# Elastic Extension of Computing Centre Resources on External Clouds

GIUSEPPE CODISPOTI

# Una Collaborazione Speciale

C. Aiftimiei[3,4], D. Bonacorsi[1,2], P. Calligola[1], V. Ciaschini[3],

***G. Codispoti[1,2]***, A. Costantini[3], S. Dal Pra[3], D. DeGirolamo[3],

R. Di Maria[1,2], C. Grandi[1], D. Michelotto[3], M. Panella[3], G. Peco[1],

L. Rinaldi[1,2], V. Sapunenko[3], M. Sgaravatto[5], S. Taneja[3], G. Zizzi[3]

[1]INFN Bologna, Bologna, Italy
[2]Physics and Astronomy, University of Bologna, Bologna, Italy
[3]CNAF, Bologna, Italy
[4]IFIN-HH, Magurele, Romania
[5]INFN, Padova, Italy

# Motivazioni

Gli esperimenti LHC richiedono ingenti risorse di calcolo a regime

Ma soprattutto lavorano spesso in «burst» mode

- devono cioè far fronte ad improvvisi picchi
- ad esempio per la ricostruzione di periodi di presa dati LHC particolarmente efficienti

Le risorse di calcolo a disposizione spesso non sono sufficienti e i tempi per assorbire i backlog posso essere lunghi per le necessità degli esperimenti

# Le soluzioni

La *virtualizzazione* dei servizi permette una estensione dinamica delle risorse

- L'aggiunta di nuove risorse equivale ad una nuova istanza di VM a partire da una stessa immagine
- Macchine idle possono essere temporanemente convertite per coprire le richieste

Il *Cloud Bursting* può evitare la creazione di lunghi backlog e ottimizzare l'acquisizione di risorse:

- Riallocazione dinamica delle risorse all'interno di una farm
- Estensione dinamica della farm su risorse Cloud (OpenStack)
- Estensione dinamica della farm su risorse Cloud publiche/private

L' *Accesso diretto Coud* come ulteriore risorsa

- già esplorato da CMS tramite *glidein-WMS*

# Tipico workflow di una analisi CMS

**Analisi Grid: esecuzione su un vero Worker Node Grid**

- software impacchettato da tool tipo CRAB
- accesso a dataset remoti ~1TB
- job dedicati alla selezione di eventi e fasi preliminari dell'analisi (eventualmente time consuming)
- creazione di user defined n-tuples
- salvataggio ntuple su storage Tier2/3 (~10GB)

**Analisi Locale: esecuzione su Nodo Farm Locale**

- Accesso ai dati selezionati (~10GB) e salvati su storage locale
  - Possibile accesso posix
- Software per analisi finale compilato in locale, a volte al di fuori del framework di esperimento
- Sottomissione via batch system locale (e.g. LSF)
  - Molti fisici sono abituati ad «girare nella propria directory»
  - Ciò implica home e aree storage accessibili dai nodi della farm
  - **Use case delicato**, ma ancora esistente, nel caso dell'estensione dinamica viene considerato poichè funzionale ai test

Nel Tier 3 Bologna i Worker Node coprono entrambi gli use case

Le immagini per WN virtuali vengono quindi create ibride, anche se in seguito specializzate

# Dettagli sulla creazione delle immagini

OZ tdl + kickstart, specializzati in base ai profili richiesti

Mapping utenti LDAP o Grid Pool Account

- No utenti locali

Dove richiesto accesso gpfs via export nfs da HV (soluzione non scalabile ma funzionale per i nostri test)

- home utenti
- area *local* con accesso posix
- area *storm* lettura posix/scrittura srm

Accesso software di esperimento tramite CVMFS (no installazione locale)

AFS dove richiesto

LSF via mounting remoto e source da script di configurazione centrale

ARGUS centrale T3 Bologna

In generale: Immagini più leggere possibile e il più possibile basate su servizi remoti

# I servizi Tier3 virtualizzati

User Interface e i servizi per l'utente

- Home utenti (ldap, home su gpfs, accesso storage, afs, accesso software esperimento e Grid via cvmfs, lsf, utilities)
- 2 nuove UI già usate sporadicamente dagli utenti CMS

Phedex e i servizi di trasferimento dati: attualmente in produzione!

- Una UI «depotenziata», software installato sulla home di un utente speciale (t3phedex)

Worker Node *«ibridi»*

- Testati con l'inserimento in una coda di test LSF dedicata quali Nodi della Farm locale (LSF)
  - Simile alla UI come configurazioni, ma nessun pacchetto utente o librerie grafiche
- Utilizzabili come Worker Node per l'accesso Grid
  - Aggiunta di Grid Pool Account, glexec, accesso storage solo via srm/xrootd

***Test di funzionalità con un workflow tipico di una analisi sul Quark Top***:

- Sottomissioni dalle nuove UI
- Job running su una coda LSF creata appositamente per i test

# Estensione dinamica della farm

Idea presentata da Vincenzo Ciaschini alla riunione CCR 30-31 marzo:

https://agenda.infn.it/materialDisplay.py?contribId=18&sessionId=0&materialId=slides&confId=9326

Il principio: aggiunta dinamica di VM quali nodi LSF tramite una VPN

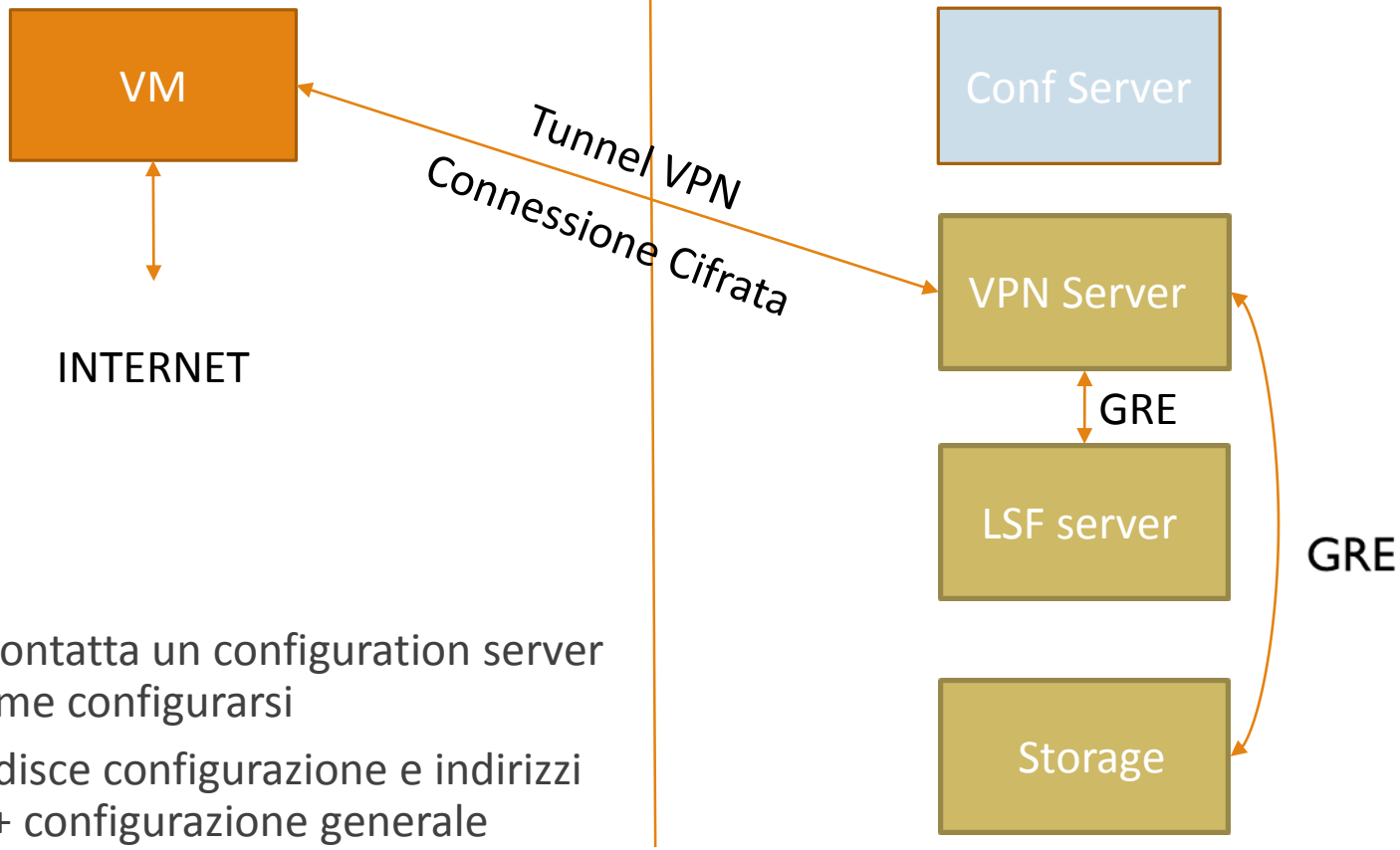
L'implementazione:

- tunnel VPN tra le VM remote e VPN server
- tunnel GRE tra il VPN server e le macchine della farm che devono essere necessariamente visibili alle VM (es: LSF server, …)

I vantaggi:

- Nessun requisito sull'hypervisor: la VM può essere lanciata su provider Cloud qualunque
- Semplici requisiti sulla VM (un RPM)
- Efficienza: il traffico è rediretto verso la farm solo per il minimo necessario.
- Dinamicità: se servono più risorse basta far partire nuove VM, se non servono più basta spegnerle

# In dettaglio (configurazione completa)




- Al boot, la VM contatta un configuration server per chiedere come configurarsi
- Conf server spedisce configurazione e indirizzi del VPN server + configurazione generale
- VM Contatta il VPN server, entra nella rete ed ha visibilità SOLO DEI NODI NECESSARI
- LSF Master prende carico del nuovo nodo
- VM completa la configurazione (route verso altri nodi quali storage, home, etc…)

# Test dell'estensione dinamica su T3

~100 Hello World per le funzionalità di base

- sottomessi su due WN inseriti nella coda di test usando un LSF server di test

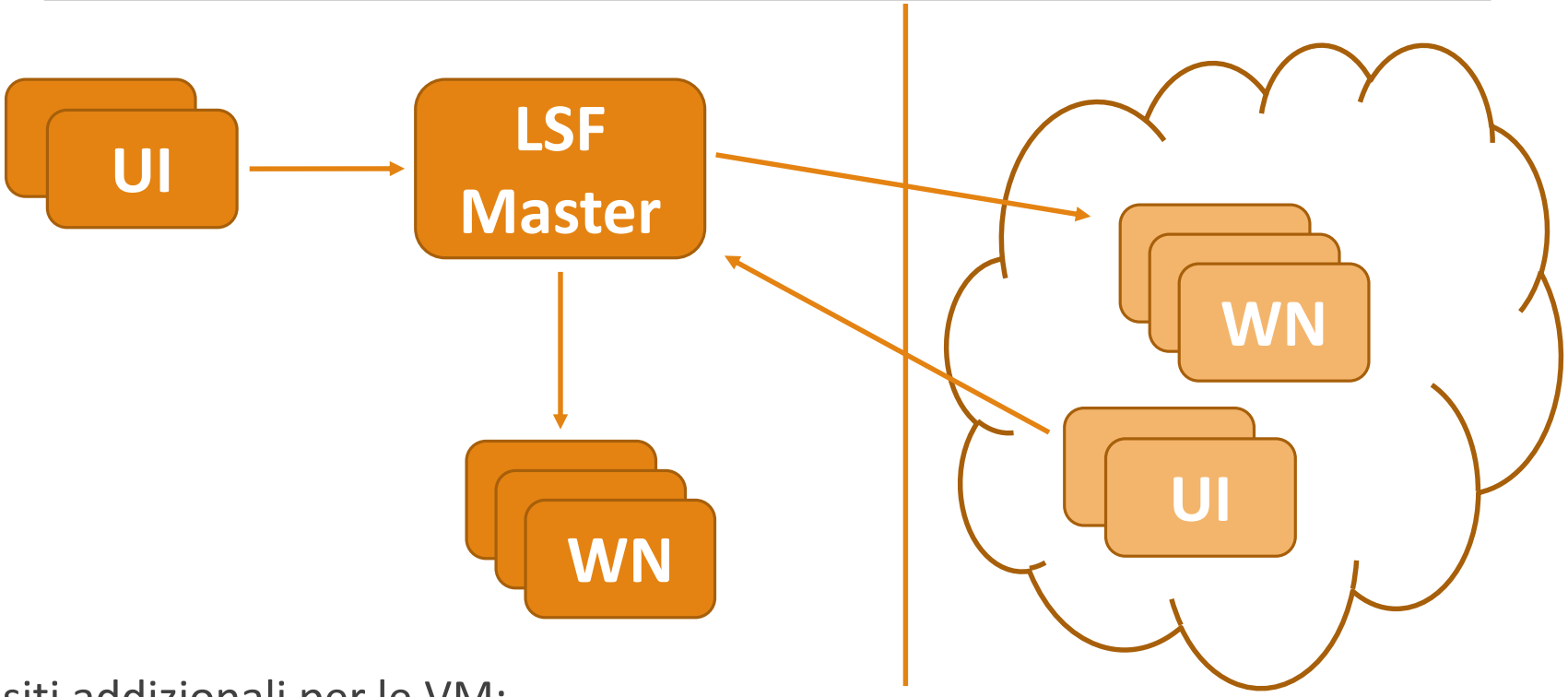
~100 Job di *analisi locale* (Top Quark) con accesso a Home e area dati gpfs

- sottomessi nelle stesse condizioni

Conclusioni:

- Il sistema risponde bene con semplici Hello World
- Unico problema riscontrato con i job di analisi: accesso allo storage locale (home utente e area dati)
  - Gpfs server è sensibile alle variazioni di nodi che vi accedono: impossibile un plug diretto
  - Il problema viene bypassato esportando le aree gpfs via nfs da hypervisor
  - La soluzione non è chiaramente scalabile
- L'estensione della farm locale con lo use-case accesso diretto ad aree utente è in linea di principio fattibile, ma richiede lo studio di una soluzione per storage locali quali gpfs

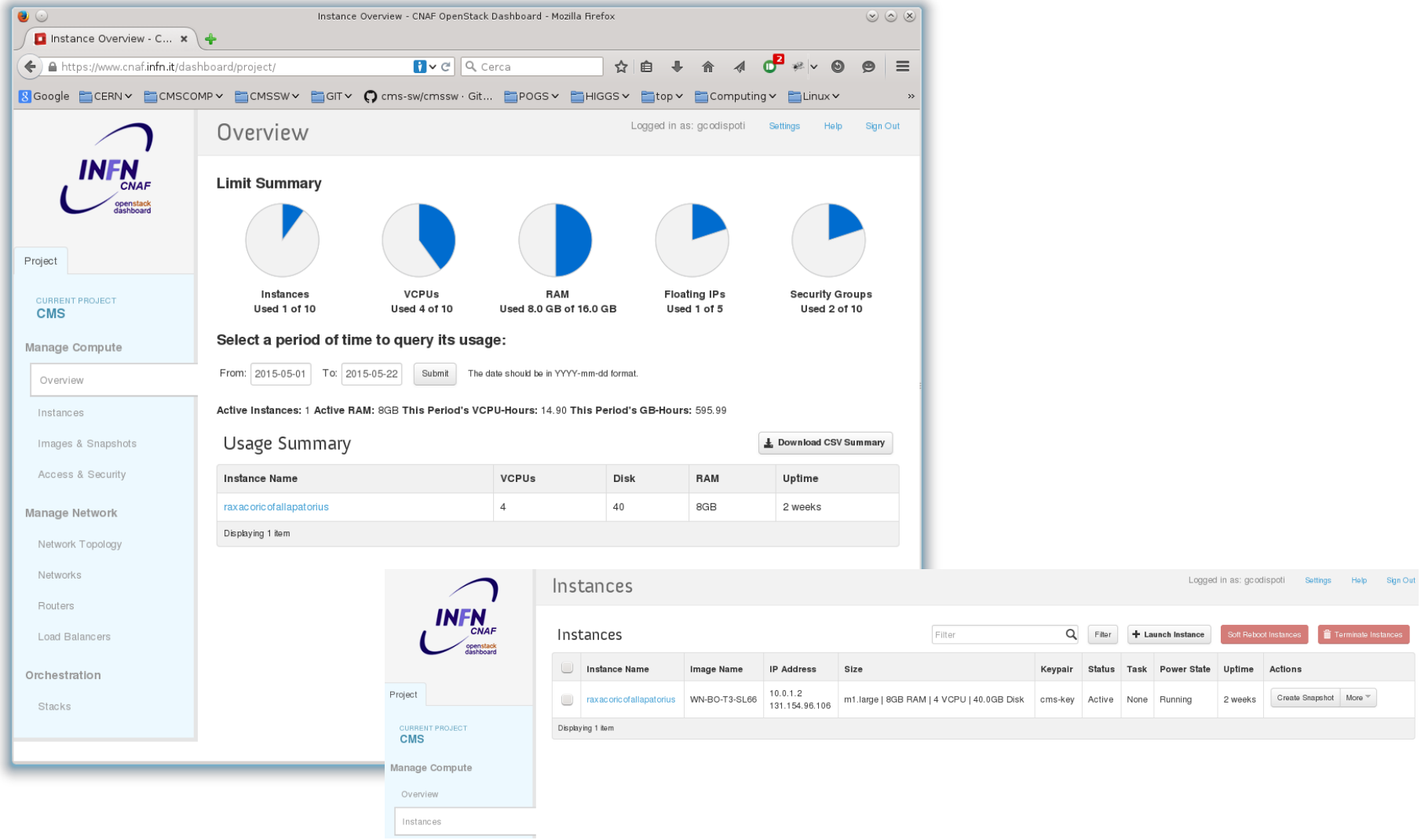
# Estensione dinamica su OpenStack




Requisiti addizionali per le VM:

- Aggiunta dei pacchetti Cloud
- Resize dinamico della partizione disco per salvare spazio
- Ridefinizione della configuazione di rete (dhcp, al momento le immagini devono essere create fuori da OpenStack)
- Eventuale ulteriore contestualizzazione

# Cloud @ CNAF: OpenStack Havana



# Considerazioni su OpenStack e Cloud

Abbiamo testato in primis lo use case «estensione della farm locale»

Performance:

- Estenzione dinamica perfettamente funzionante
- Unico problema di infrastruttura: performance gpfs (area contenente le immagini delle VM), per buona parte risolto
- Il bottleneck di accesso utente a gpfs via nfs è chiaramente la maggiore limitazione

Conclusioni:

- L'estensione dinamica su OpenStack di una farm locale, qualora desiderata, richiede lo studio di una soluzione per l'accesso diretto allo storage
- Nel seguito questo use case viene messo da parte a favore della mera funzione Worker Node ovvero: estensione dinamica di una farm Grid (Cloud Bursting)

# Cloud Bursting del Tier 3 Bologna

Ovvero estensione del sito Grid: WN istanziabili su richiesta su Cloud

- Non richiede accesso diretto allo storage ma sfrutta srm/xrootd
- Non usa utenti locali ma solo Grid Pool Account
- È lo use case naturale per l'utilizzo di Cloud commerciali

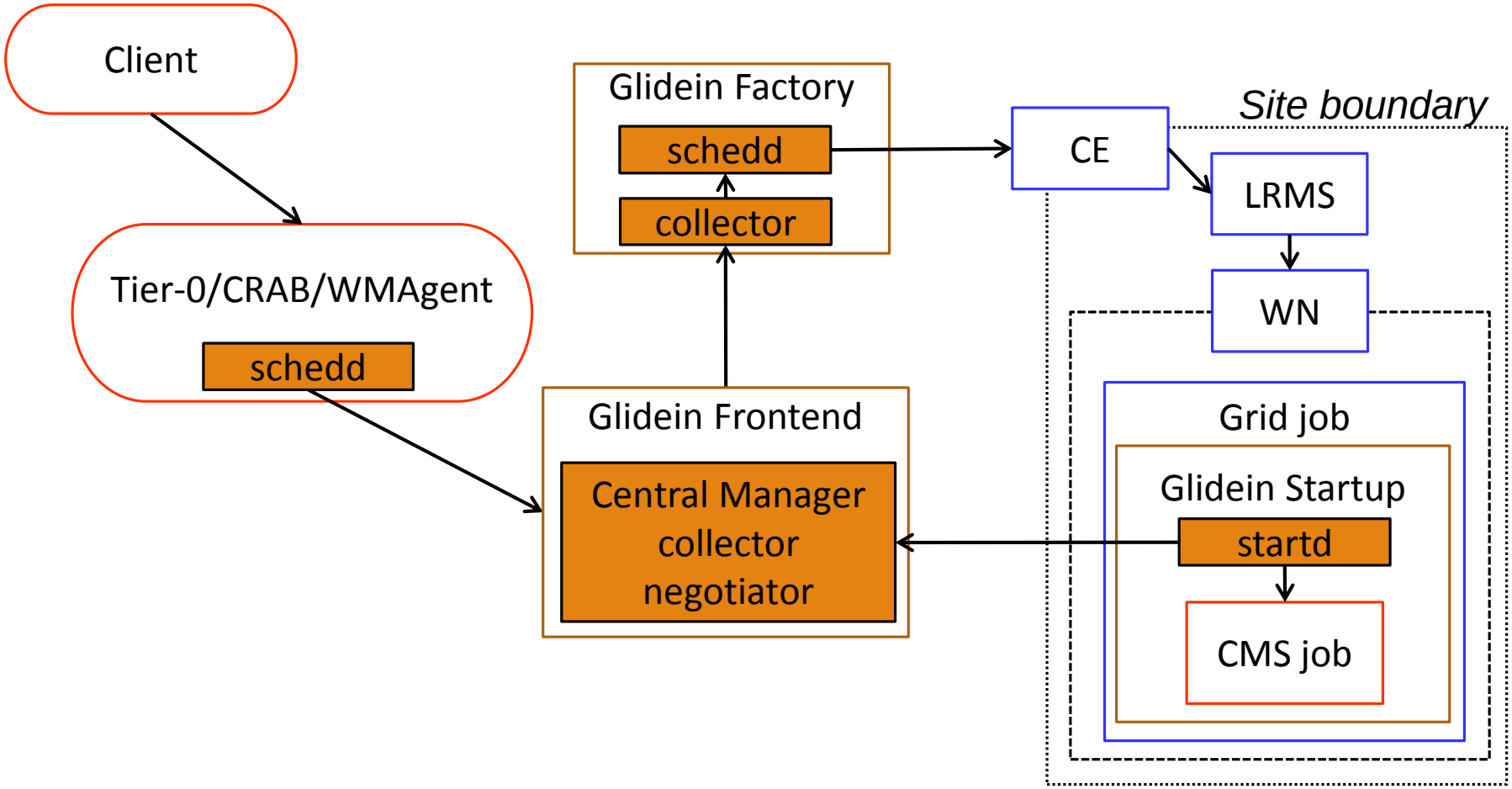
Stato dell'arte: setup completato

- I primi test sfruttano le stesse immagini e la stessa coda di test LSF
- semplici Grid jobs sottomessi direttamente al CE
- I job arrivano al WN ma abbiamo ancora qualche piccolo problema di comunicazione
  - jobWrapper non raggiunge il WN e i job falliscono

Risolto l'ultimo problema (pochi giorni si spera), contiamo di:

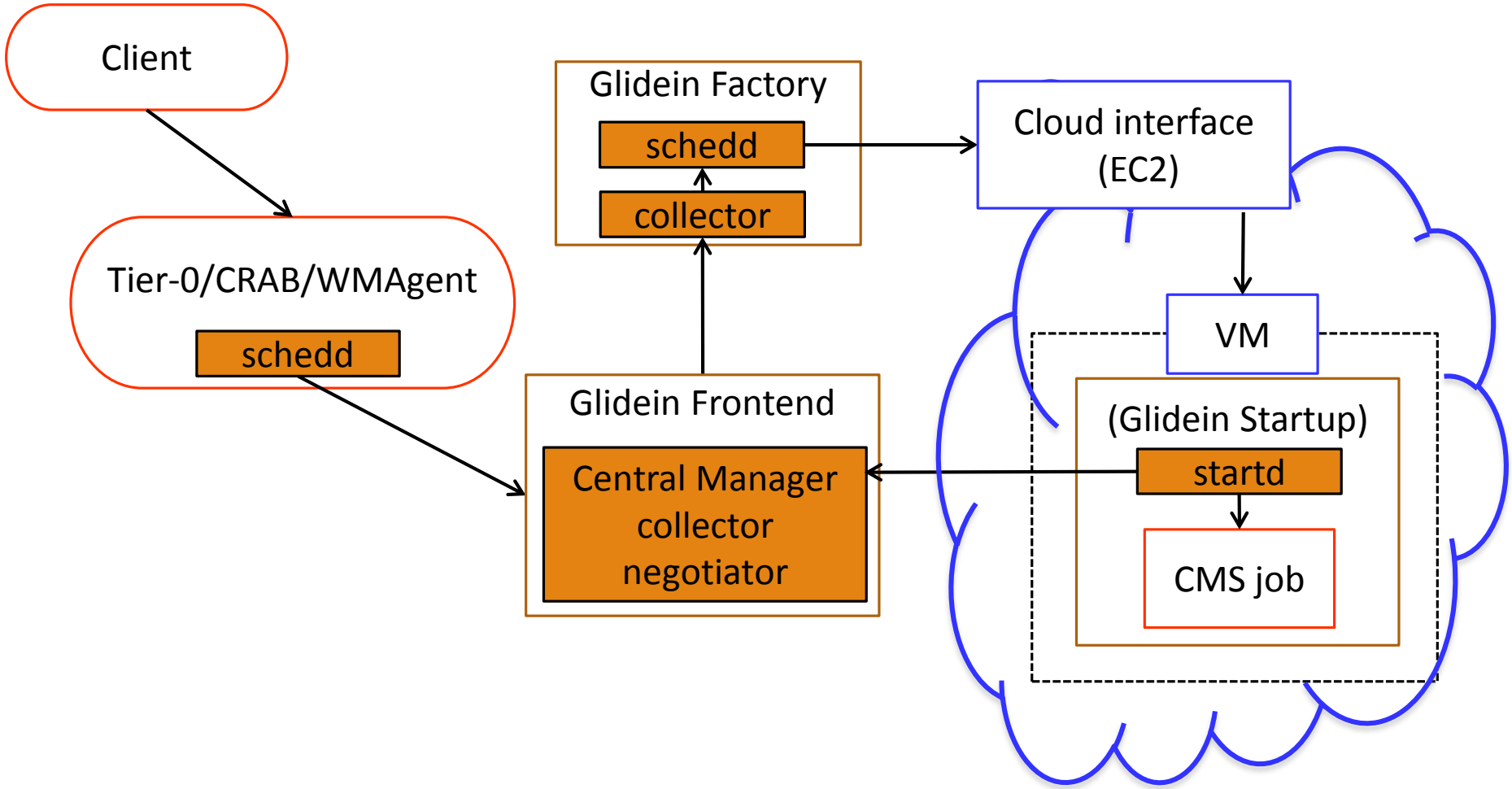
- Inserire i WN istanziati su OpenStack nella coda Grid del Tier3
- Sottomettere il *«solito»* workflow di analisi per un test di performance

# Accesso Grid tramite glidein-WMS




Claudio Grandi, 27/03/14

# Accesso Cloud diretto tramite glidein-WMS




Claudio Grandi, 27/03/14

# Sottomissione diretta a risorse OpenStack tramite glidein-WMS

CMS è interessato alla sottomissione diretta a risorse Cloud tramite glidein-WMS

Use case già realizzato in ambiente estremamente «protetto» e ben definito

- Presentato da Massimo Sgaravatto nel 2013
  - https://agenda.infn.it/getFile.py/access?contribId=37&sessionId=7&resId=0&materialId=slides&confId=6179
- Agile infrastructure (basata su OpenStack) per creare il Tier0
- Riutilizzo della farm HLT CMS durante le pause della presa dati
- Test in UK (coordinamento di Imperial College)
- Approvato grant di Amazon per l'estensione del Tier-1 di FNAL (e non solo) su AWS
  - Significativa collaborazione INFN

Nostro obiettivo: generalizzare lo use case sfruttando un glidein di test e l'infrastruttura OpenStack CNAF

# Glidein-WMS: stato dell'arte:

Glidein di test fornito da Massimo Sgaravatto
- già testato negli anni passati con **CRAB2**

Prossimo goal è il test con **CRAB3**
- Da verificare con Massimo la configurazione glidein-WMS

Immagine WN praticamente ultimata
- rimane da testare la configurazione semplificata glexec
  - YAIM porta con se troppe dipendenze e la versione su cvmfs non è usabile

Ottenuta l'apertura dell'infrastruttura OpenStack all'esterno per casi specifici quali lo use case in studio

Primi test previsti per inizio giugno

# Conclusioni

Abbiamo virtualizzato le principali risorse Grid necessarie ad un esperimento LHC

- Buona parte dei test fatti per CMS ma con un occhio alle esigenze comuni con ATLAS

Abbiamo testato la possibilità di virtualizzare la Farm locale per ottimizzare la gestione consentendone l'estensione, qualora necessaria

Abbiamo verificato la fattibilità di una estensione dinamica sulle risorse esistenti

Abbiamo verificato la fattibilità di una estensione dinamica su OpenStack fornita dal CNAF

Siamo quasi pronti per il test di Cloud Bursting del Tier3 di Bologna

Abbiamo preparato il terreno per l'accesso diretto a risorse Cloud utilizzando il glidein-WMS di CMS

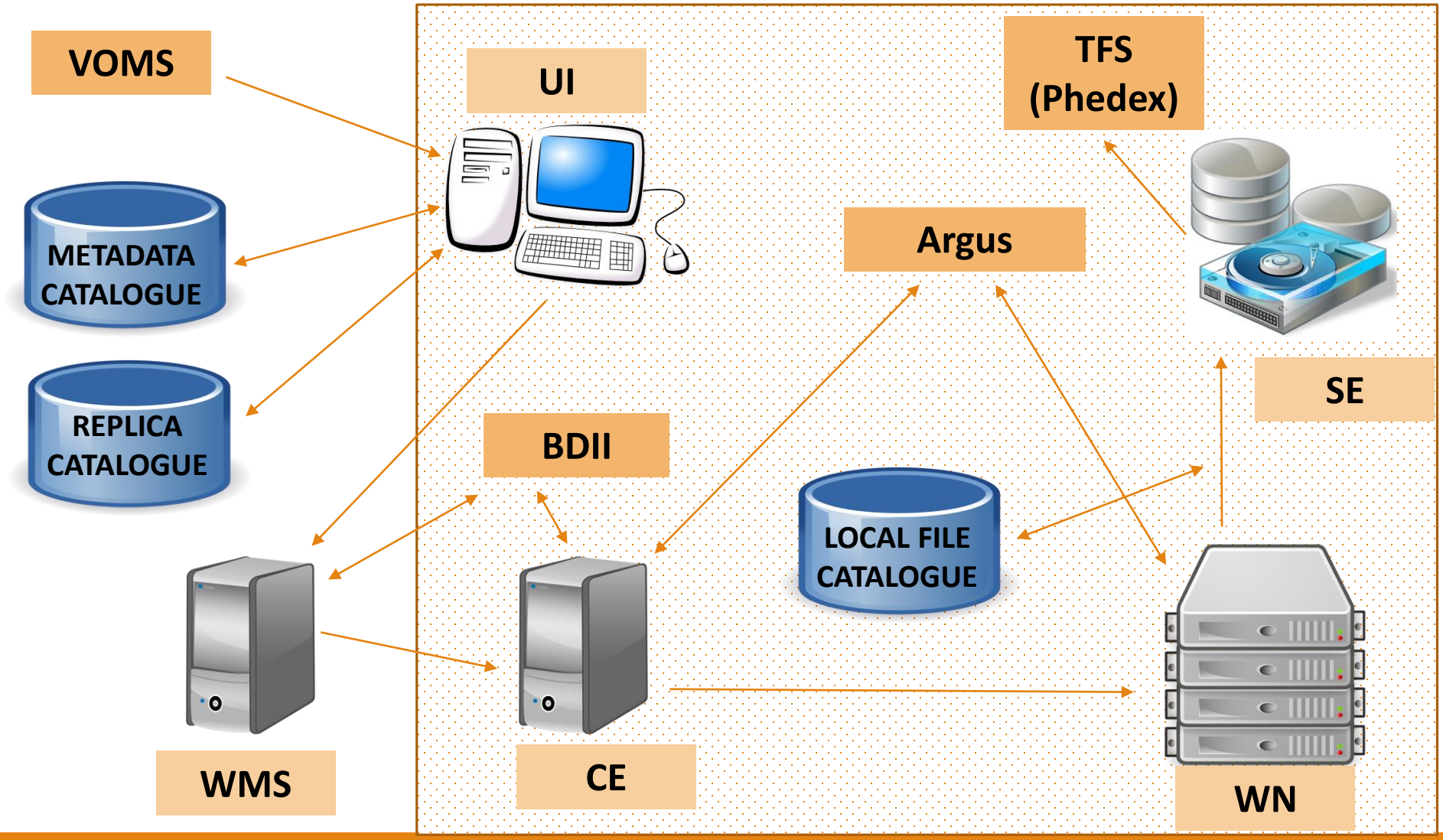
Tutti i test sono stati effettuati anche con *use case reali di CMS*!

Stiamo lavorando per poter ripetere anche *con use case reali ATLAS*

In futuro l'obiettivo è generalizzare l'approccio per consentire l'estensione dinamica su risorse Cloud di altro tipo e di altri provider, anche commerciali

# Backup

# Anatomia di un sito Grid



# Caratterizzazione di una farm CMS: l'esempio Tier 3 Bologna

3 **Hypervisors** hosting VMs (UI and test machines, ad esempio macchina dedicate **Phedex** per CMS: vmbo-t3-[01-03])

2 CE (cebo-t3-[01-02])

1 Argus (cebo-t3-03)

2 Top BDIIs (sgbo-t3-[01-02])

6 **UI** (VM) + 4 fisiche (2 CMS+2 ATLAS)

1 SE (sebo-t3-01) + 1 per test e reinstallazioni di nodi ( sebo-t3-02)

40 **WN**

Cluster gpfs (Home utente, area local con accesso posix, area storm lettura posix/scrittura srm)

Gridftp server (Gridftp-storm-t3.cr.cnaf.infn.it)

# Dettagli Cloud CNAF

IaaS infrastructure basata su OpenStack Juno

2 controller nodes set up in active/active high availability

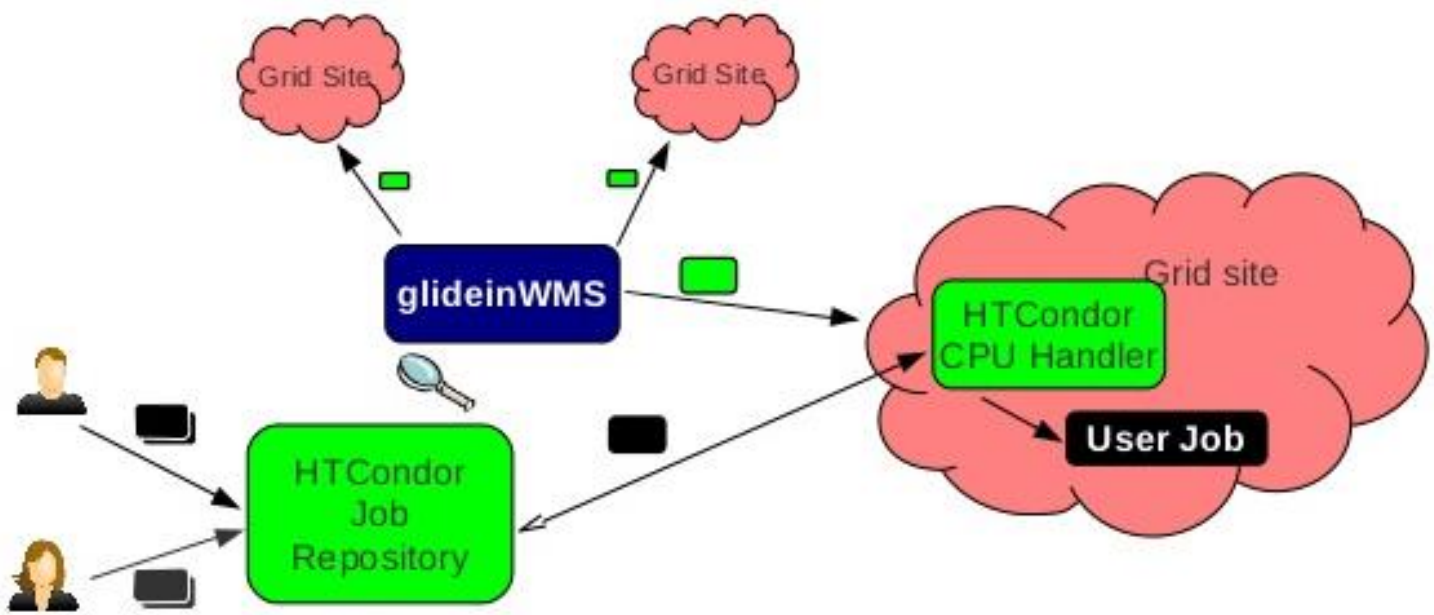
2 network nodes in active/active high availability making use of a HAproxy & Keepalived system

4 compute nodes

cluster of 3 nodes providing the database and message queueing services

IBM GPFS cluster providing shared storage for the entire infrastructure.

# CMS glidein-WMS



# CRAB 3 Workflow