Esperienza con uso di PROOF

Dario Berzano (ALICE, Subatech Nantes/TO),
Stefano Bagnasco (ALICE, TO), **Umberto De Sanctis** (ATLAS, TS/UD),
Piergiulio Lenzi (CMS, FI), **Filippo Pascolo** (ATLAS, TS/UD),
Roberto Vitillo (ATLAS, PI)

Workshop CCR-INFN Grid
Acireale, 20 maggio 2010

L'analisi interattiva

Se si vuole eseguire un job sulla Grid, esso viene messo in una coda ed eseguito non appena si trovano risorse libere ⇒ analisi batch

Ci sono però calcoli che non sono adatti ad essere eseguiti sulla Grid:

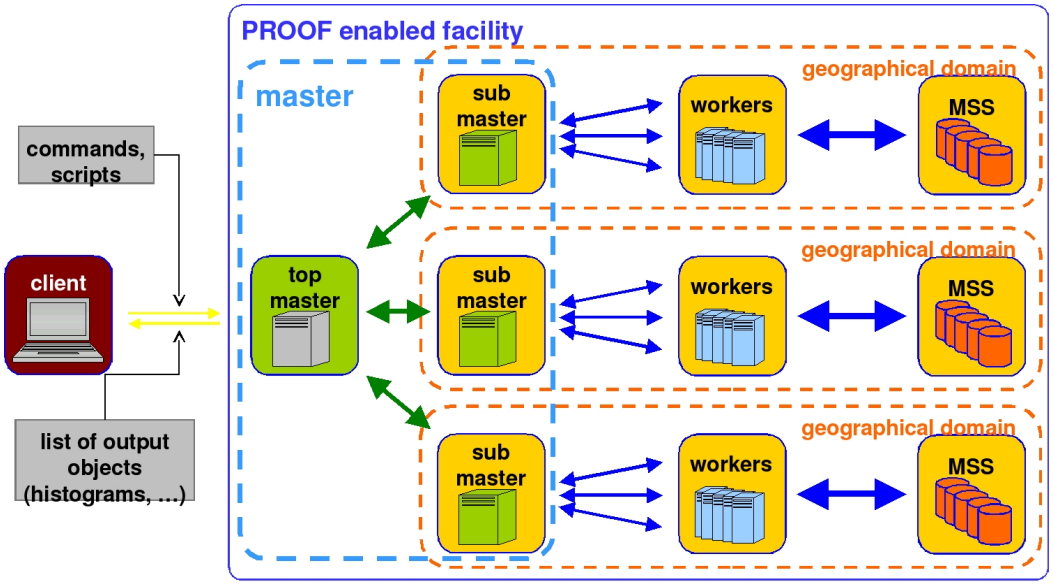
- Ottimizzazione di una grande produzione Grid prima dell'invio
  ⇒ *Tuning di tagli e calibrazioni, debug su campioni di dati*
- Analisi la cui durata è breve
  ⇒ *Tempo di attesa in coda paragonabile alla durata dell'analisi*

PROOF, the Parallel ROOT Facility: strumento di analisi parallela ed interattiva basato su ROOT

Si è consolidato come punto di riferimento:

- ROOT-based ⇒ minimizza l'adattamento del codice esistente
- Parallelismo ⇒ è in grado di sfruttare le caratteristiche sia delle macchine multi-core che del calcolo distribuito su piccola e larga scala
- Interattività ⇒ niente code, risorse subito disponibili

Analysis facility PROOF



- L'utente si collega al top master lanciando una macro di analisi su una lista di file o un dataset
- Ogni worker carica la macro di analisi e le librerie dell'utente
- Ogni submaster sa su quali nodi sono i file da analizzare ⇒ analisi "vicino" ai nodi per ridurre il traffico di rete[†]
- Ogni worker invia i risultati al proprio master che si occupa del merging
- Il top master invia il risultato finale al client (parziale se qualche worker è crashato o se qualche dato non era disponibile)

[†] G. Ganis, J. Iwaszkiewicz and F. Rademakers, *PoS* **ACAT** (2007) 022.

PROOF-Lite

PROOF-Lite è un sistema che consente immediatamente di utilizzare ROOT in parallelo sul proprio sistema desktop multicore

```
root[0] TProof::Open("")
```

Vantaggi

- Configurazione zero: niente demoni, storage, sysadmin
- Ambiente di lavoro identico ad una facility "vera"
- Ideale per test

Limiti

- Almeno 8 core per risultati migliori
- Adatto solo per test ed uso personale
- Il calcolo diventa disk-bound

I test di fattibilità presentati da ATLAS e CMS si basano largamente sull'uso di PROOF-Lite

Event Data Model in CMS

- L'Event Data Model di CMS prevede la scrittura di file ROOT contenenti per ogni evento varie collezioni di oggetti ricostruiti
- L'utente può analizzare gli eventi utilizzando:
  - il full framework ⇒ in locale e su Grid
  - il framework leggero FWLite ⇒ solo in locale
- CMS ha sviluppato dei formati di dati molto compatti e facili da usare per l'analisi ⇒ PAT, Physics Analysis Toolkit

Gli utenti tradizionalmente preferiscono ridurre l'informazione contenuta negli eventi ricostruiti in un plain ROOT tree, questo ha degli svantaggi

Problema

- Si perde la provenance ⇒ set di parametri usati per ricostruire gli eventi e per produrre il tree
- Nessuna informazione aggiunta, si copia soltanto

Perché usare PAT

- Dimensioni comparabili ma interfacce user-friendly
- Dati pubblicabili sul Data Bookkeeping System ⇒ resi disponibili alla collaborazione

Fattibilità di PROOF in FWLite

CMS non dispone di disposizioni comuni chiare per l'utilizzo di PROOF, più un problema di organizzazione che di risorse ⇒ solo iniziative locali

Test INFN Firenze[†]

- Analisi finale del canale Z+jets
- PROOF-Lite su una macchina 8 core da 2 GHz l'uno
- Dati di input: $4 \div 10$ kB per evento in formato PAT
- Tempo: un'ora circa per 20 milioni di eventi
- Storage: dati copiati localmente

I risultati dei test sul framework sono decisamente positivi:

- FWLite perfettamente compatibile con PROOF
- Formato dati PAT direttamente usabile senza convertire i dati in ROOT plain ⇒ mantenuta la provenance

[†] Studio di Piergiulio Lenzi

Attività PROOF in ATLAS

Lo studio preliminare dell'adozione di PROOF in ATLAS rientra nel lavoro di definizione di un modello italiano per i Tier-3

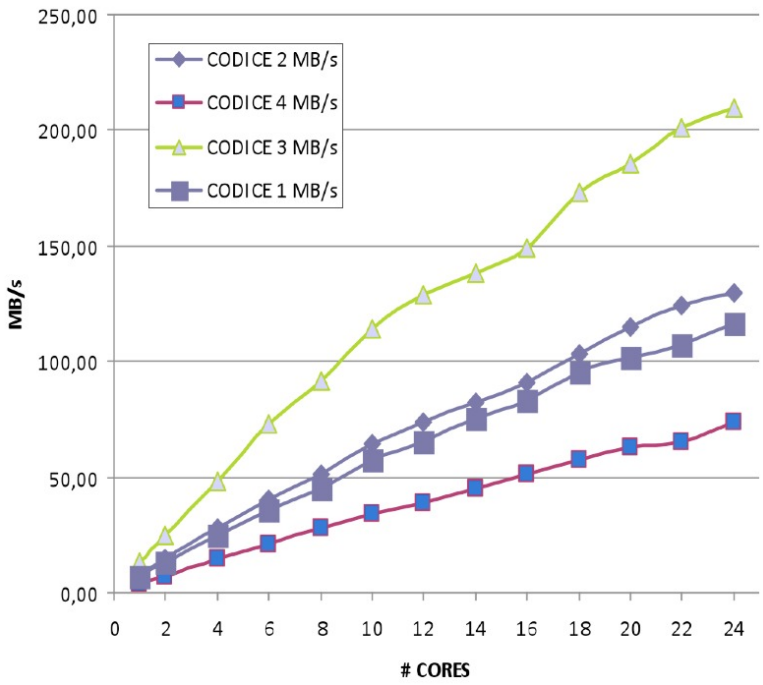
Premesse dell'attività PROOF:

- PROOF sarà usato per l'analisi finale nei Tier-3
- Il formato D3PD usato nelle analisi finali è un formato ROOT
- I D3PD sono prodotti dal framework Athena nei Tier-2
- Dati copiati manualmente nello SE locale prima dell'analisi su PROOF

*Per ogni utente stimati **8 core** e **2 TB** di storage*

- Un nodo PROOF a 8 core assorbe fino a $\approx 100\ MB/s$
  $\Rightarrow$ *porta Ethernet da 1 GB/s saturata*
- Il pool di storage deve fornire dati a molte macchine
  $\Rightarrow$ *infrastruttura interna a 10 GB/s e topologia senza colli di bottiglia*

Risultati dei test preliminari



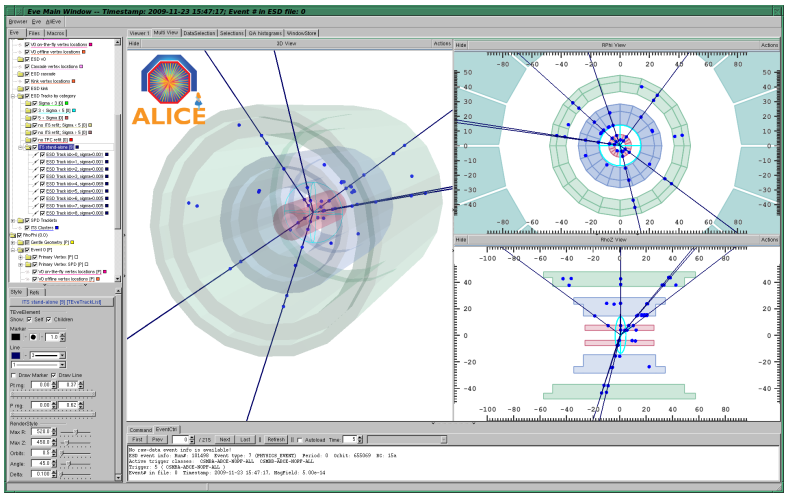
Premesse del test†

- 4 tipi di codice differenti per throughput
- Fino a 8 core
  ⇒ *PROOF-Lite*
- Da 9 a 24 core
  ⇒ *cluster di 3 nodi*

- PROOF scala bene fino a 6 core poi sempre meno ideale
- Per un'analisi di media complessità ⇒ accesso allo storage non limitato né dalla velocità del disco né dalla rete
- Parallelizzazione ⇒ throughput nettamente superiore al single core
- Più l'analisi è CPU-bound, più la scalabilità di PROOF migliora
- Scostamento dall'idealità dovuto alla lettura dei dati ed alle allocazioni di memoria ⇒ fondamentale leggere solo i branch utilizzati

† Test preliminari dell'INFN di Milano, Udine/Trieste e Pisa

CAF e "new CAF"



- A Ginevra: Central/CERN Analysis Facility (CAF)
- L'analisi del primo articolo sui dati presi a 900 GeV è stata interamente eseguita sulla CAF al CERN[†]
- Il framework Aliroot è utilizzato per eseguire sia ricostruzioni, sia analisi finali su PROOF

Per ALICE l'uso di PROOF è una realtà consolidatasi negli ultimi 2 anni

- Dopo la prima CAF, si sta in questi giorni configurando la "new CAF"
- Il core team fornisce linee guida precise per utenti e sysadmin

[†] ALICE Collaboration, First pp collisions at the LHC as observed with the ALICE detector: measurement of the charged particle pseudorapidity density at $\sqrt{s} = 900$ GeV *Eur. Phys. J. C* **65** 1-2 (2010) 111-125

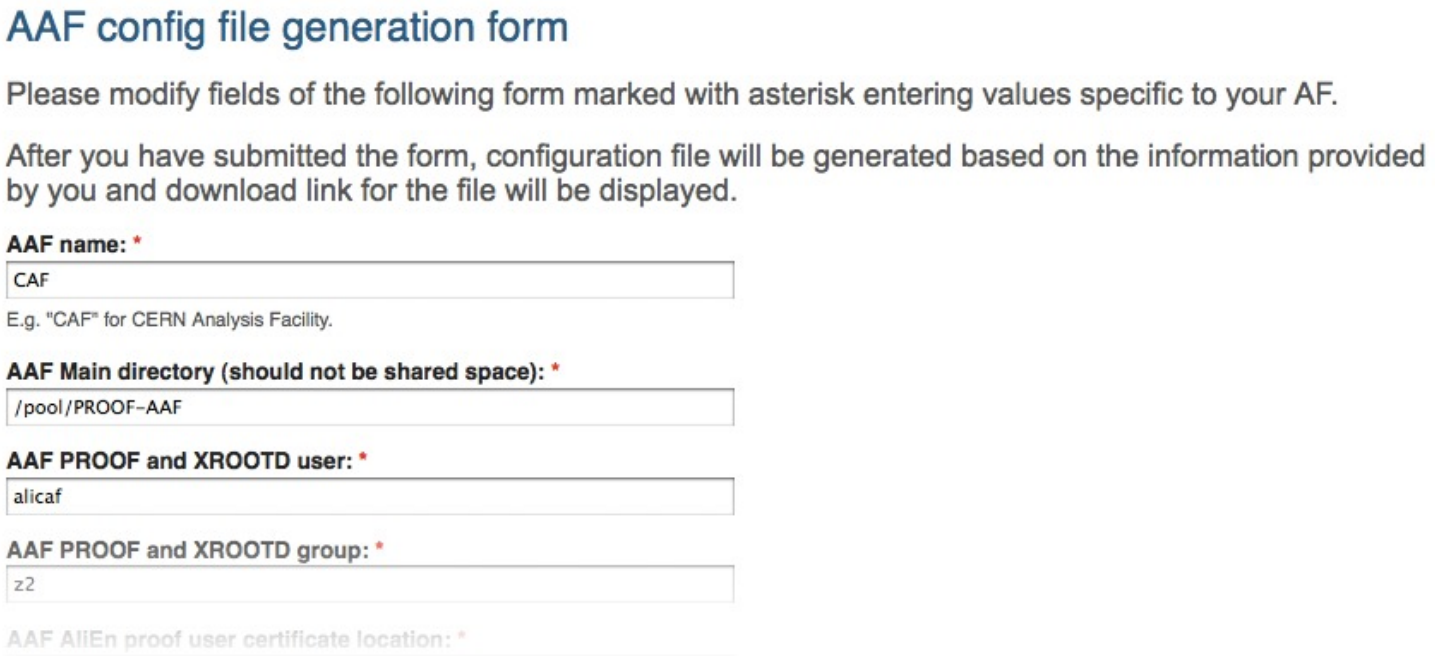
Caratteristiche di una AF ALICE

Configurazione standard di una AF ALICE:

- Di preferenza, nessun accesso allo storage Grid
  ⇒ *ogni facility PROOF ha il suo storage locale ad accesso esclusivo*
- I nodi di calcolo sono anche parte del pool di storage
  ⇒ *PROOF può inviare le analisi vicino ai dati*
- Gestione dei dati attraverso i dataset di PROOF
  ⇒ *se l'utente vuole dei dati, invia una richiesta registrando un dataset*
- Staging dei dati automatico da AliEn
  ⇒ *un sistema automatico stagia i nuovi dati richiesti*
- Upgrade automatico di ROOT e Aliroot su tutte le macchine
  ⇒ *mediante l'utilizzo del PackMan di AliEn su Scientific Linux 5*
- Monitoring della facility attraverso MonALISA
  ⇒ *già installato nei siti per la Grid, plugin PROOF già pronti*
- Ancora nessuna soluzione per il purging dei dati "vecchi"
  ⇒ *work in progress per gestire il purging mediante i dataset*

Deployment della configurazione

Per facilitare la creazione di una analysis facility con queste caratteristiche, ALICE dispone di un installatore con interfaccia web



- Più facile ottenere supporto e unire le forze per lo sviluppo
- Ambiente uniforme per l'utente in tutte le AF del mondo
- Aperta la strada per un PROOF Global Master ⇒ mi redirige automaticamente dove sono i dataset interessanti

[†] Lavoro di Martin Vala ed Arsen Hayrapetyan (http://aaf.cern.ch/node/47)

Disponibilità delle risorse

Considerazioni sulle risorse richieste:

1. PROOF-Lite insufficiente
   $\Rightarrow$ *analysis facilities preferite*
2. Facilities PROOF centralizzate (es. CAF)
   $\Rightarrow$ *bene per iniziare, ma mai su misura per le esigenze locali*
3. PROOF è una facility in più da installare e mantenere
   $\Rightarrow$ *costa in termini di risorse di calcolo e di manutenzione*

La possibilità di disporre delle risorse sufficienti a PROOF e il modo per utilizzarle varia a seconda della dimensione del centro di calcolo

Tier-1

- **CPU: $\approx$ 1k**
- Convertire macchine esistenti in cluster PROOF
- Job draining e PROOF On Demand[†]

Tier-2 e Tier-3

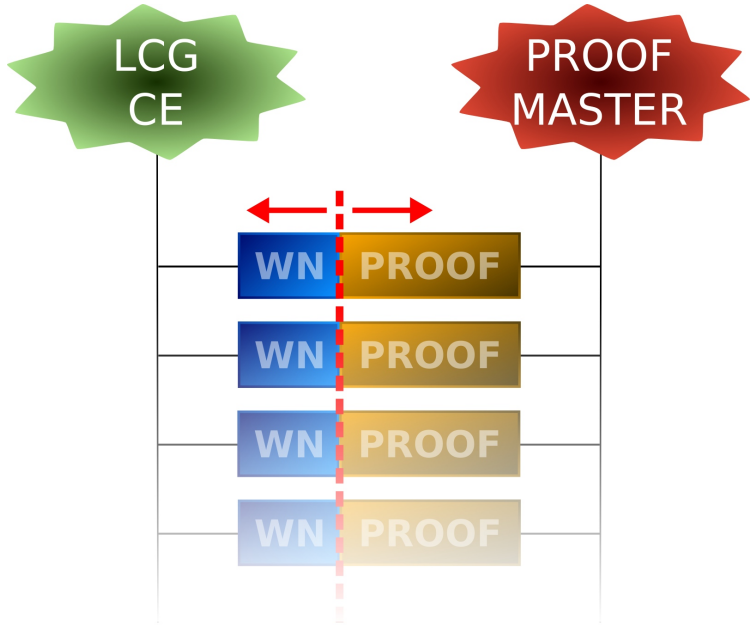
- **CPU: $\approx$ 100, o meno**
- Vi sono i fisici che fanno le analisi finali $\Rightarrow$ richiesta alta
- PoD può non essere troppo reattivo (outlook ATLAS)

[†] Lavoro di Anar Manafov (http://pod.gsi.de/)

Facility PROOF virtuale

INFN Torino: in un Tier-2/3, condivisione delle risorse di calcolo mediante macchine virtuali ⇒ Virtual Analysis Facility[†] basata su Xen

- Nessuna perdita di prestazioni
  ⇒ *per task CPU-bound la perdita è stata misurata nulla con Xen*
- Non è "cloud computing"
  ⇒ *semplice, non c'è deployment, due VM statiche per nodo*
- La Grid esegue job batch che durano parecchie ore
  ⇒ *è sensato rallentarli per poco tempo in favore di PROOF*
- Tempo di risposta rapido
  ⇒ *risorse PROOF disponibili dopo appena ≈ 1.5 min dalla richiesta*
- Isolamento e sandboxing
  ⇒ *crash dei nodi PROOF e Grid rimangono confinati nelle VM*



Facility virtuale dinamica
Ogni macchina fisica contiene sia un nodo Grid che un nodo PROOF, con risorse variabili su richiesta

[†] Lavoro di Dario Berzano

Metodi di storage e di accesso ai dati

Pool PROOF (es. con xrootd)

- I nodi di calcolo hanno i dati
- Maggiore velocità
- Ridotti trasferimenti di rete
- Impossibile facility PROOF dinamica (PoD, VAF)

Storage centralizzato

- Minima perdita di prestazioni
- Bene per facility dinamica
- Reti più veloci richieste
- Uno storage in meno da mantenere

- I pool PROOF contengono mere copie dei dati e non necessitano di ridondanza (RAID, backup): gli storage centrali forniscono i dati per ricostruire il pool in caso di rottura di dischi
- Anche con accesso ai dati dallo storage remoto è caldamente consigliato trasferirli prima sul proprio storage locale
- Soluzione intermedia (outlook ALICE): accesso centralizzato dove il pool funge da cache $\Rightarrow$ possibile con l'interfaccia vMSS (Virtual Mass Storage System) di xrootd
- ATLAS: estremamente più semplice la configurazione se lo storage è POSIX (GPFS, Lustre), difficile l'integrazione con DPM

Gestione dei dataset e staging

PROOF dispone dei dataset, liste di file su cui sono eseguite le analisi contenenti sia gli URL sia metadati (dimensioni, numero di eventi, tree...)

- Ogni utente può registrare i suoi dataset
- La gestione (opzionale) della quota dischi è interna a PROOF
- Meccanismi di staging e purging non interni a PROOF
  $\Rightarrow$ *registrare/rimuovere un dataset non implica copia/rimozione di dati*

afdsmgrd: demone di gestione dei dataset per analysis facilities[†]

- Usa configurazione di PROOF e reagisce ai cambiamenti dei dataset
- Comando personalizzato se un nuovo file è registrato
  $\Rightarrow$ *staging sul pool PROOF, copia dal Tier-2 al proprio Tier-3...*
- Riduce la gestione dello storage alla gestione dei dataset
- In produzione al CERN (nuova CAF), distribuito nell'aaf-installer
- Azioni personalizzabili: non limitato ad ALICE

[†] Lavoro di Dario Berzano (http://afdsmgrd.googlecode.com/)

Conclusioni

Punto della situazione

Esperienze diverse tra esperimenti

- **ALICE** deployment di una configurazione comune
- **CMS** fattibilità del formato PAT su PROOF
- **ATLAS** test dei sistemi di storage

Definizione di modelli comuni

- → Condividere gli sforzi di sviluppo per evitare sovrapposizioni
- → Necessario negli istituti dove più esperimenti convivono
- → Obiettivi: risorse di calcolo, storage

Proposte di lavoro

Test di fattibilità e velocità dei modelli di accesso ai dati

- → Pool xrootd già in produzione con successo
- → Per storage centralizzati: confronto GPFS e Lustre
- → Difficoltà con DPM superabili?

Studio delle soluzioni di condivisione dinamica delle risorse

- → PoD e VAF: soluzioni mature per Tier-2/3?
- → Risposta veloce nell'ottenimento delle risorse?
- → Isolamento efficace dei crash?