GIORNATA DI INCONTRO BORSE DI STUDIO GARR "ORIO CARLINI"
MARTEDI' 12 DICEMBRE 2017 - ROMA

Consortium GARR

Progetto SCOReS

Studio di sistemi di Caching per l'utilizzo Ottimizzato di Risorse opportunistiche e siti senza pledged storage per applicazioni di e-Science

Davide Michelino

INFN
Istituto Nazionale di Fisica Nucleare

Introduzione generale del progetto

- Studiare, progettare, mettere in esercizio e testare **sistemi di Cache** per lo stoccaggio temporaneo e dinamico di data-set scientifici, **provenienti da sorgenti multiple**, distribuite geograficamente e disponibili con interfacce Cloud o Grid tradizionali.

- Il sistema pensato a supporto di siti e infrastrutture sulla rete GARR che offrono servizi per l'analisi dati di esperimenti o di altre applicazioni.

- Utilizzo di tecnologie standard, quali **HTTP** per un'introduzione **invisibile** e una larga applicazione di casi d'uso, anche in **differenti contesti**.

Motivazioni ed opportunità

Genesi
- Indagine argomenti CHEP/CERN
- Working Group EGI/WLCG/INFN

Necessità di ridurre nella gestione dei data-center
- **Manutenzione**
- **Forza lavoro**
- **Requisiti di affidabilità**

Intervenire su **storage** e **network**

Sistemi di Cache con l'utilizzo di risorse opportunistiche o diversi stakeholders (Cloud, etc) da inserire all'interno dei workflow applicativi e dei computing model degli esperimenti

Utilità di una cache

I sistemi di cache si prestano per molteplici casi d'uso e possono aprire nuovi scenari dando centralità al ruolo della rete.

Scenario 1: siti **storage-less**:

- Immagazzina e trasferisce solo i **dati necessari**;
- Se un file è **già presente** nella cache i trasferimenti risultano molto più veloci.

Scenario 2: sistema di cache come nuovo servizio sulla rete. Posizionato in un **POP GARR** o in un centro di servizi regionale o nazionale (Tier1/Tier2):

- **Avvicina** i dati lontani per i grossi utilizzatori;
- **Diminuisce** il traffico sulle lunghe tratte geografiche.

Scenario 3: accesso storage in **cloud**:

- Ammortizza l'accesso ripetuto a dati presenti su storage a pagamento (ex. Cloud S3).
- Facilitare l'**accesso** a dati conservati in storage accessibili su **link lenti** commerciali;

Il mio contributo

- Implementazione e sviluppo di un Sistema di cache generale integrabile nelle attuali infrastrutture di calcolo per le e-science grid o cloud
- Valutazione delle performance del prototipo sviluppato
- Verticalizzazione sui casi d'uso pilota Belle II e ATLAS
- Contribuire alla discussione internazionale interagendo con i principali working group
- Creare le condizioni per una facile replicazione del sistema su altri siti o in altri contesti

Metodologia di avanzamento

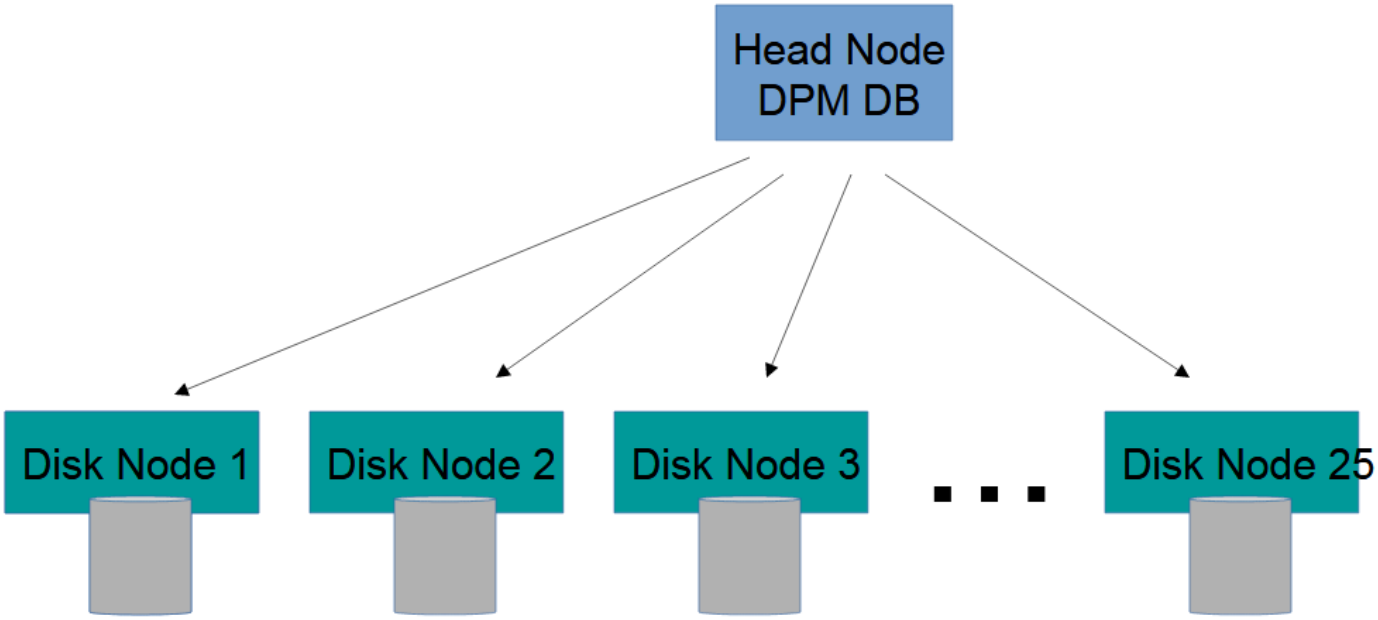
La prima parte dell'attività:

- Lo studio è partito da alcuni paper presenti in letteratura dove si illustravano varie soluzioni per l'introduzione di sistemi di cache HTTP nelle **infrastrutture di calcolo distribuito**, utilizzando tool comuni (squid, varnish, apache, etc)
- I limiti risultano essere la mancanza di supporto nativo HTTPS
- Non compliance con lo standard VOMS
- Difficoltà di integrazione con i tool standard dei sistemi di Grid/Cloud

Tecnologie individuate

- DPM – Sistema per la gestione di grossi pool di storage
- DYNAFED – tecnologia per federare endpoint di HTTP/webdav/s3

Le tecnologie: DPM (Disk Pool Manager)



Esempio di installazione (Napoli)
Totale: 1.8PB
Disk Node:

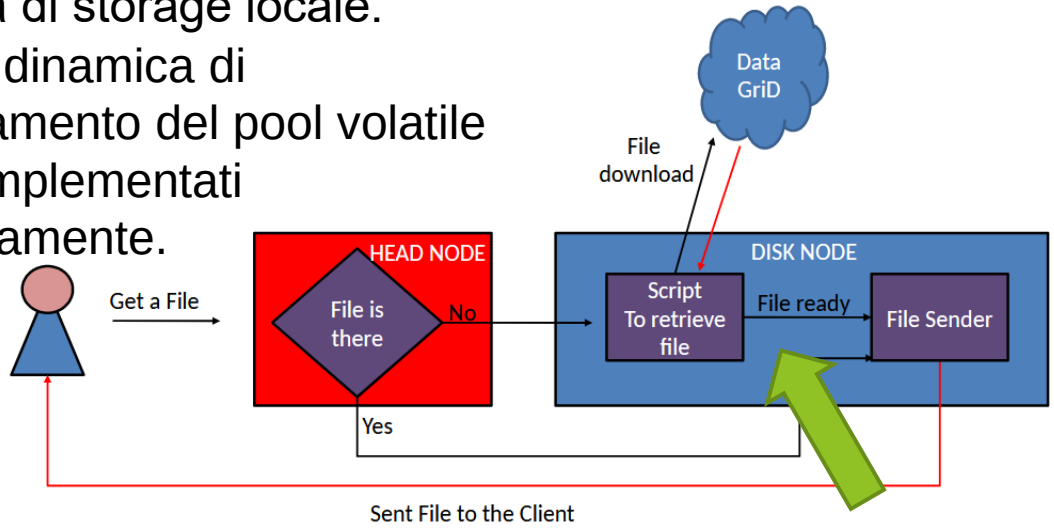
- connessioni 10/20Gbps
- Storage locale fino a 400TB

Le tecnologie: DPM (Disk Pool Manager)

- Tecnologia largamente diffusa
- **Scalabilità e performance** già dimostrate
- Già **testato** negli anni con installazioni in produzione con storage nell'**ordine dei PB**
- Supporto **HTTP** e XrootD
- **Know-how** locale già presente e contatti con gli **sviluppatori**
- Le nuove versioni offrono funzionalità utili all'implementazione di un sistema di cache con l'utilizzo di **pool volatili**

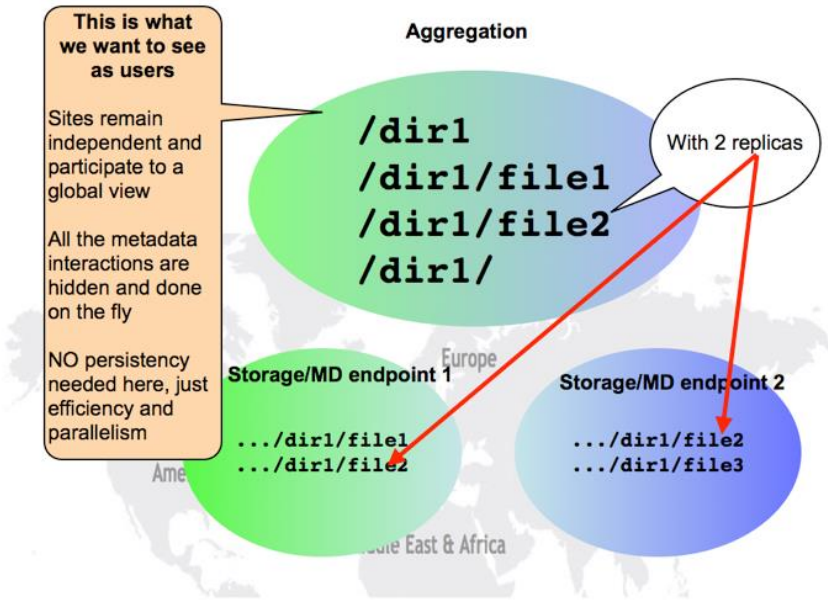
Le tecnologie: DPM – Pool Volatili

- Un pool in DPM è una collezione di File System visti come un unico namespace.
- **Un pool volatile** è un pool speciale che può recuperare file da fonti esterne, mediante l'ausilio di plugin scritti dall'utente.
- Quando un **client** effettua uno STAT o un GET di un file all'interno di un pool volatile, DPM esegue tali **script** per recuperare i metadati o il file stesso nel caso non sia presente nell'area di storage locale.
- Script e dinamica di funzionamento del pool volatile vanno implementati completamente.



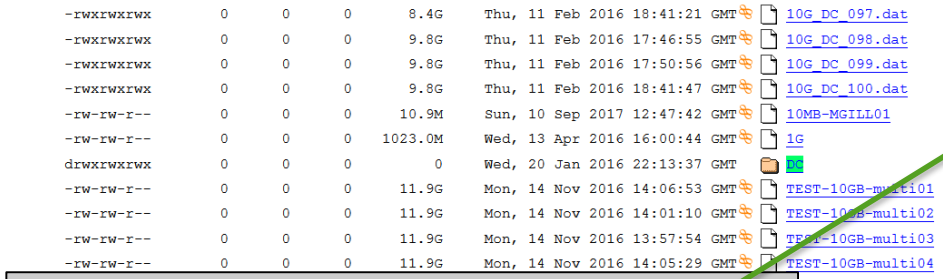
Le tecnologie: Il federatore DynaFed

Un altro elemento che ha catturato l'attenzione è il **federatore DynaFed**, prodotto opensource sviluppato dal CERN che aggrega endpoint di tipo **http/webdav**:



- Mostra un **unico albero** di directory
- Le repliche di uno stesso file sono rappresentate da un unico **metalink**
- Effettua il caching dei **metadati** per un browsing veloce dei file system remoti
- Capacità di individuare la **replica più vicina** al client sulla base dell'IP
- Gestisce i tipi di **autenticazione** comunemente utilizzati in ambito **grid** (certificati proxy + estensioni voms)

L'implementazione: DynaFed+Pool Volatile



-rwxrwxrwx	0	0	0	8.4G	Thu, 11 Feb 2016 18:41:21 GMT	10G_DC_097.dat
-rwxrwxrwx	0	0	0	9.8G	Thu, 11 Feb 2016 17:46:55 GMT	10G_DC_098.dat
-rwxrwxrwx	0	0	0	9.8G	Thu, 11 Feb 2016 17:50:56 GMT	10G_DC_099.dat
-rwxrwxrwx	0	0	0	9.8G	Thu, 11 Feb 2016 18:41:47 GMT	10G_DC_100.dat
-rw-rw-r--	0	0	0	10.9M	Sun, 10 Sep 2017 12:47:42 GMT	10MB-MGILL01
-rw-rw-r--	0	0	0	1023.0M	Wed, 13 Apr 2016 16:00:44 GMT	1G
drwxrwxrwx	0	0	0	0	Wed, 20 Jan 2016 22:13:37 GMT	DC
-rw-rw-r--	0	0	0	11.9G	Mon, 14 Nov 2016 14:06:53 GMT	TEST-10GB-multi01
-rw-rw-r--	0	0	0	11.9G	Mon, 14 Nov 2016 14:01:10 GMT	TEST-10GB-multi02
-rw-rw-r--	0	0	0	11.9G	Mon, 14 Nov 2016 13:57:54 GMT	TEST-10GB-multi03
-rw-rw-r--	0	0	0	11.9G	Mon, 14 Nov 2016 14:05:29 GMT	TEST-10GB-multi04

Il file XML specificato apparentemente non ha un foglio di stile associato. L'albero del documento è mostrato di seguito.

```
- <metalink version="3.0" generator="lcgdm-dav" pubdate="Mon, 14 Nov 2016 14:01:10 GMT">
  - <files>
    - <file name="/belle-">
        <size>12778995712</size>
      - <resources>
        - <url type="https">
            https://recas-dpm-01.na.infn.it/dpm/na.infn.it/home/belle/cache/TEST-10GB-multi02
          </url>
        - <url type="https">
            https://dpm1.egee.cesnet.cz:443/dpm/cesnet.cz/home/belle/TMP/belle/user/spardi/testhttp/TEST-10GB-multi02
          </url>
        </resources>
      </file>
    </files>
  </metalink>
```

L'idea è aggregare uno storage element con interfaccia webdav/http con un pool volatile di DPM.

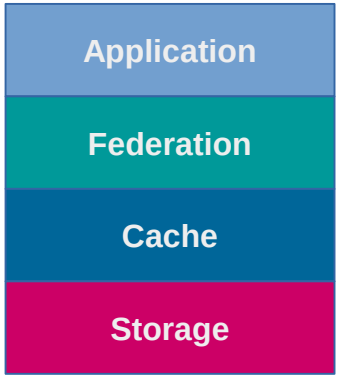
Con questa configurazione ogni file risulta avere almeno **due repliche**:

- Quella virtuale presente sul **pool volatile**
- Quella fisicamente presente nello **storage remoto**

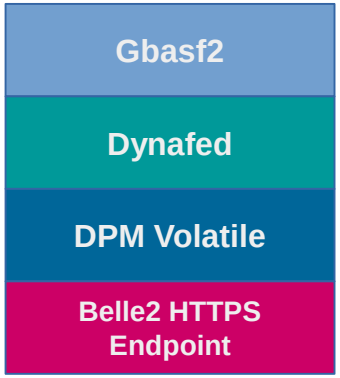
Quando un client accede al file se **è presente nel pool volatile**, viene restituito immediatamente, altrimenti viene invocato lo **script** per il **PULL** del file remoto. **Effetto cache!**

L'implementazione: Architettura

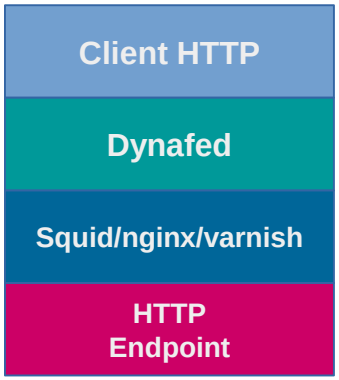
Stack dell'HTTP Caching

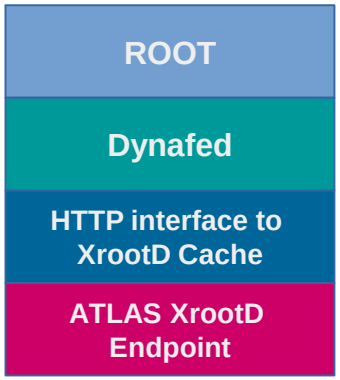


Implementazione attuale

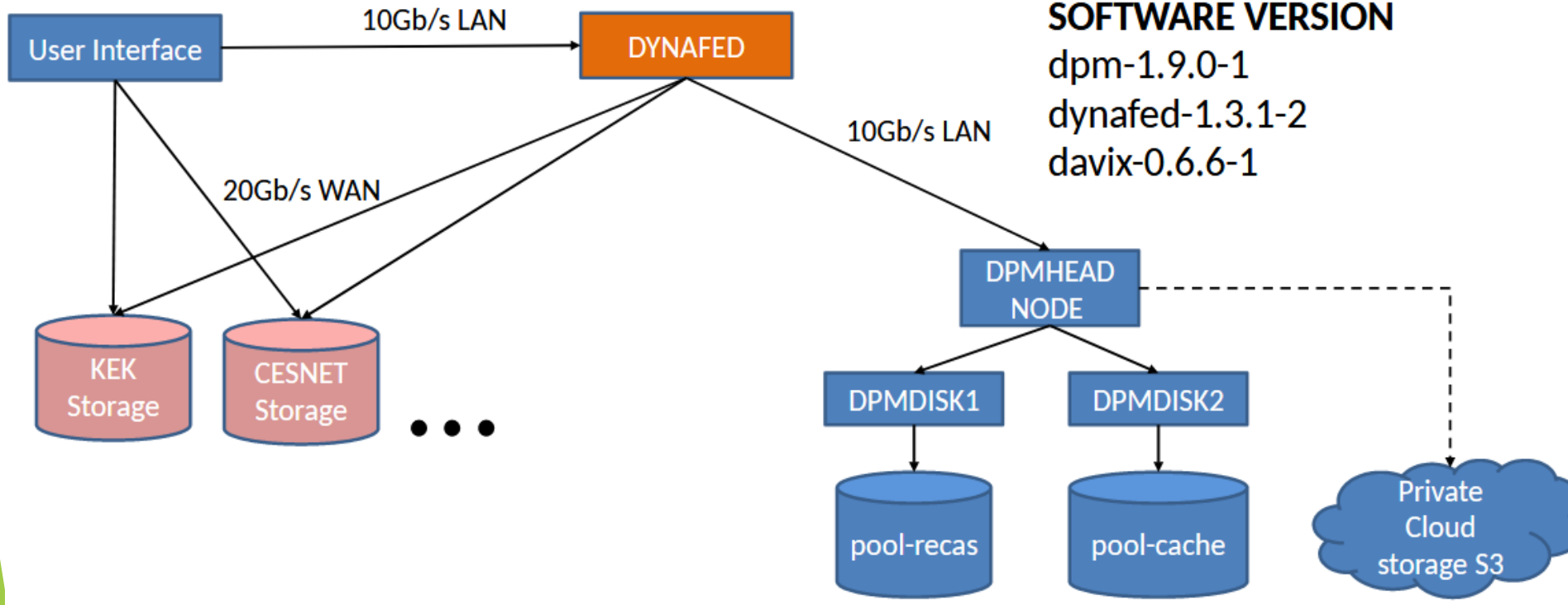


Possibli implementazioni alternative



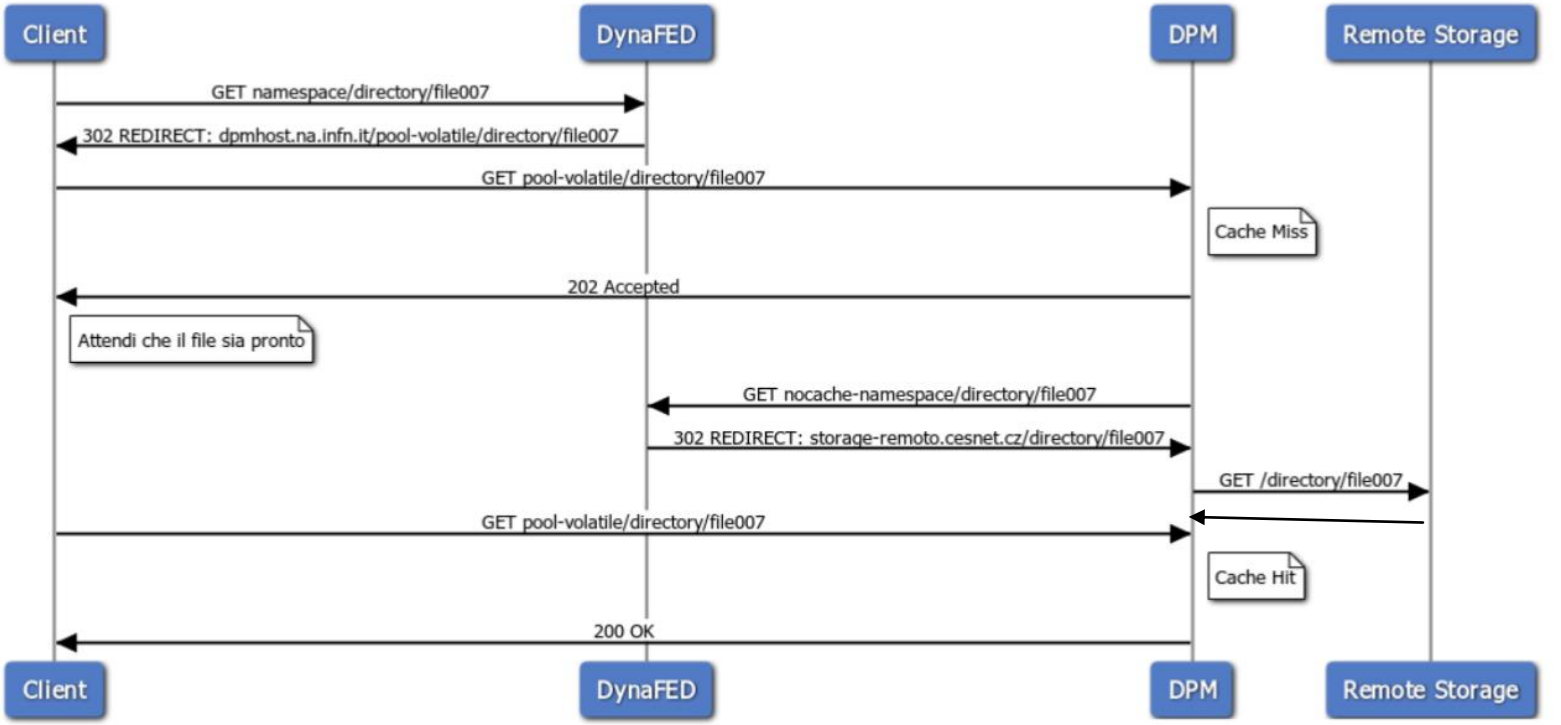


L'implementazione: DPM



SOFTWARE VERSION
dpm-1.9.0-1
dynafed-1.3.1-2
davix-0.6.6-1

L'implementazione: Cache Workflow

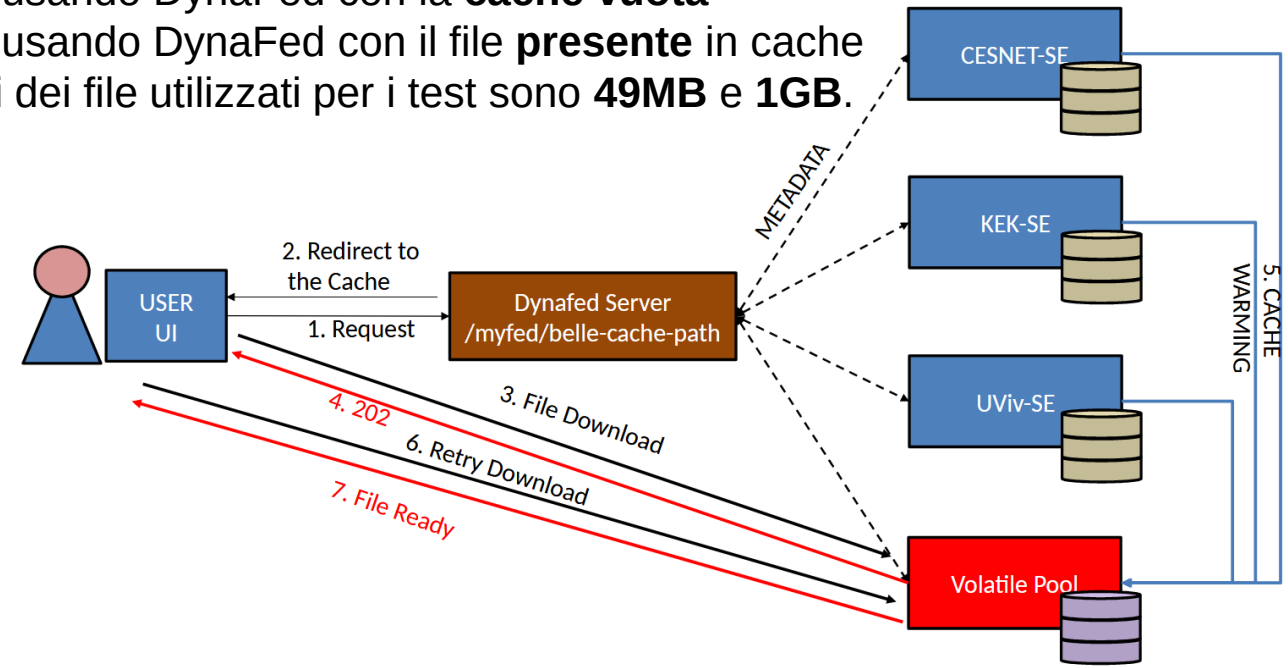


Il piano di test

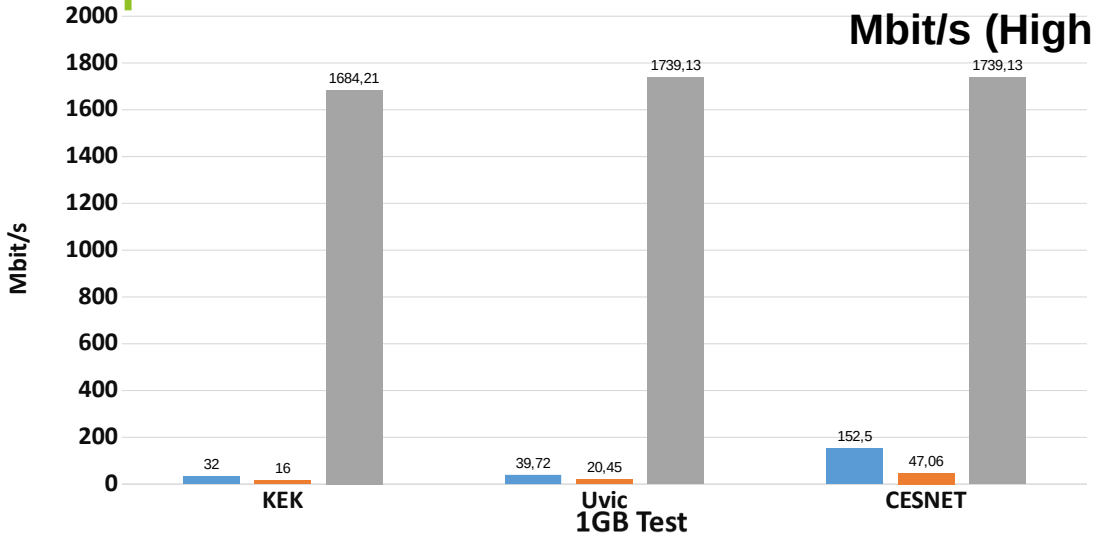
Il piano di test prevede il download (da una UI locale a Napoli) di **file di differenti dimensioni** dagli storage di Belle II distribuiti **geograficamente** (CESNET, Europa; KEK, Asia; UVic, America). Ogni file è scaricato tre volte come segue:

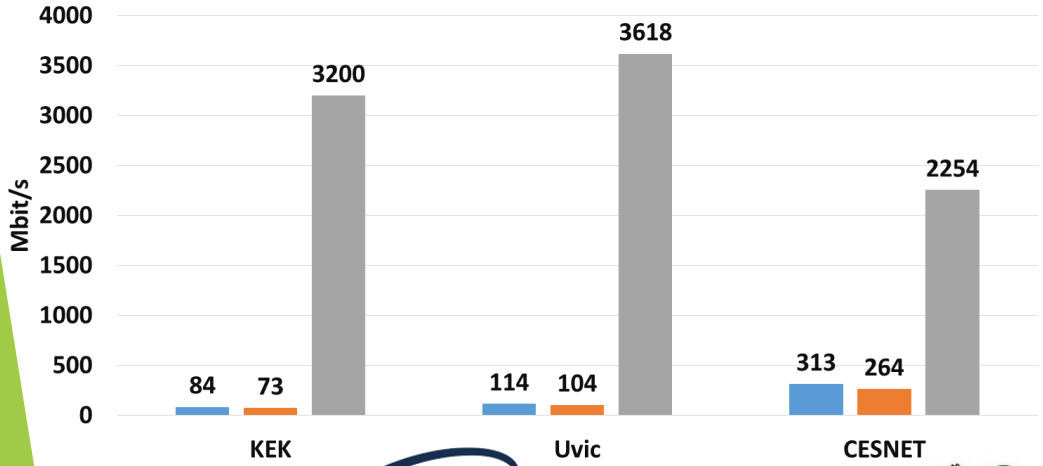
- Download diretto dallo **storage remoto**
- Download usando DynaFed con la **cache vuota**
- Download usando DynaFed con il file **presente** in cache

Le dimensioni dei file utilizzati per i test sono **49MB** e **1GB**.



Il piano di test: risultati





Le performance sono state calcolate al lordo degli overhead di connessione (accesso metadati, handshake, redirect, etc) e mostrano il raggiungimento dell'effetto desiderato.

Risultati raggiunti

I primi 10 mesi di attività hanno permesso di:

- E' stato definito un **modello**;
- Individuate le **tecnologie** necessarie;
- Implementare un **PoC** del sistema di cache;
- **Verificare** in maniera sperimentale l'efficacia utilizzando un'infrastruttura di produzione;
- Dare un contributo alle **discussioni** attualmente in corso nei **working group** internazionali che si stanno occupando dell'argomento:
  - Si è aperta la **collaborazione** con il **gruppo DPM** del CERN, entusiasta dell'attività in corso;
  - Lavoro presentato nel computing **workshop di Belle II**;
  - Confronto con il working group **ATLAS** Italia;
  - In sottomissione un contributo al **CHEP 2018**.

Attività proposta per l'anno di proroga

- Proseguire i lavori sul testbed di cache al fine di raggiungere uno stato di **pre-produzione**;
  - Funzionalità di **svuotamento** della cache;
  - Upgrade DPM, bugfix;
  - **Ottimizzazione** codice scritto;
  - Performance **stress test.**

- Sviluppare il **caso d'uso pilota** previsto dalla proposta di progetto dell'esperimento **Belle II**
  - Integrazione del testbed con **il framework basf2**;
  - Workflow 1 individuato: download e lettura locale;
  - Workflow 2 individuato: lettura remota.

Attività proposta per l'anno di proroga

- Studiare l'**integrazione** delle cache nel **computing model** degli esperimenti WLCG:
  - L'introduzione di un sistema di cache può avere un **impatto significativo** sul computing model degli esperimenti;
  - E' importante fare un'**analisi e un confronto** dei diversi modelli individuati con la cache posizionata nei POP/Tier1/Tier2/Siti contribuendo anche alle discussioni dei working group nazionali ed internazionali.

- **Massimizzare** l'impatto per **altre comunità** scientifiche e per la **rete GARR**:
  - Studiare un setup per **use-case generici**;
  - Creare dei tools per **replicare il setup** del sistema;
  - **Misurare** l'impatto della presenza di sistemi di cache sulla **rete** globale
  - Interazione con **risorse cloud** general purpose.
  - **Ottimizzazione** per l'accesso a storage in cloud

FINE

Backup Slides

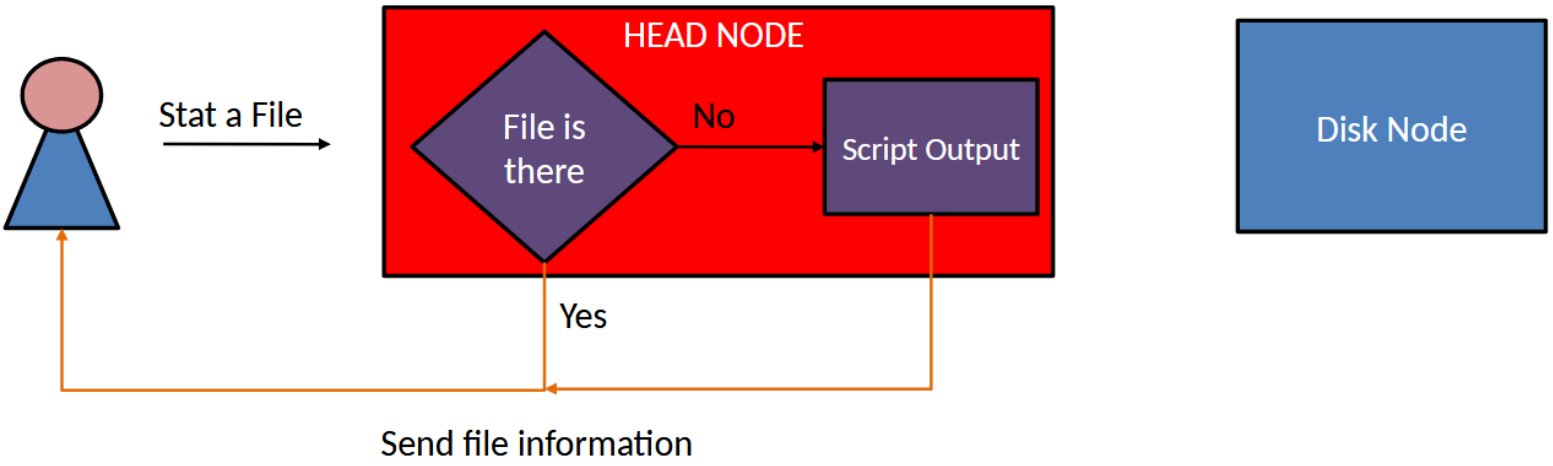
Backup Slide 1

Il federatore DynaFed

- **Dynafed**
  - Il federatore è l'elemento essenziale del sistema di cache, consente di introdurre una cache in maniera **trasparente**.
  - Gestisce in maniera ottimale il recupero dei file dai siti grid (geoip plugin) e si occupa della risoluzione dei path.
  - Il suo utilizzo è attualmentein studio da parte di molte comunità (WLCG, HEPiX).
  - Gestisce i tipi di autenticazione comunemente utilizzati in ambito grid (certificati proxy + estensioni voms)

Backup Slide 2

File Stat



Backup Slide 3

Dynafed Setup

Two views configured:

1. Aggregation of a set of Belle II storage endpoints [path /belle]

2. Aggregation of a set of Belle II storage endpoints + with the cache endpoint in Napoli. [path /belle-cache-path]

```
Example configuration for the view that include cache
….
locplugin.*.xlatepfx: /belle-cache-path/ /
….
glb.locplugin[]: /usr/lib64/ugr/libugrlocplugin_dav.so CESNET-SE 5
https://dpm1.egee.cesnet.cz:443/dpm/cesnet.cz/home/belle/TMP/belle/MC/merge1/

glb.locplugin[]: /usr/lib64/ugr/libugrlocplugin_dav.so SCOReS-CacheSE 5 https://recas-dpm-01.na.infn.it/dpm/na.infn.it/home/belle/cache/
```

Behaviour: in the example before, Dynafed creates a metalink with two endpoints, even in the file is not yet in the cache.

If the geoip plugin is activate the first endpoint for a client in Napoli will be always the local cache.

Backup Slide 4

Cache Implementation via DOME

Script on the Head Node:

The implemented script recognize if the requested path is a file or a directory then reply to the client.

Script on the Disk Node:

When a file is not in the cache, the disk node pulls the requested file by resolving the location via Dynafed using view that does not contain the cache.

Backup Slide 5

Client Behaviour

- If the cache is not ready the client receive a 202 Message that ask for waiting.
- Davix or gfal client will retry after a n-seconds (retry_delay) up to max_retry.
- Then the file will be download from the volatile pool

Backup Slide 6

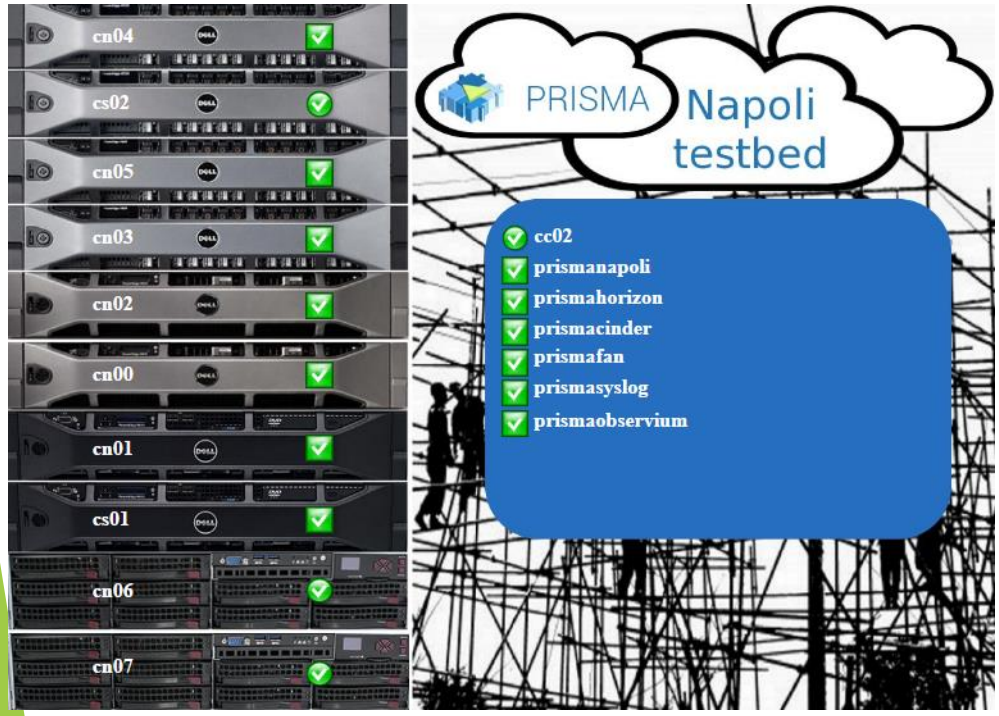
- DPM 1.9 with Dome will allow investigation of operating WLCG storage as a cache
- Scenarios
  - Data origin a regional federation of associated sites
  - Data origin the global federation
- A volatile pool can be defined which calls out to a stager on a miss
  - Caching logic implemented in a pluggable way
  - Hybrid cache/conventional setup
- Questions to investigate
  - Cache management logic
  - Different client strategies on miss
    - blocking read, async read, redirection to origin
  - Authentication solutions
  - Workflow adaptation for locality

CHEP 2016

We are trying to answer at these questions

Backup Slide 7

SCOReS Project: Facilities



Per il testbed è stata messa a disposizione la cloud OpenStack del PON Prisma, che per l'occasione è stata completamente ridisegnata.

- 2 Server (tot 80 Cores to store the collective service)
- 384 cores for computation
- 88TB Raw Data
- 10Gbps Network

Inoltre ho implementato uno storage DPM attualmente in fase di pre-produzione:

1 Head Node
2 Disk Node
150 TB Disk Space

Backup Slide 8

Autenticazione ed utilizzo di certificati

- Il client si autentica con un certificato proxy su DynaFed
- Dynafed funziona da redirector, consente al client di autenticarsi direttamente sul sistema di destinazione (cache/endpoint remoto)
- Dynafed internamente usa un proxy di un certificato robot per acquisire i metadati dei file e delle directory dagli storage che aggrega
- Il client si autentica con il proprio certificato utente (proxy) su DPM
- Nell'implementazione attuale DPM utilizza un proprio certificato proxy (che viene rinnovato periodicamente) per autenticarsi sugli storage remoti dove recuperare i dati da mettere nella cache
  - Può essere sostituto da un certificato server inserito nella VO
  - Verrà indagata la possibilità di delega del certificato proxy del client

Backup Slide 9

Test Results 49MB

CESNET -> Napoli

	Min (s)	Max(s)	Mean(s)	Mbit/s
Download from CESNET	2,3	3,1	**2,6**	**152,5**
Download from Dynafed (empty cache)	7,4	10,5	**8,5**	**47,1**
Download from Dynafed (warm cache)	0,2	0,3	**0,2**	**1.739,1**

UVic -> Napoli

	Min (s)	Max(s)	Mean(s)	Mbit/s
Download from UVic	8,9	11,6	**10,1**	**39,7**
Download from Dynafed (empty cache)	16,9	22,6	**19,6**	**20,5**
Download from Dynafed (warm cache)	0,2	0,4	**0,2**	**1.739,1**

KEK -> Napoli

	Min (s)	Max(s)	Mean(s)	Mbit/s
Download from KEK	9	11	**10**	**32,0**
Download from Dynafed (empty cache)	20	22	**20**	**16,0**
Download from Dynafed (warm cache)	0,2	0,2	**0,2**	**1.684,2**

Backup Slide 10

Test Results 1GB

CESNET -> Napoli

	Min (s)	Max(s)	Mean(s)	Mbit/s
Download from CESNET	17,71	39,99	**25,57**	**312,9**
Download from Dynafed (empty cache)	22,17	39,34	**30,35**	**263,6**
Download from Dynafed (warm cache)	1,93	6,76	**3,55**	**2.253,5**

UVic -> Napoli

	Min (s)	Max(s)	Mean(s)	Mbit/s
Download from UVic	65,31	84,74	**70,21**	**113,9**
Download from Dynafed (empty cache)	73,6	79,7	**77,17**	**103,7**
Download from Dynafed (warm cache)	2.0	3,18	**2,211**	**3.618,3**

KEK -> Napoli

	Min (s)	Max(s)	Mean(s)	Mbit/s
Download from KEK	89	104	**95**	**84,2**
Download from Dynafed (empty cache)	107	113	**109**	**73,4**
Download from Dynafed (warm cache)	1,97	5,65	**2,5**	**3.200,0**

Backup Slide 11

Dettagli dello stato attuale

Il sistema funziona su una serie di use-case ad hoc, mancano allo sviluppo una serie di dettagli in parte derivanti da alcuni limiti del pool volatile di DPM, che saranno superati nelle prossime release, in parte dall'attività del progetto ancora in corso.

- Il pool volatile è limitato ad un livello di gerarchia di directory (bug open at the DPM team)
- La libreria davix (utilizzata dai client) limita il numero di redirect a 5 (hardcoded). Limite momentaneamente superato da una ricompilazione della libreria con una patch sviluppata nell'ambito del progetto. Bugfix da sottomettere agli sviluppatori.
- Malfunzionamento della logica di cancellazione dei file dalla cache quando piena.

Scalabilità

Backup Slide 12

Scalabilità Dynafed

- Dynafed è un agente state-less all'interno del sistema di cache, può scalare orizzontalmente semplicemente replicando il server (anche mantenendo un database comune per il caching dei metadati tra le varie instanze)
- Test già effettuati dai developer dimostrano che raggiunge performance di picco tra i 500k e 1M hits/seconds per core. Rif: https://indico.jinr.ru/getFile.py/access?contribId=110&sessionId=7&resId=0&materialId=slides&confId=60

Scalabilità DPM

- Per DPM è dimostrata la scalabilità grazie alle installazioni in produzione nei tiers LHC (esempio Tokio 6PB, Napoli 1.8PB)
- rif: https://indico.cern.ch/event/559673/contributions/2268612/attachments/1375991/2090978/2016-11-23_TNakamura.pdf