this menu
show versions of installed software
```

Un ambiente containerizzato come possibile soluzione

- L'immagine atlas-on-demand, è stata pensata per essere il più **customizzabile possibile**: il container che viene avviato con quell'immagine deve essere in grado di adattarsi ad un flusso di lavoro definito dall'utente;
- Per dare questa flessibilità, il container che è stato implementato, all'avvio, recupera **uno script di analisi secondo le indicazioni dell'utente** e lo fa partire;
- Dove recuperare lo script di analisi e quale script fare girare -> sono informazioni che devono essere settate dall'utente per mezzo di **variabili di ambiente** nel file **docker-compose.yml**, indispensabile per l'avvio del container.

Protocolli di autenticazione dell'utente

- Autenticazione ad EOS:
 - Si presuppone che l'utente abbia un account CERN e sia parte della collaborazione ATLAS -> in generale, per accedere ad EOS, è necessario autenticarsi inserendo **nome utente e password** -> questa **autenticazione è sempre interattiva** -> **problema ad inserire la password utente se si deve lanciare in differita.**
- Soluzione implementata e funzionante al momento: **il protocollo Kerberos (basato su protocollo RSA)**
- In aggiunta alla sopracitata autenticazione interattiva user+pass, esso consente l'autenticazione di un utente per mezzo di un file **KeyTab (file binario e dell'ordine di qualche centinaio di byte)** -> inserendo questo file nel container è possibile utilizzare il suo contenuto per portare a termine un'autenticazione asincrona.
 - Questa procedura presuppone che:
 - sia generato un file KeyTab con lxplus [1], [2];
 - sia caricato questo file nel container in fase di avvio (usando una variabile di ambiente nel docker-compose.yml).

Protocolli di autenticazione dell'utente

- Autenticazione ad EOS: possibile strada alternativa (che si sta investigando) -> **utilizzare, come alternativa a Kerberos, i token di EOS;**
- L'autenticazione mediante token, risulta, in generale, in grado di fare delle **autenticazioni più mirate** rispetto Kerberos (permettendo all'utente di autenticarsi per periodi di tempo limitati, su determinate aree circoscritte);
- In accordo con quanto riportato sulla guida [1], **i token di EOS sono proprietà da attribuire a file, cartelle, alberi di cartelle** -> in Potenza può essere necessario un lavoro, a carico dell'utente, di preparazione di questi token;
- I token possono essere rilasciati su cartelle e file di proprietà dell'utente -> **le cartelle di progetto non sono necessariamente di proprietà dell'utente che li usa;**
- L'implementazione di tale metodo è in fase di investigazione; **si riscontrano problemi nella generazione dei token su Ixplus dovuti a permessi mancanti** (anche su file personali).

[1] <https://eos-docs.web.cern.ch/using/tokens.html>

Un cluster containerizzato

- **10 macchine virtuali** su cui sono stati istanziati questi container Docker; tale configurazione è stata creata a scopi di studio e per testare un primo workflow di analisi e caratterizzare le performance dell'ambiente dockerizzato;
- Si sta valutando la possibilità/fattibilità di **collegare queste risorse ad uno storage su INFN Cloud** (in progress);
- Si sta inoltre cercando di capire **che tipo di interfaccia utente adottare** (interfaccia grafica, terminale etc...) (in progress);
- Come step futuro, si prevede di **istanziare HTCondor per gestire la sottomissione dei diversi job**;
- Si deve **valutare la scalabilità** di questa configurazione;
- E' una configurazione che ci ha permesso e ci permetterà di socializzare con le principali problematiche che saranno comuni anche ad implementazioni future **MA NON è un'implementazione cloud native.**

Una prospettiva Cloud-native

Verso un'implementazione (più) Cloud-native

Requisiti individuati per una futura implementazione:

- **Ambiente containerizzato** che offre una suite CERN di base (CVMFS + EOS);
- In grado di soddisfare due principali workflow (**non ortogonali**): **batch like ed interattivo**;
- **Architettura scalabile** in base al carico/numero di utenti che usano tali risorse
- Raggiungere la massima autonomia/indipendenza tra:
 - Chi scrive l'analisi;
 - Chi mantiene il container;
 - Chi mantiene l'infrastruttura.

Verso un'implementazione (più) Cloud native

- Implementazione definitiva conforme ad una **filosofia "on-demand" e "pay-per use"**:
 - Il cloud offre **risorse fluide**, che possono essere allocate e liberate con facilità
 - Il cloud consente un controllo mirato dei costi di utilizzo, perché **l'allocazione delle risorse è limitata al walltime** del processo
 - Un'allocazione dinamica delle risorse **slega l'applicazione dal layout dell'architettura**
- Progetto in collaborazione con INFN CLOUD:
 - **Andare oltre ai template** offerti dalla dashboard ed interagire direttamente con gli orchestratori (OpenStack e/o equivalenti)
 - **Modellare le risorse richieste** sulla base del particolare workload; e.g. allocare molte CPU e meno RAM per lavori di calcolo puro; allocare molto storage per flussi con grandi file di dati; allocare molta RAM per analisi di eventi molto complessi, etc.
 - Ad oggi, questa flessibilità non sembra disponibile. L'obiettivo è un'**interazione con INFN CLOUD** volta a realizzare tutto ciò, anche sulla base dei progetti già in essere.

Conclusioni

Conclusioni

- Progetto finalizzato a sviluppare un **modello di workflow di analisi** per l'esperimento ATLAS in grado di sfruttare le **risorse di INFN CLOUD**;
- Sono stati isolati due use cases: **batch-like** e **interattivo**;
- Fase iniziale basata su un approccio batch-like, i.e. sottomissione di job ad un orchestratore;
- Alcuni **requisiti fondamentali** sono stati **individuati e implementati** in un'unità atomica basata su container;
- Specialmente per le applicazioni interattive, è necessario **allontanarsi dal paradigma del calcolo su cluster** in favore di soluzioni cloud-native;
- Data la complessità del tema (architetture, software, gestione dati, gestione utenti, etc.), si ha l'obiettivo di **creare un dialogo ATLAS-INFN CLOUD** per un progetto condiviso.

