GARR

The Italian Academic & Research Network

www.garr.it

INFN BARI

# Sperimentazione del file-system distribuito HDFS in ambiente GRID

Giovanni Marzulli

Tutor: Domenico Diacono

III Borsista Day, Roma, 06.12.2012

Consortium GARR

# Outline



- Use cases
- Hadoop Distributed File System
- Test di funzionalità
- Sviluppo di politiche di replica dei dati
- Monitoring di sistema
- Test di performance
- Sviluppi futuri e conclusioni

# Use cases



- Analisi dei dati scientifici su cluster di host con CPU e storage condivisi
- HA del servizio di accesso ai dati anche con hardware modesto ed economico
- HA del servizio di accesso ai dati per resistere alle failure di interi centri di calcolo

# Hadoop Distributed File System

# Cosa è Hadoop



- Hadoop è un software framework per il calcolo distribuito che privilegia l'affidabilità e la scalabità
- Il progetto Hadoop include:
  - Hadoop Distributed File System (HDFS)
  - YARN
  - MapReduce
  - Altri progetti correlati:
    - Avro, Cassandra, Hive, Chukwa, HBase, ecc..

# HDFS: Features

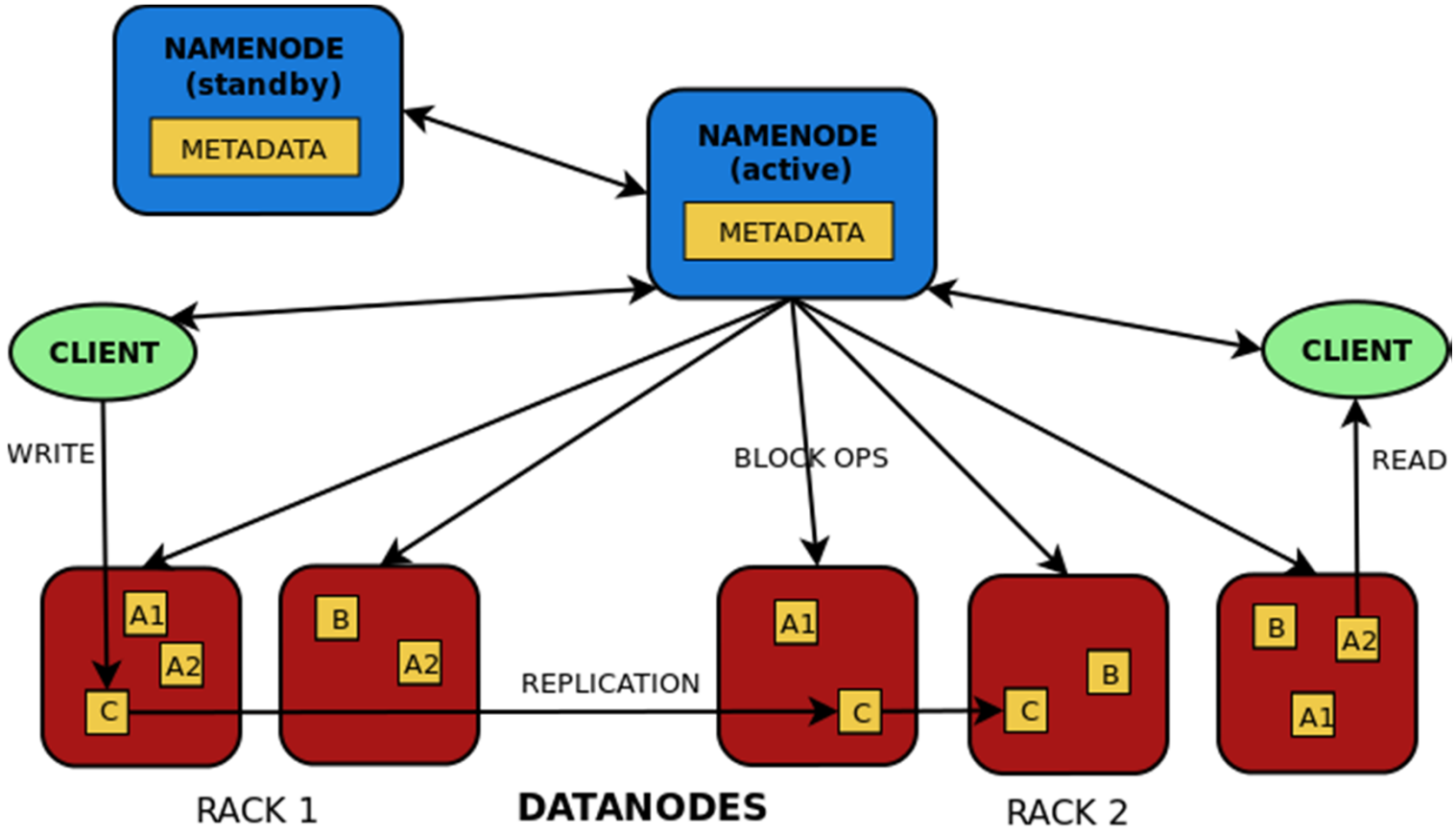


**H**adoop **D**istributed **F**ile **S**ystem

- Open source
- Large dataset
- Fault tolerance
- Scalabilile
- Commodity hardware
- Rack awareness

# HDFS: Architettura





# Fault tolerance



- “The primary objective of HDFS is to store data reliably even in the presence of failures.”
- File dati divisi in blocchi
- Ridondanza dei blocchi dati
  - Politiche di replica
- High Availability namenode

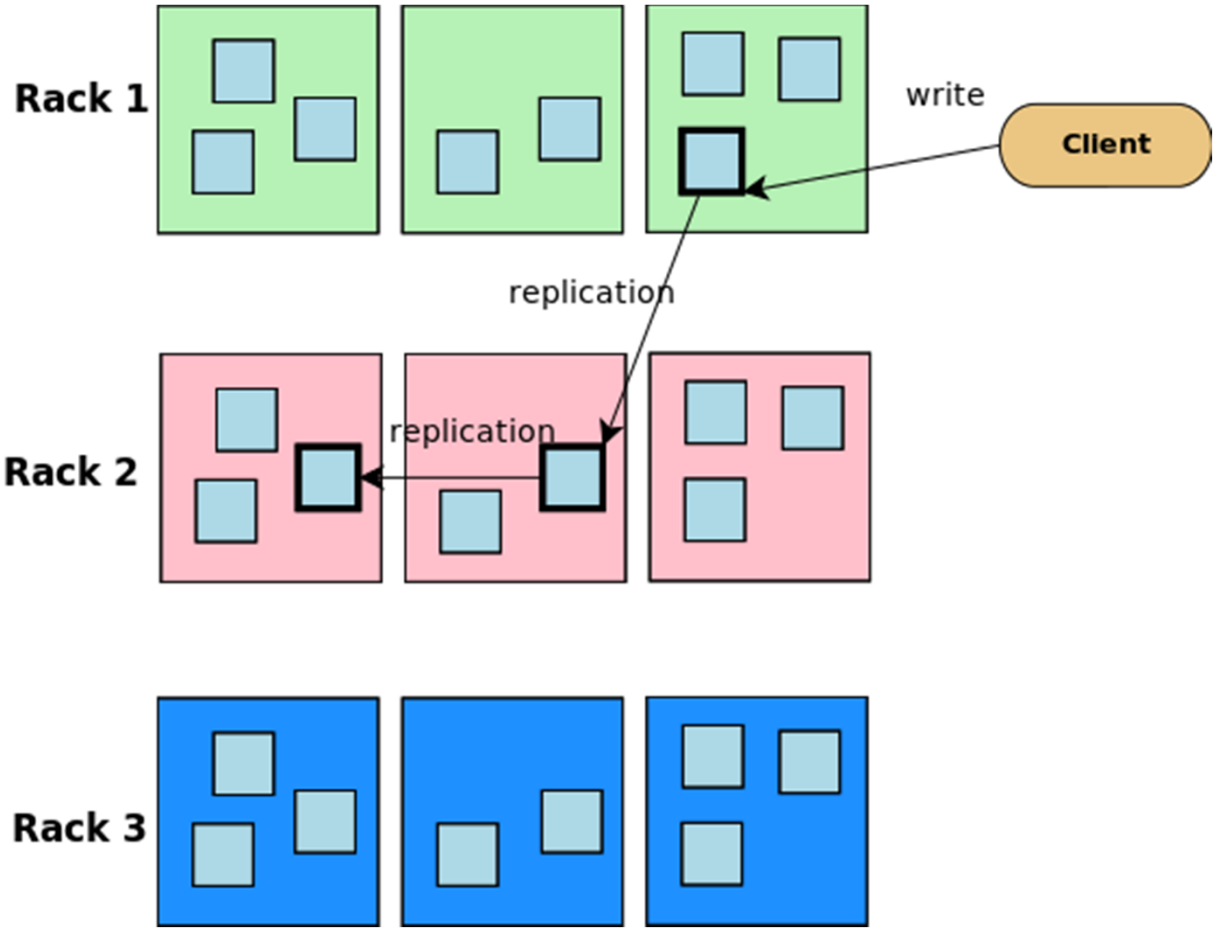
# Politiche di replica



- Predefinita in HDFS:
  - Una replica su un nodo del rack locale, due repliche su nodi differenti di un rack remoto
- Politiche sviluppate:
  - One Replica per Rack
  - Hierarchical

# Politica di replica predefinita

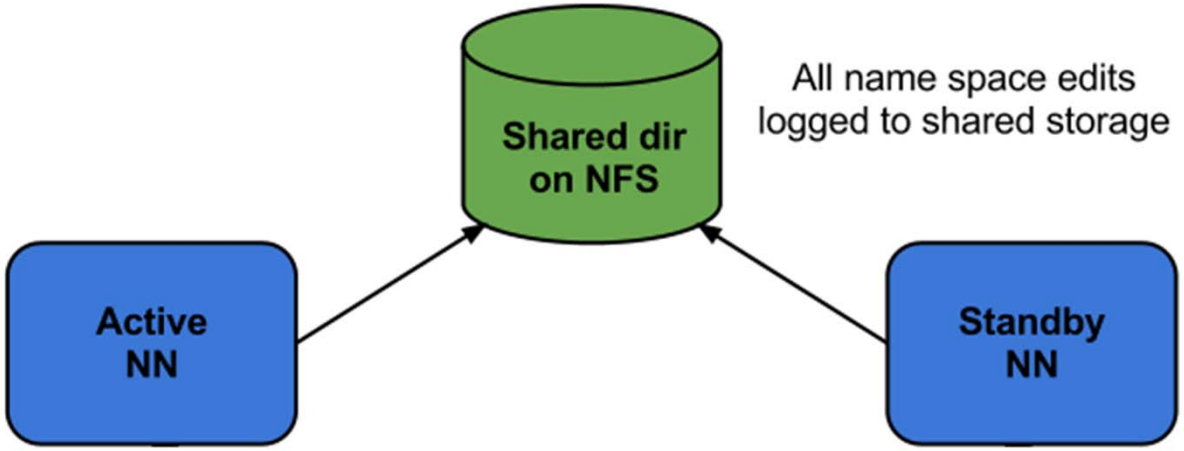




# High Availability namenode



- Sincronizzazione dei metadati attraverso una directory condivisa




- Failover:
  - Active → Standby
  - Standby → Active

# Test di funzionalità

# Test di installazione



- Cluster formato da 7 macchine del sito INFN-Bari
  - Differenti SO, hardware different, reti separate con firewall
- 7 datanode
- 1 primaray namenode
- 1 secondary namenode
- Da Hadoop 0.20 a 2.0 (CDH4.1)

# Test dei namenode

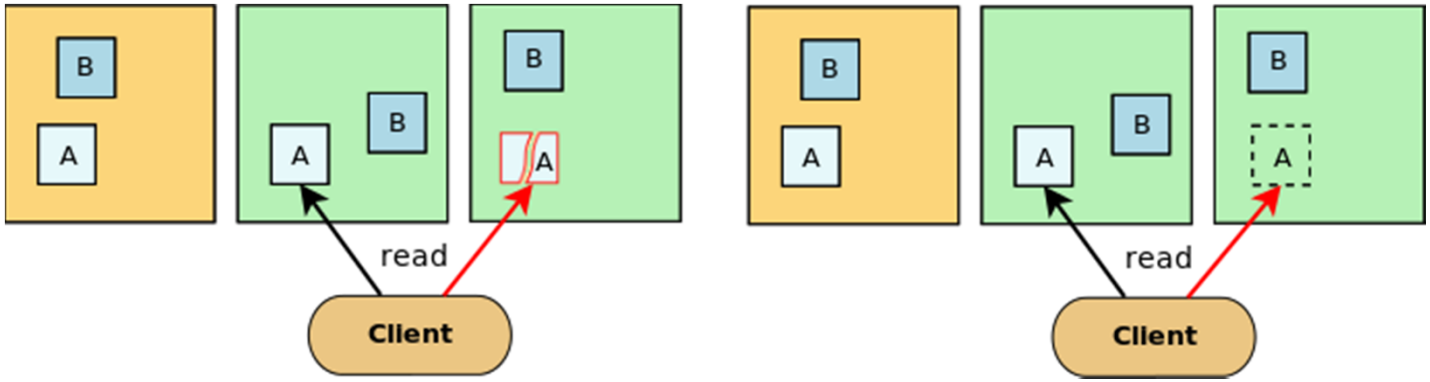


- Metadati danneggiati o persi
  - Ripristino dal secondary namenode:
    - `hadoop-daemon start namenode –importCheckpoint`
- Fallimento del namenode
  - I client e di datanode restano in attesa
  - Failover:
    - `hdfs haadmin –failover nn1 nn2`

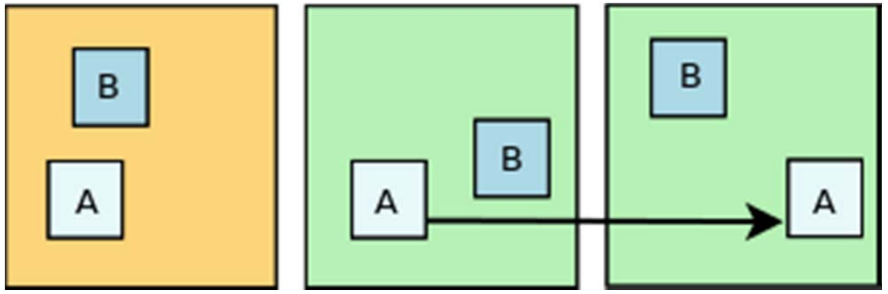
# Test dei datanode



- Blocchi dati danneggiati o persi




- Ripristino automatico



# Test dei datanode

- Blocchi under-replicated
  - Dopo il fallimento di un datanode
- Blocchi over-replicated
  - Dopo il ripristino e riavvio di un datanode
- Blocchi mis-replicated
  - Violazione della politica di replica
- Fallimento di un datanode durante processi di scrittura e lettura
  - Switch su altri nodi attivi
- Workload

# Client HDFS



- Client predefinito Hadoop
  - `bin/hadoop fs –put testfile.dat /marzulli/`
  - `bin/hadoop fs –get /marzulli/testfile.dat ./`
- Client Fuse-Dfs
  - Monta HDFS in userspace
  - `cp testfile.dat /mnt/hadoop/marzulli/`
  - `cp /mnt/hadoop/marzulli/testfile.dat ./`

# Autenticazione Kerberos



- Di default, ogni meccanismo di sicurezza è disabilitato
- Autenticazione dei nodi e degli utenti tramite il servizio Kerberos
  - Keytab
  - Ticket
- Livelli di autorizzazione gestiti dai permessi sui file

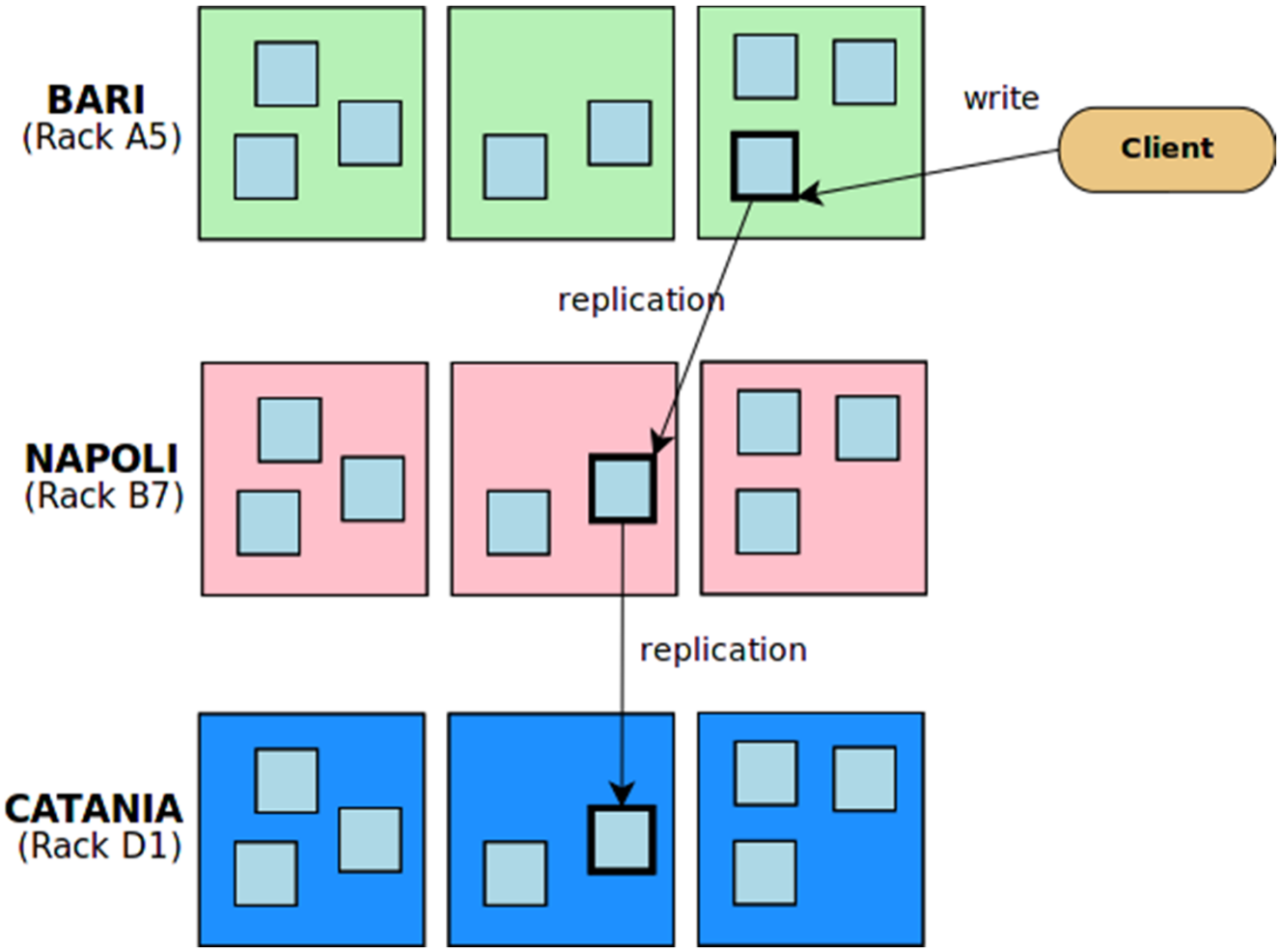
# Sviluppo delle politiche di replica

# One replica policy



- 1 replica per rack
- Maggiore affidabilità
  - Tolleranza al fault di 2 racks (se fattore di replica 3)
- Su scala geografica
  - Maggiore distribuzione dei dati
  - Riduzione del costo in lettura
    - Lettura dalla copia più vicina
  - Incremento del costo in scrittura
    - Maggiore quantità di dati trasmessi

# One replica policy



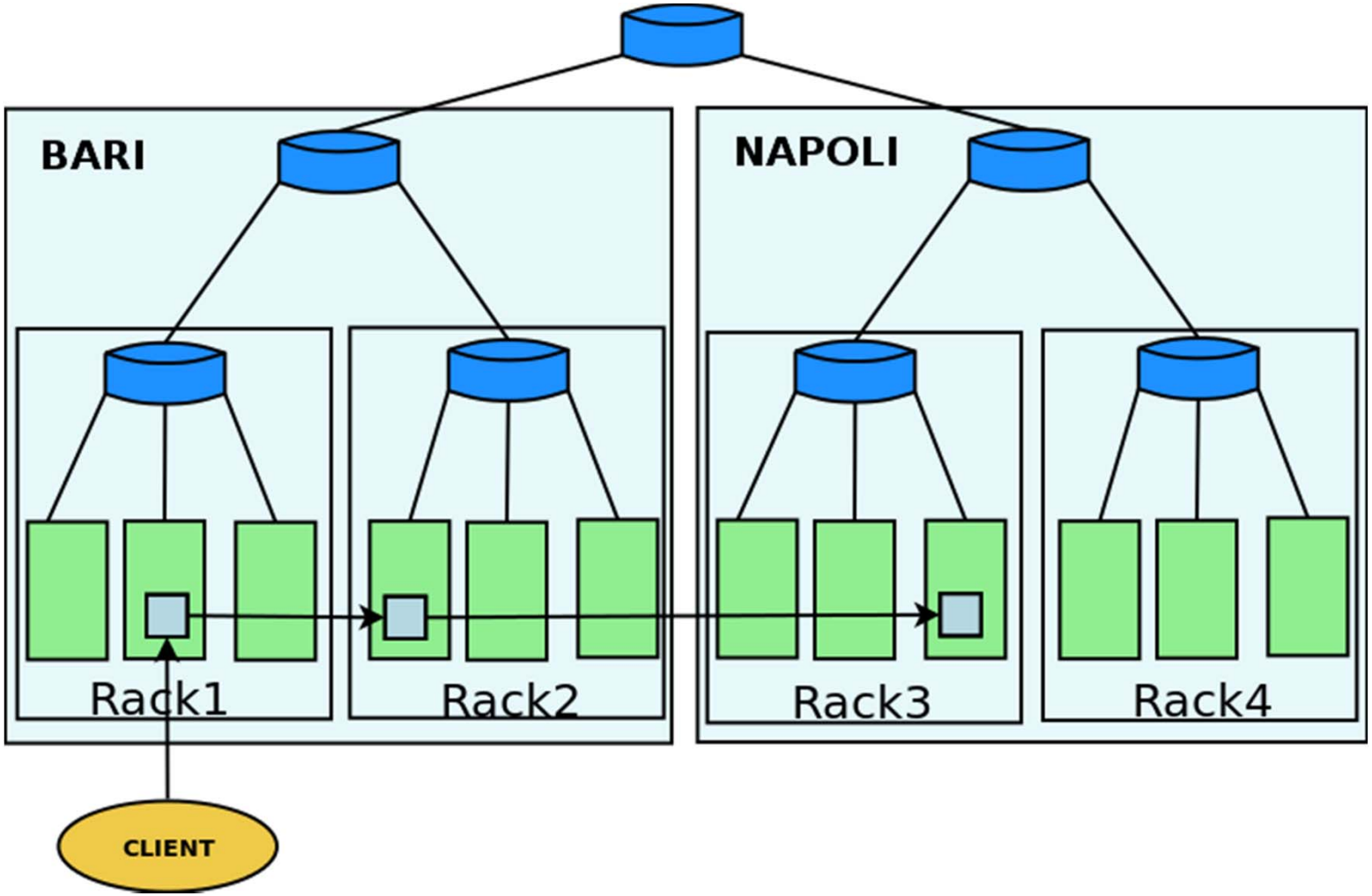
# Hierarchical policy



- Consapevolezza dell'organizzazione gerarchica della topologia di rete
- 2 repliche in rack differenti della farm locale
- 1 replica su un rack di una farm remota
- Tolleranza al fallimento di un intera farm

# Hierarchical policy





# Implementazione delle politiche

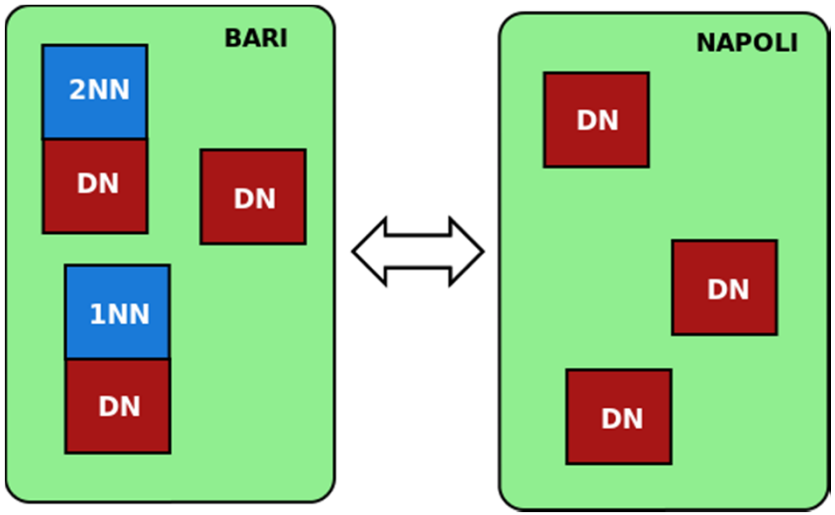


- Classe astratta Java *BlockPlacementPolicy*
  - Implementazione predefinita:
    - *BlockPlacementPolicyDefault*
  - Implementazioni realizzate:
    - *BlockPlacementPolicyOneReplica*
    - *BlockPlacementPolicyHierarchical*
- Politica configurabile attraverso l'apposito parametro

# Cluster HDFS geografico

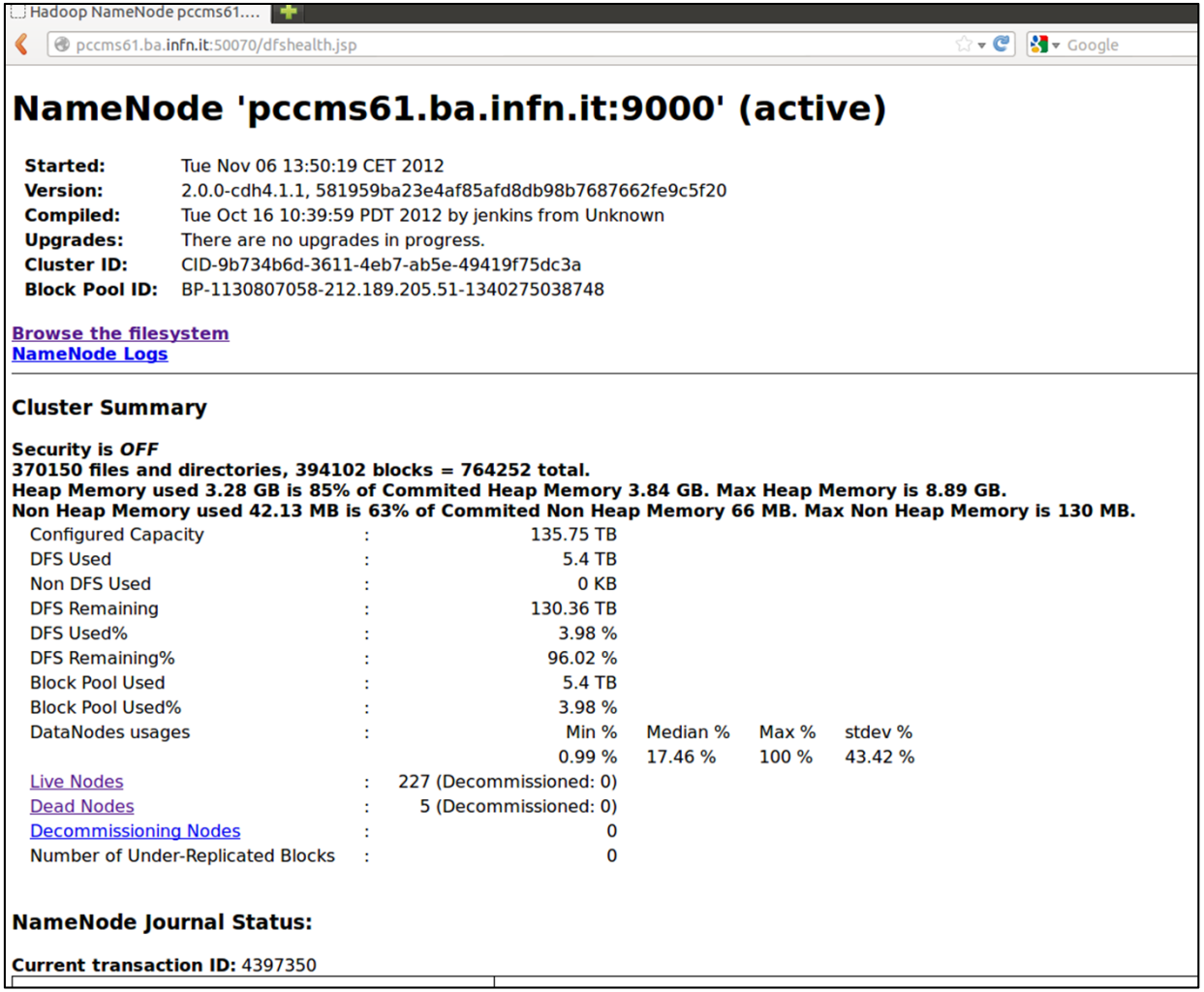


- INFN Bari and INFN Napoli




- Ripetizione dei test di funzionalità
- Test delle politiche realizzate

# Cluster HDFS in (pre)produzione a Bari

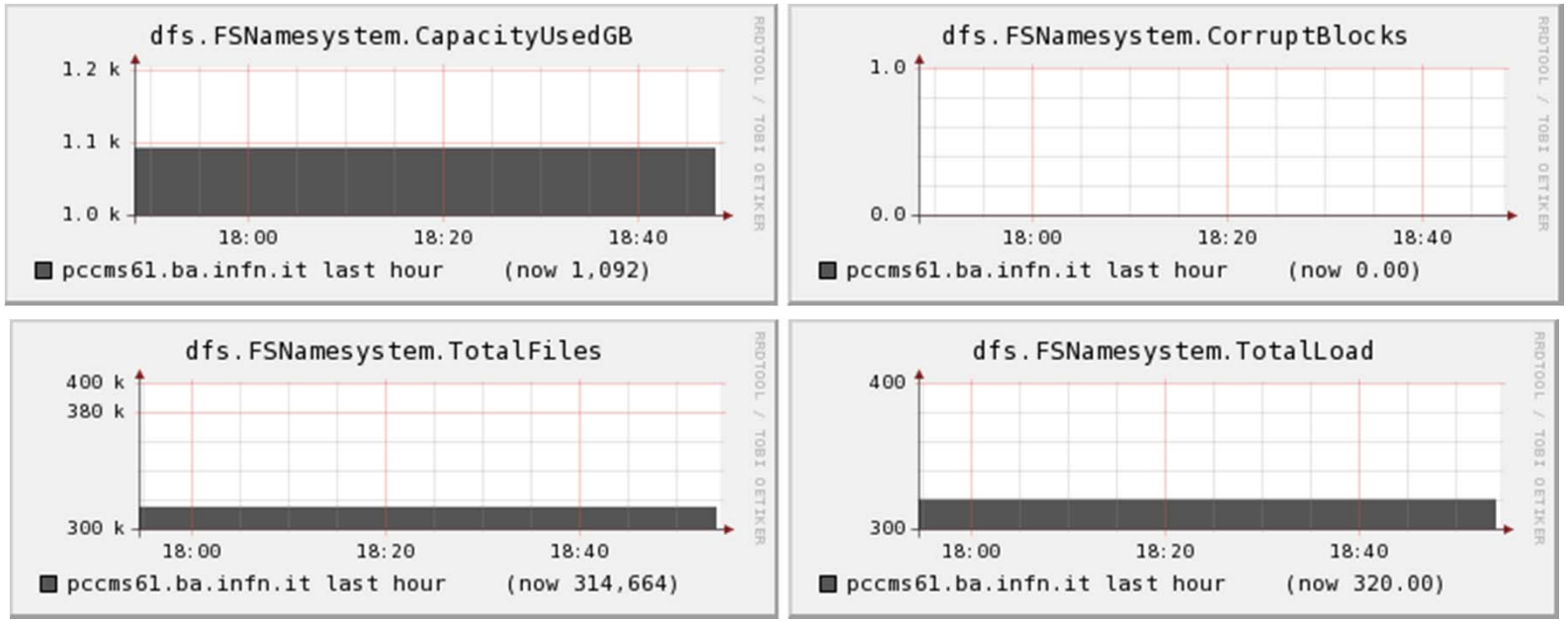


# Monitoring di sistema

# Monitoring con Ganglia



- In produzione sul cluster in (pre)produzione a Bari



# Custom monitoring



- È stato sviluppato un ulteriore sistema di monitoring al fine di tracciare:
  - Locazioni dei blocchi dati
    - Recente cronologia
    - Blocchi danneggiati o persi
  - Operazioni sui blocchi
- Memorizzati in un database MySQL

# Installazione e configurazione automatica dei nodi



- È stata sviluppata una procedura parametrizzata da eseguire su ogni nodo che provvede a:
  - Installazione del software
    - Packages repository
  - Configurazione in base al tipo di nodo
    - formattazione, mount e assegnazione ad HDFS di dischi/partizioni inutilizzati
  - Riavvio del processo quando fallisce
- Riusabile in altri centri di calcolo

# Test di performance

# Test di performance



- È stata misurata la velocità media di scrittura e lettura su 3000 operazioni sui file
  - Eseguiti da un solo client

Test settings

| Parametri | Valori |
|---|---|
| Datanode | Active, passive |
| Block size (MB) | 64, 128, 256 |
| Client | Fuse-dfs |
| Replication factor | 1,2 |
| File dimension (MB) | 4096 |
| Complete dataset | 3K File operations |

# Test di performance: risultati



Velocità media di scrittura (MB/s)

| Block size (MB) | Fatt. Replica 1 | | Fatt. Replica 2 | |
|---|---|---|---|---|
| | Datan. inattivo | Datan. attivo | Datan. inattivo | Datan. attivo |
| 64 | 45,66 | 83,75 | 40,24 | 59,96 |
| 128 | 46,13 | 79,41 | 42,45 | 59,85 |
| 256 | 47,95 | 76,48 | 42,71 | 49,77 |

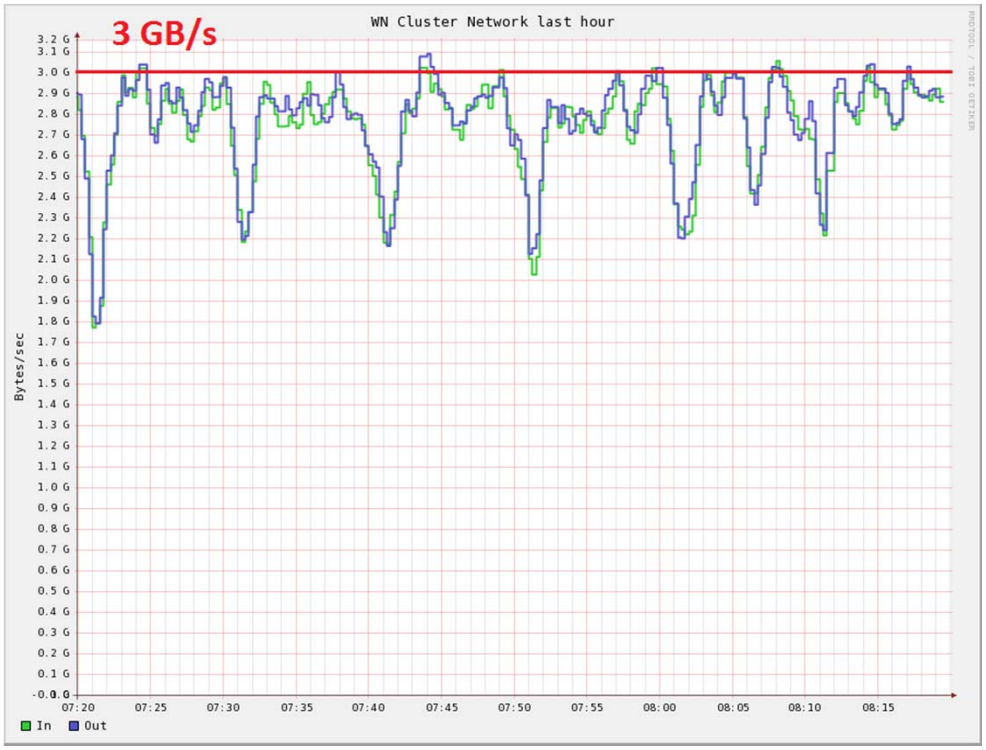
Velocità media di lettura (MB/s)

| Block size (MB) | Fatt. Replica 2 |
|---|---|
| 64 | 59,96 |
| 128 | 59,85 |
| 256 | 49,77 |

# Test di performance: caso reale



- 600 job concorrenti di Pamela che leggono file di ROOT da HDFS via Fuse-dfs
- buon risultato: picchi di oltre 3GB/s



# Sviluppi futuri e conclusioni

# Sviluppi futuri



- Testing del modulo HDFS-RAID
  - Ripristino di blocchi dati danneggiati
  - Riduzione del fattore di replica
    - Conseguente risparmio di spazio disco
- Ampliamento e testing del cluster geografico con un ulteriore sito
  - Collegamento ad elevate prestazioni (10Gbit/s)
  - Test delle politiche di replica
  - Test di performance

# Sviluppi futuri



- Dopo l'espansione dell'infrastruttura di calcolo
  - Testing di long-run di una farm basata su HDFS misurando performance e scalabilità
    - Con 300 nodi di calcolo
    - 500TB di spazio disco
    - Possibilità di eseguire fino a 4000 processi di analisi dati contemporaneamente
- Testing delle configurazioni necessarie a livello di mount point fuse e dimensione blocco per ottimizzare le performance

# Sviluppi futuri



- Implementazione di ulteriori sistemi di monitoring avanzati
  - Per la gestione di un cluster complesso
  - Per la gestione delle failures e di dati di accounting
    - Nagios
    - Database non relazionali
- Testing di sistemi di autenticazione/autorizzazione avanzati
  - Federazioni di sistemi di autenticazione per supportare utenti provenienti da diverse istituzioni

# Conclusioni



- L'attività svolta ha dimostrato
  - Efficacia della feature di affidabilità dei dati
  - Efficacia del comportamento di ripristino automatico
  - L'importanza della possibilità di ottimizzare le politiche di replica dei dati a favore dell'affidabilità e delle prestazioni
  - Feedback positivi ricevuti dai primi utenti utilizzatori

# Grazie per l'attenzione