



**DEVELOPMENT, IMPLEMENTATION AND TESTING OF
DISTRIBUTED STORAGE INFRASTRUCTURE BASED
ON HADOOP-FS OF DATA MANAGEMENT FOR HIGH
RELIABILITY SCIENTIFIC COMPUTING IN
DISTRIBUTED COMPUTING CENTERS**

Workshop GARR - Calcolo e Storage Distribuito
MIUR, Roma – 29-30 Novembre 2012

Giovanni Marzulli - GARR-INFN Bari
Giacinto Donvito - ReCas-INFN Bari
Domenico Diacono – INFN Bari

Unione Europea
Fondo Europeo di Sviluppo Regionale
investiamo nel vostro futuro

ReCaS Rete di Calcolo per SuperB e altre applicazioni

INFN Istituto Nazionale di Fisica Nucleare
PONa3_00052, Avviso 254/Ric

Questo Progetto è stato cofinanziato dal FESR - Fondo Europeo di Sviluppo Regionale

Ricerca e Competitività 2007-2013
www.ponrec.it

Outline

- Use cases
- Hadoop Distributed File System
- Functionality test
- Custom policies development
- System monitoring
- Performance test
- Future works and conclusions

Use cases

- Scientific computation in cluster with shared resources
- High availability of data access with commodity hardware
- Fault tolerance of whole data center



Hadoop Distributed File System

What is Hadoop

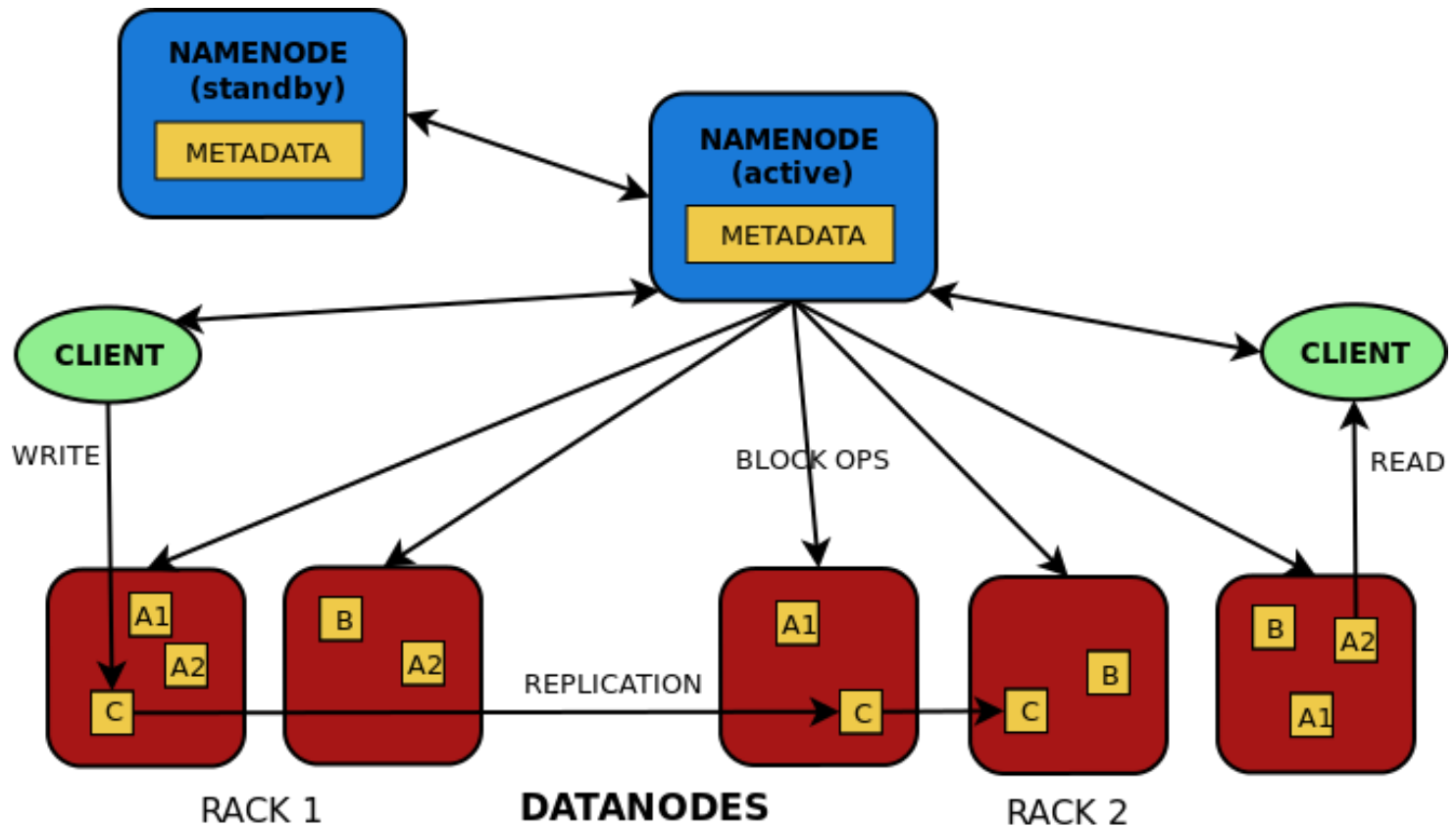
- The Apache Hadoop is an open-source software framework for reliable, scalable, distributed computing.
- Hadoop project includes:
 - Hadoop Distributed File System (HDFS)
 - YARN
 - MapReduce
 - Other related projects such as
 - Avro, Cassandra, Hive, Chukwa, HBase, etc.

HDFS features

Hadoop **D**istributed **F**ile **S**ystem

- Large dataset
- Fault tolerance
- Scalability
- Commodity hardware
- Rack awareness

HDFS Architecture



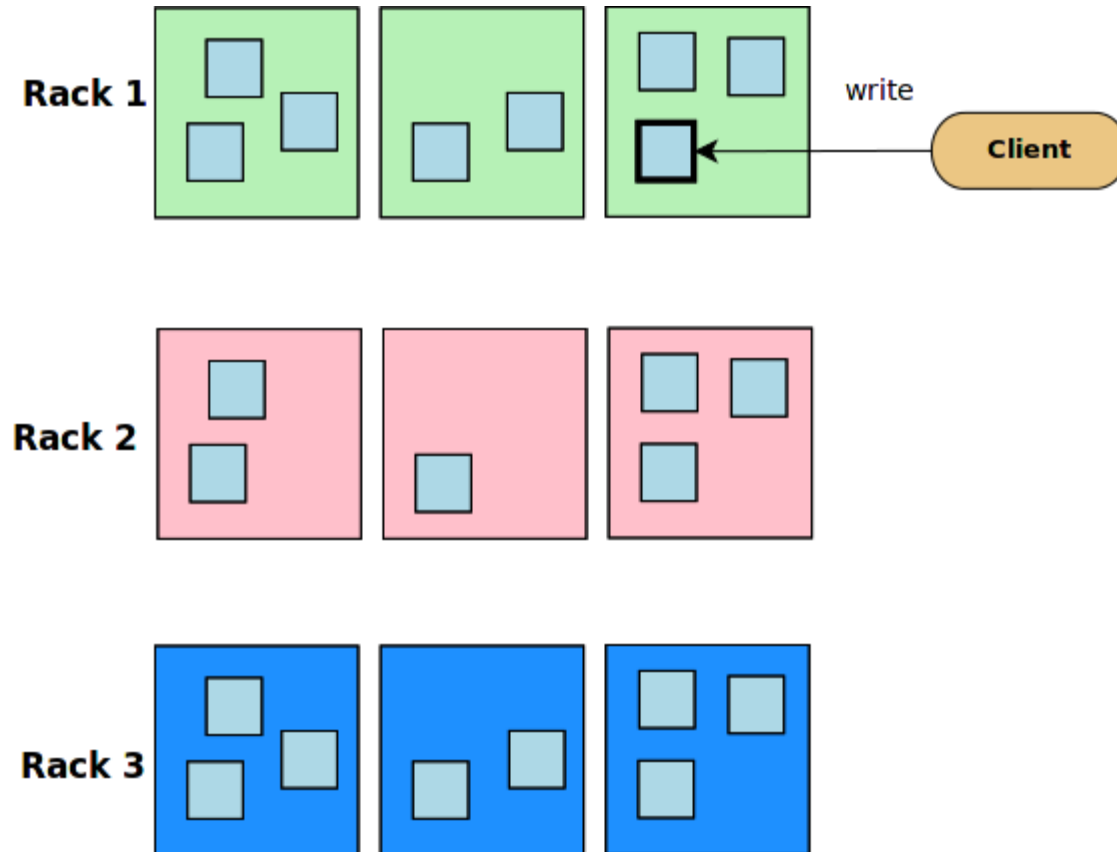
Fault tolerance

- “The primary objective of HDFS is to store data reliably even in the presence of failures.”
- Data splitting
- Data replication
 - Block placement policy
- High Availability namenode

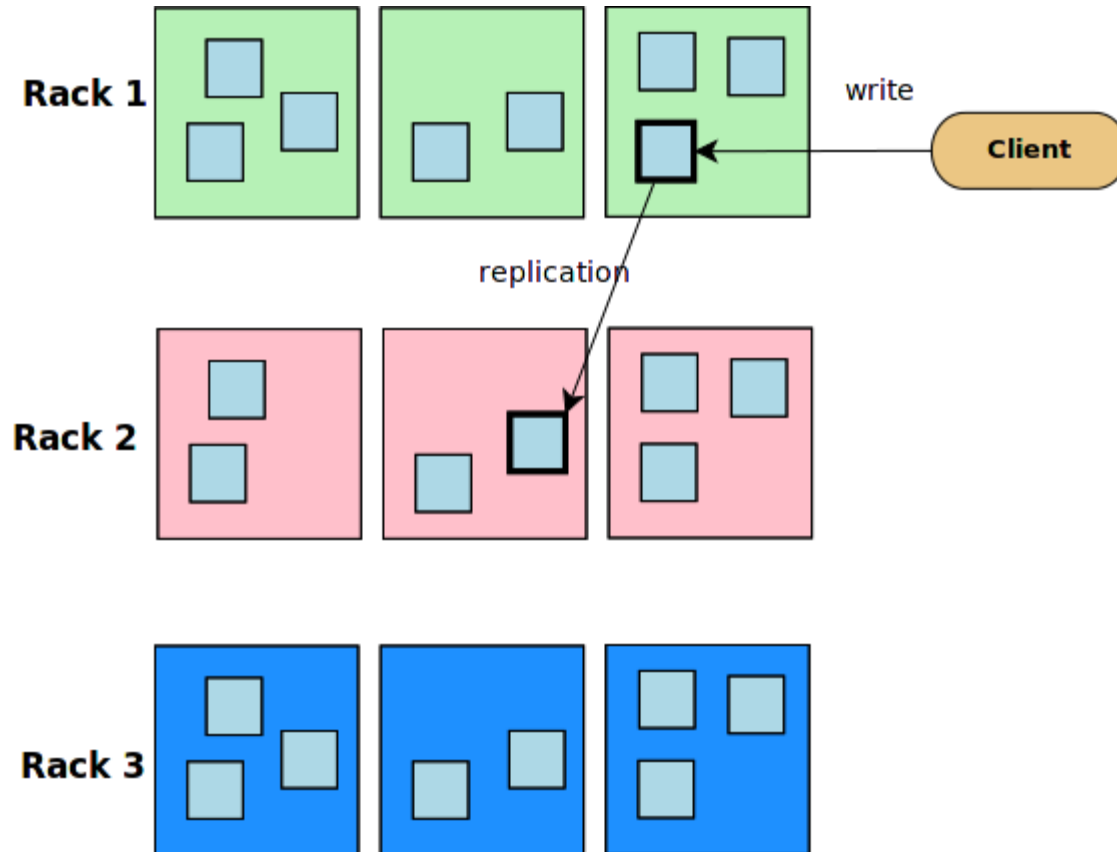
Block placement policies

- Default policy
 - 1 replica on a node of local rack, 2 replicas on different nodes in the same remote rack
- Developed policies
 - One Replica
 - Hierarchical

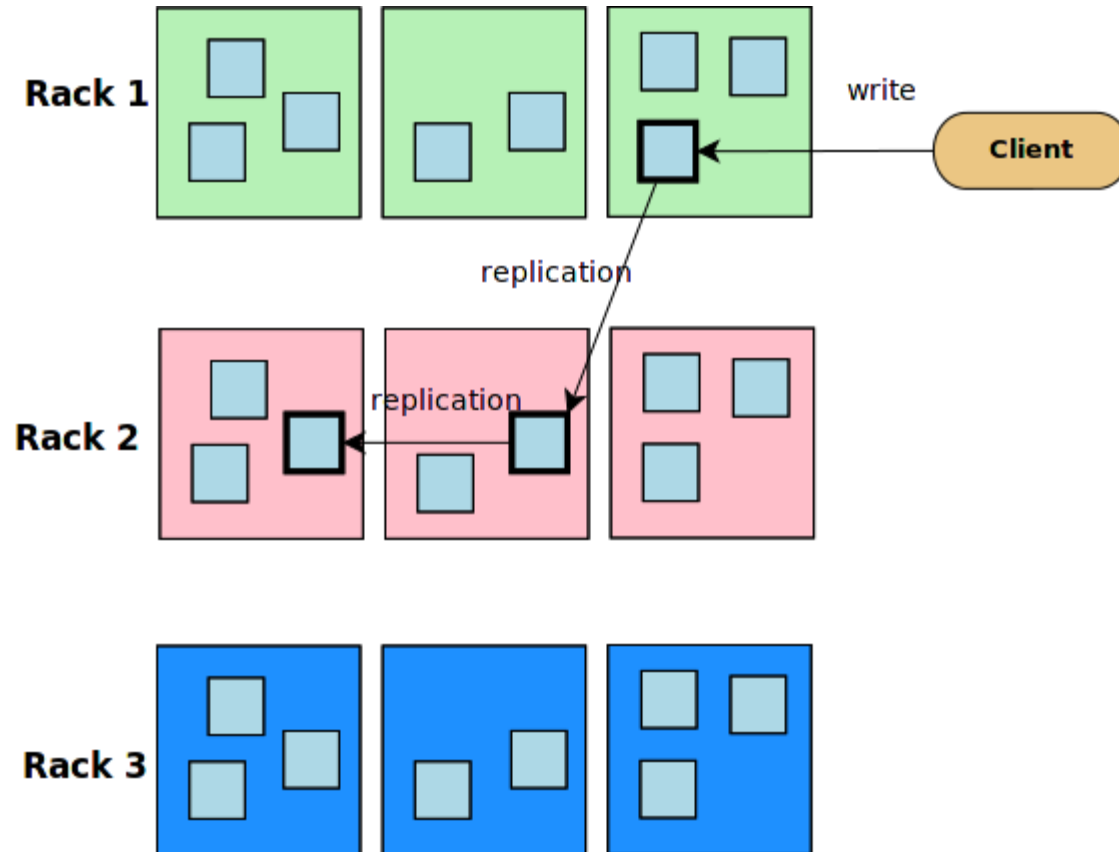
Default placement policy



Default placement policy

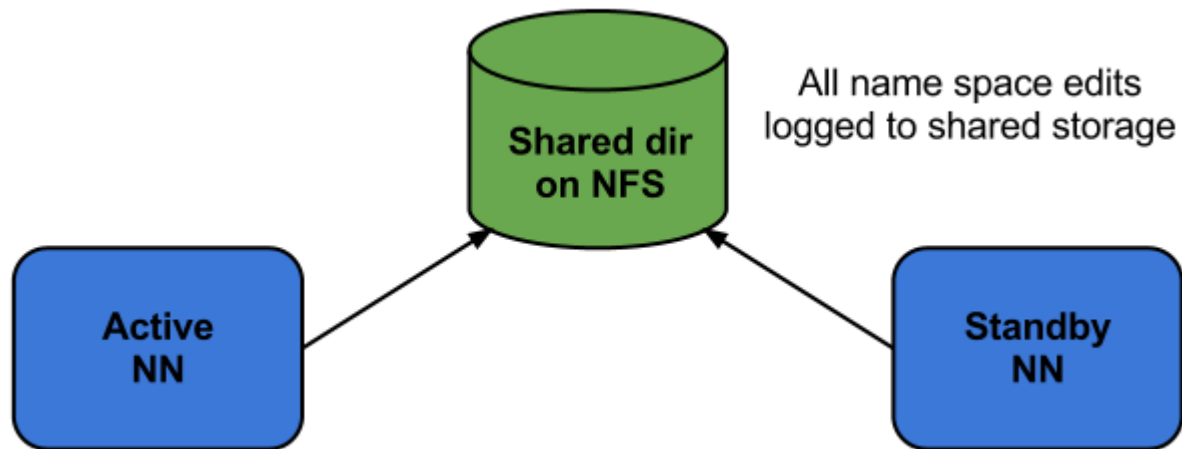


Default placement policy



High Availability namenode

- Metadata synchronization in a shared directory



- Failover:
 - Active → Standby
 - Standby → Active



Functionality test

Installation test

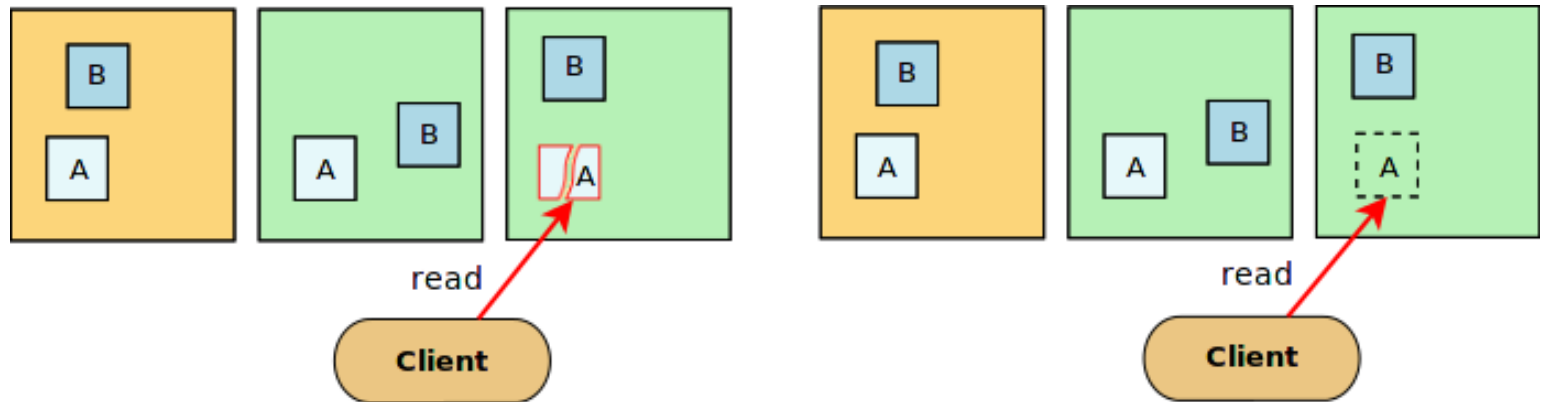
- Small cluster of 7 physical nodes of INFN-Bari site
 - Different OS, different hardware, different networks with firewall
- 7 datanodes
- 1 primary namenode
- 1 secondary namenode
- From Hadoop 0.20 to 2.0 (CDH4.1)

Namenodes test

- Corrupted or lost metadata
 - Recovery from secondary namenode:
 - `hadoop-daemon start namenode -importCheckpoint`
- Namenode down
 - Waiting of clients and datanodes
 - Failover:
 - `hdfs haadmin -failover nn1 nn2`

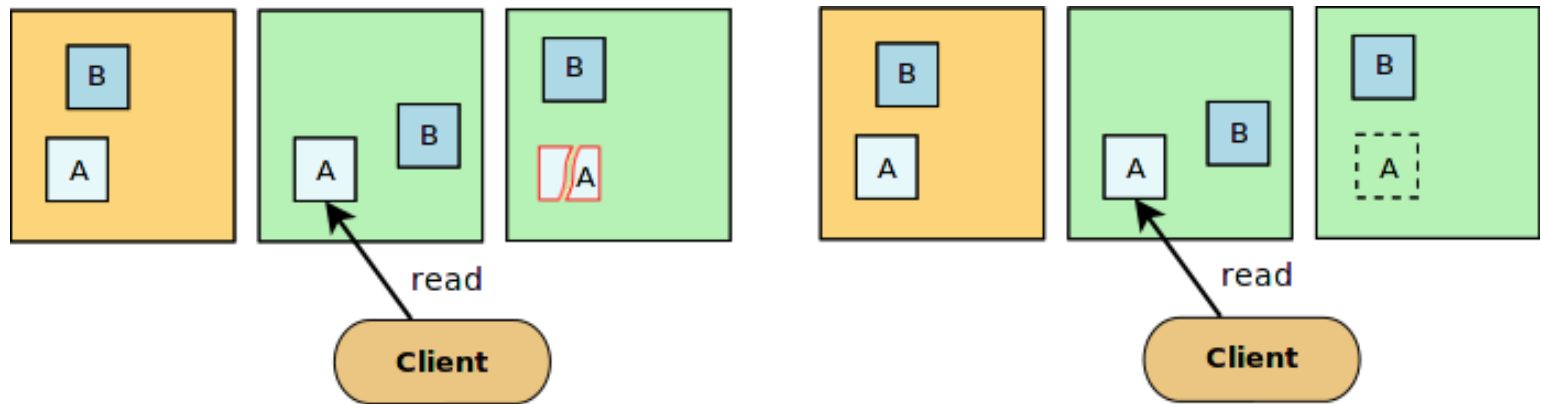
Datanodes test

- Lost or corruption data



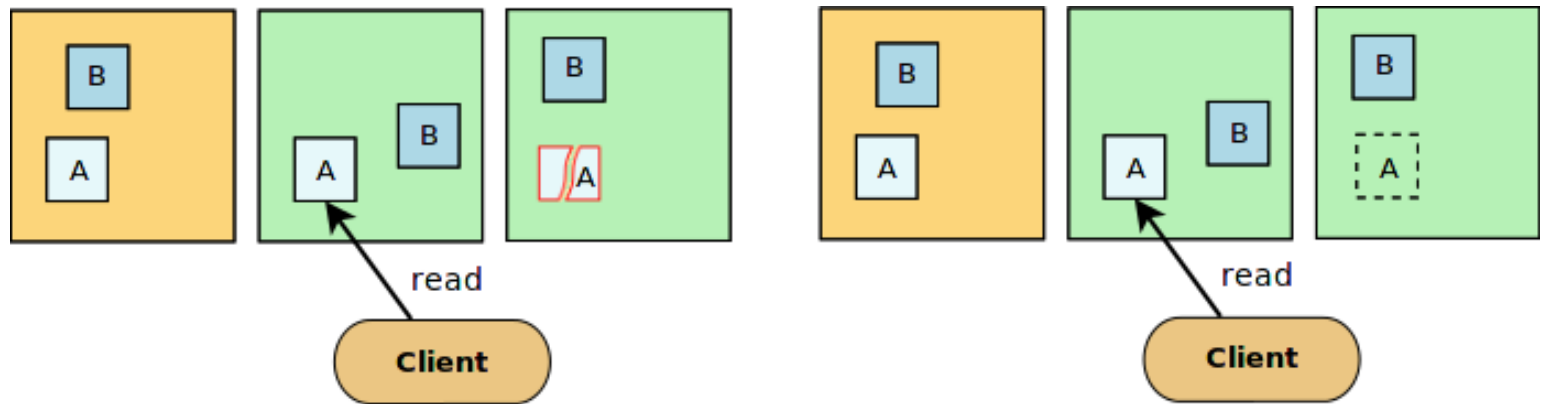
Datanodes test

- Lost or corruption data

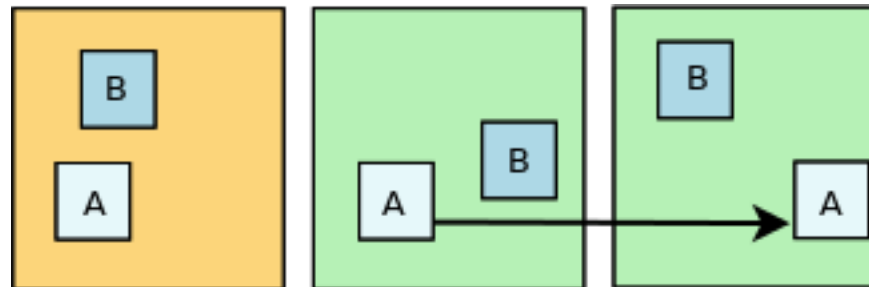


Datanodes test

- Lost or corruption data



- Automatic recovery



Datanodes test

- Under-replicated blocks
 - After datanode failure
- Over-replicated blocks
 - After recovery and restart datanode
- Mis-replicated blocks
 - Policy violation
- Datanode failure during writing/reading
 - Switch to other live nodes
- Workload balance

HDFS clients

- **Default Hadoop client**

- `bin/hadoop fs -put testfile.dat /marzulli/`
- `bin/hadoop fs -get /marzulli/testfile.dat ./`

- **Fuse-Dfs client**

- **Mount HDFS in userspace**

- `cp testfile.dat /mnt/hadoop/marzulli/`
- `cp /mnt/hadoop/marzulli/testfile.dat ./`

Kerberos authentication

- No Hadoop default security
- Node and user authentication by Kerberos
 - Keytab
 - Ticket
- User authorization by Hadoop file permissions

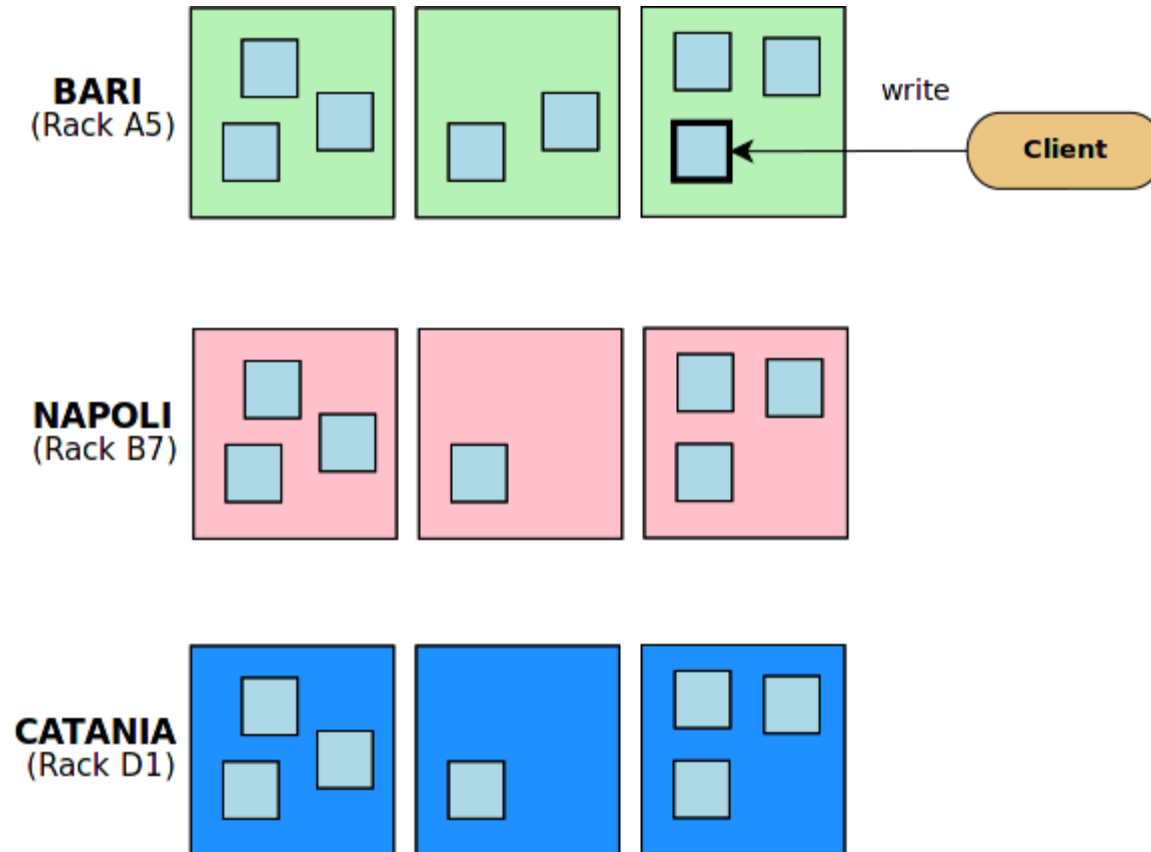


Block placement policies development

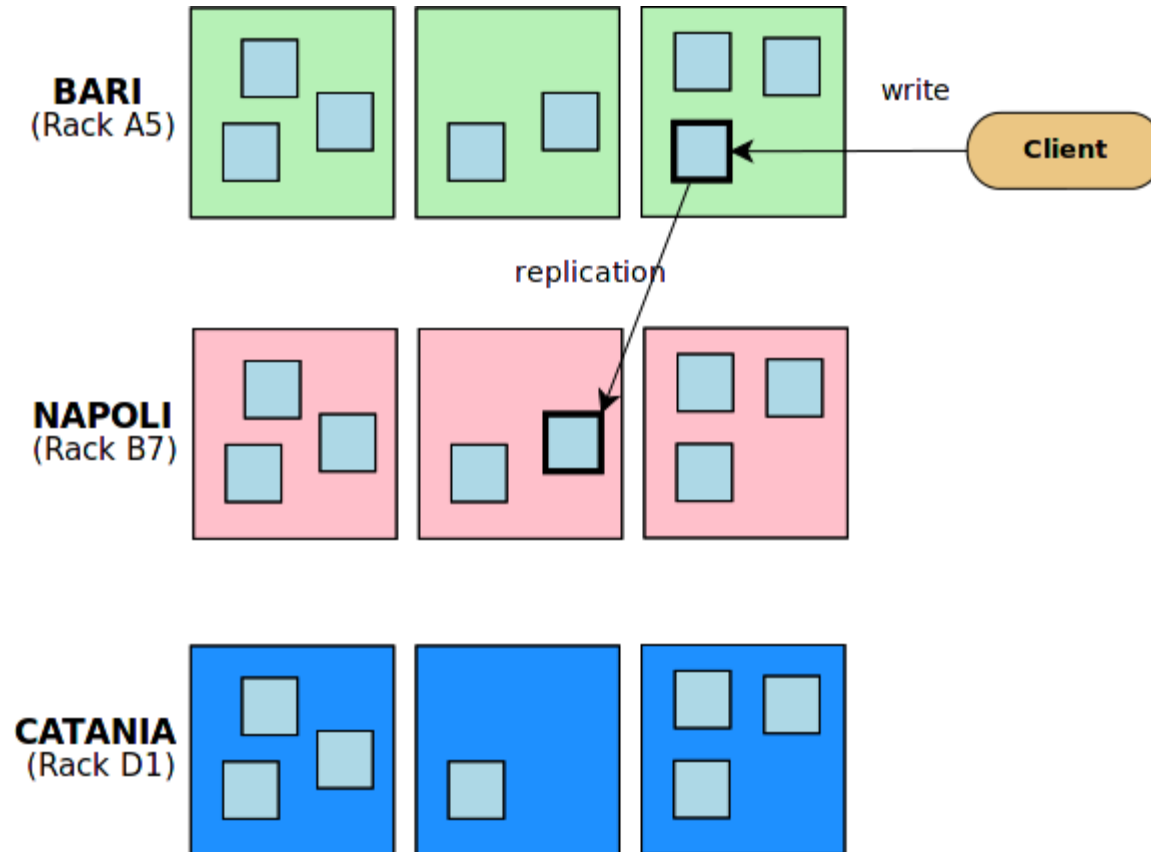
One replica placement policy

- 1 replica per rack
- More reliability
 - 2 racks fault tolerant (if replication factor is 3)
- Geographical cluster
 - More data distribution
 - Less read cost
 - Reading from nearest replica
 - More write cost
 - More data transmission

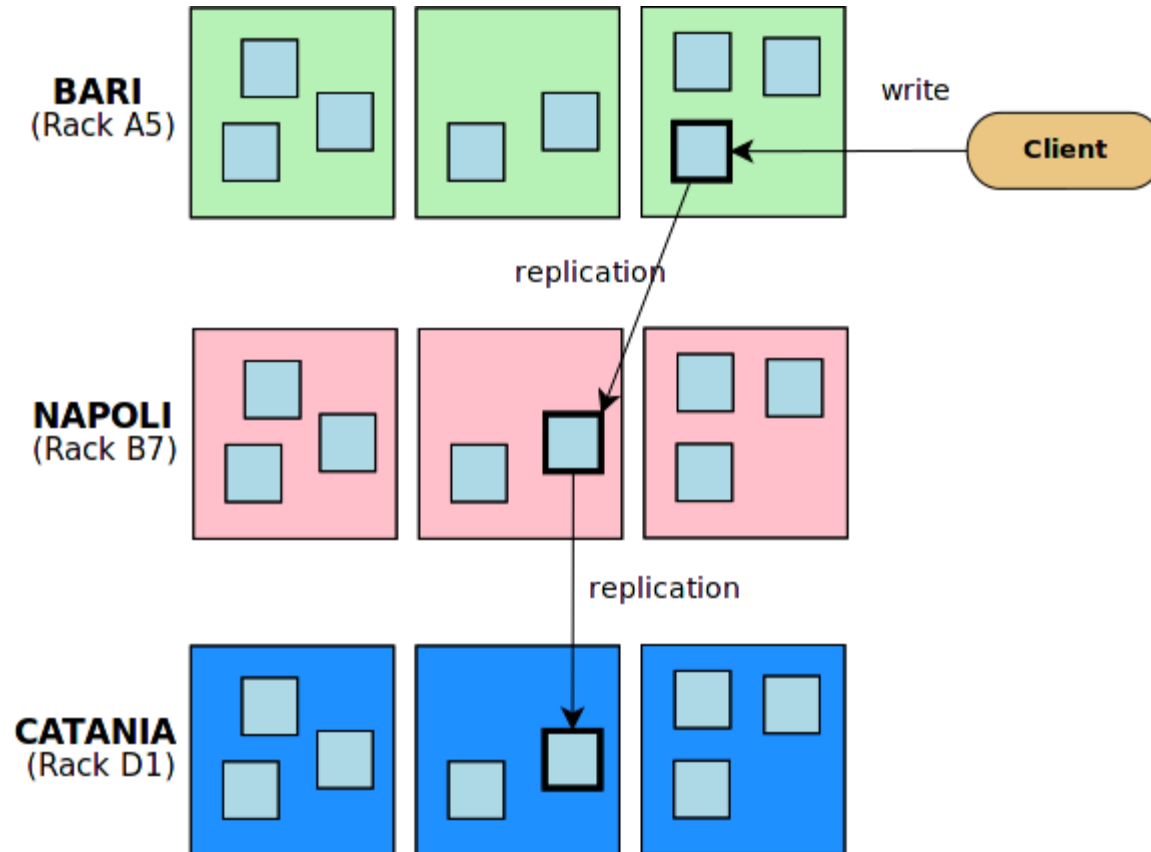
One replica placement policy



One replica placement policy



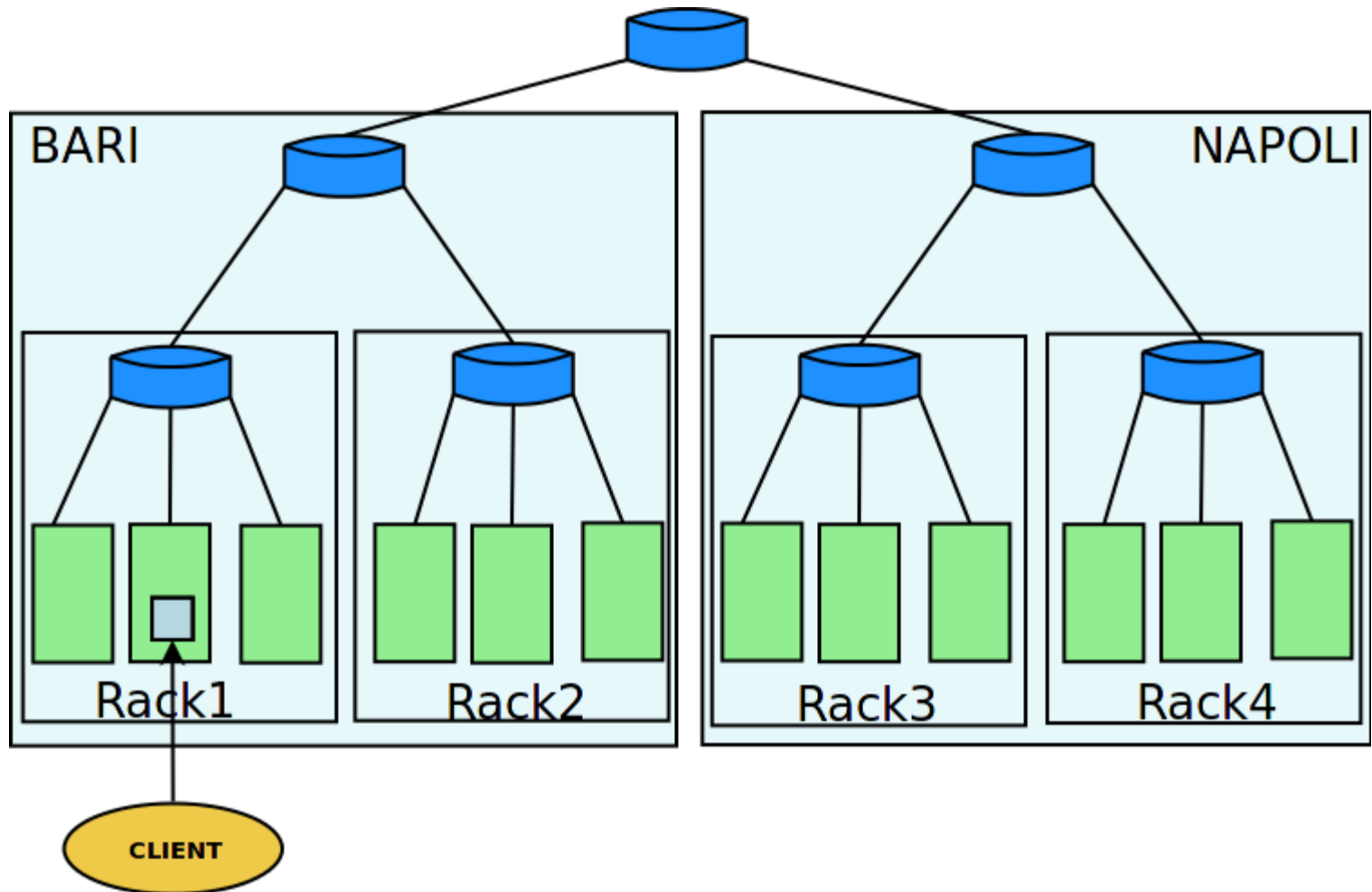
One replica placement policy



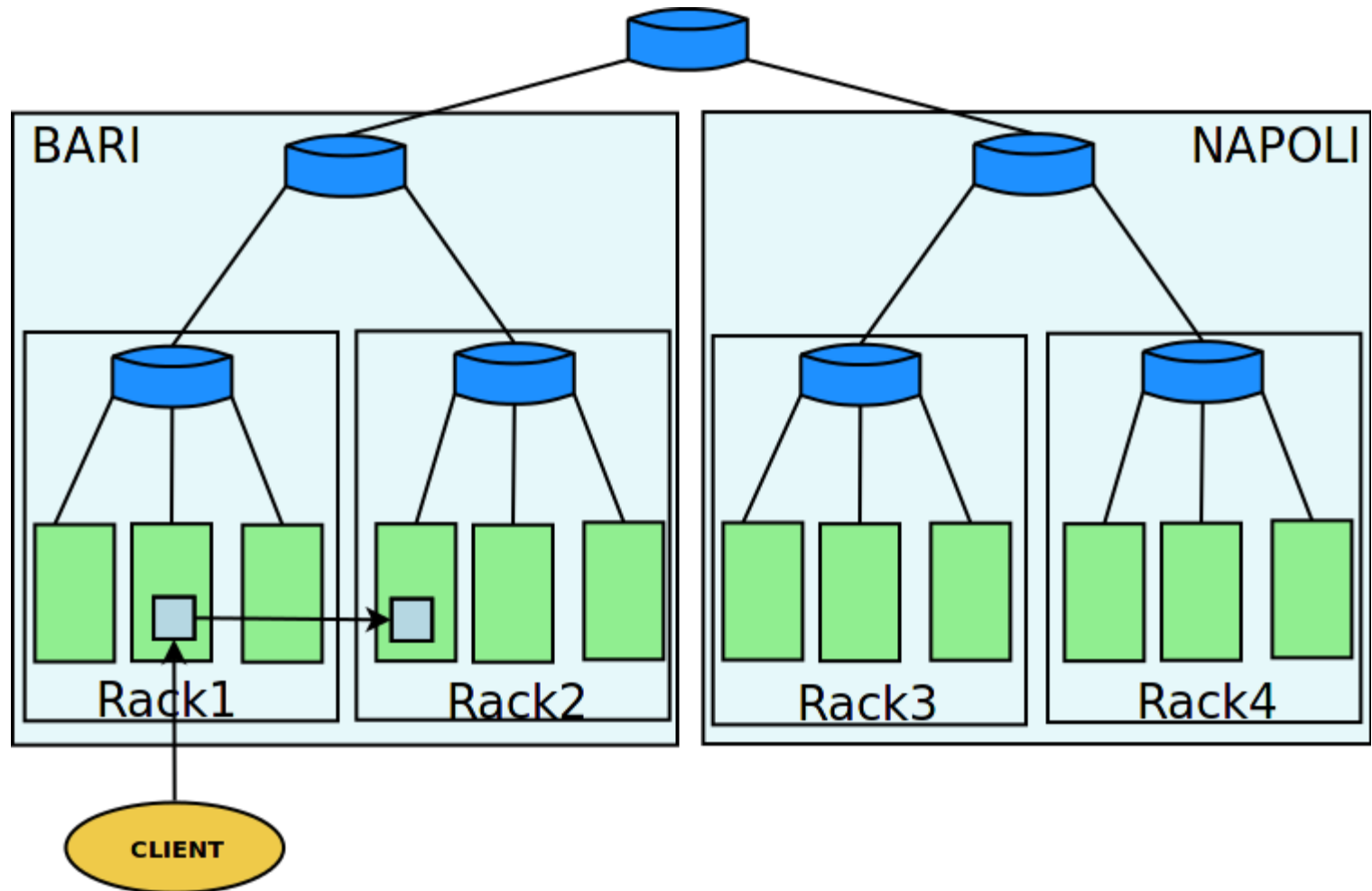
Hierarchical placement policy

- Awareness of hierarchical network topology
- 2 replicas in local farm but in different racks
- 1 replica on a rack of remote farm
- Tolerance of whole farm fault

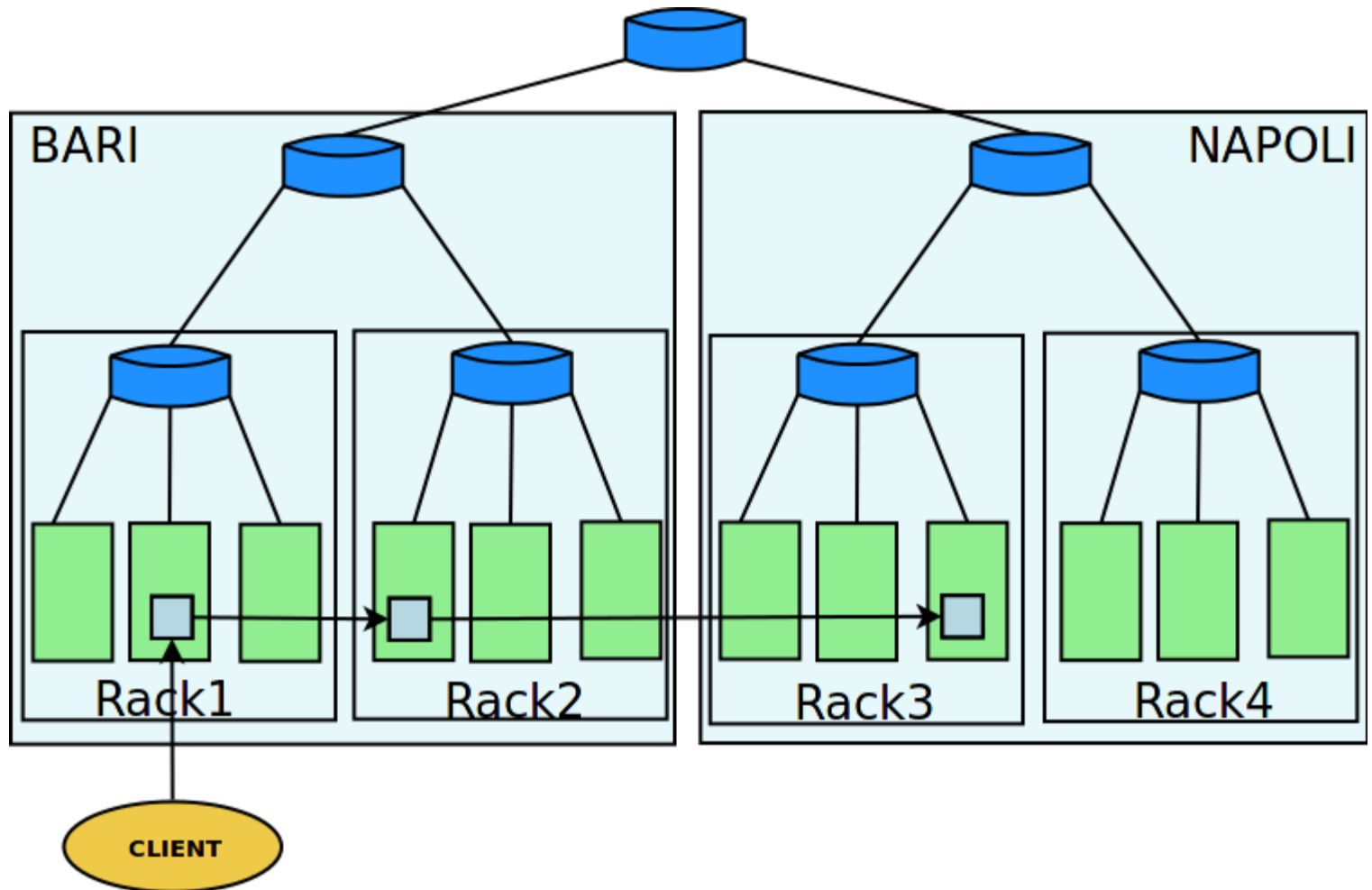
Hierarchical placement policy



Hierarchical placement policy



Hierarchical placement policy

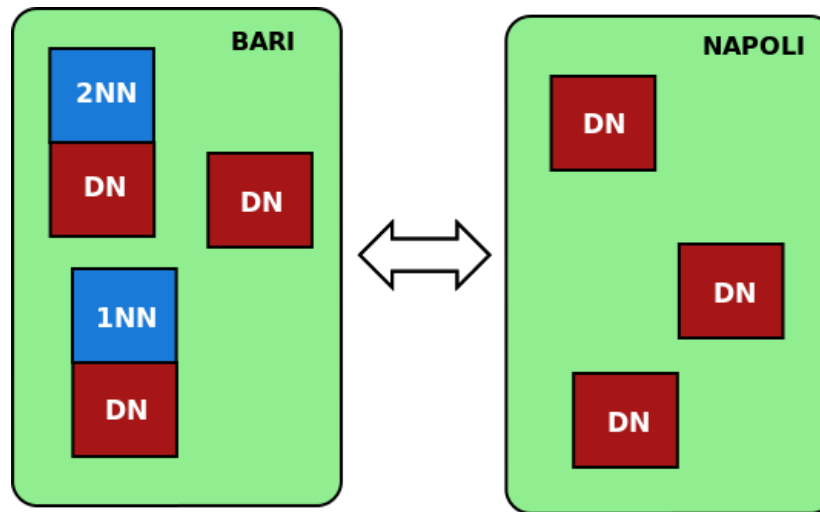


Implementation of custom policies

- *BlockPlacementPolicy* Java abstract class
 - Default implementation:
 - *BlockPlacementPolicyDefault*
 - Custom implementations:
 - *BlockPlacementPolicyOneReplica*
 - *BlockPlacementPolicyHierarchical*
- Policy selectable in the configuration file

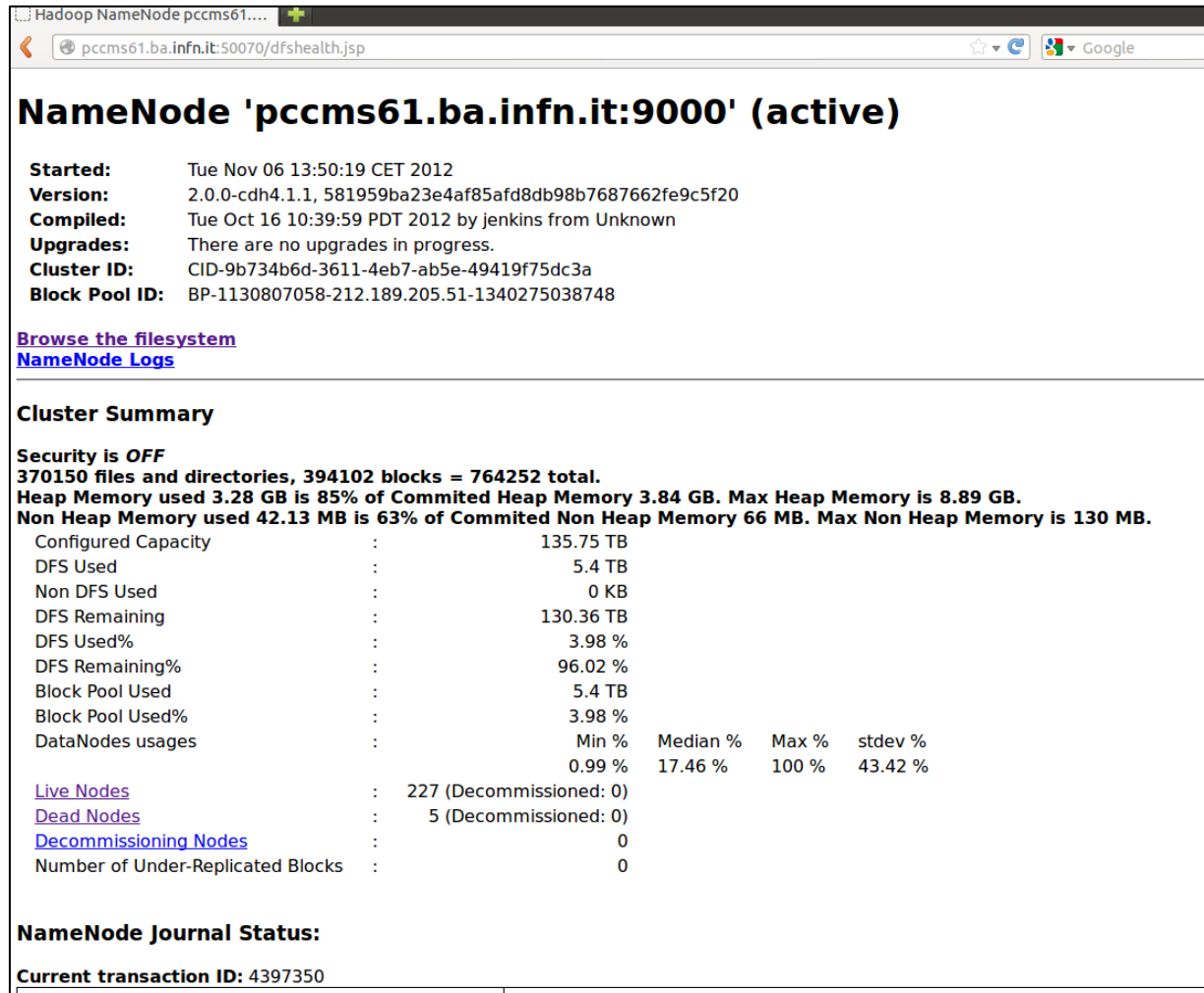
Geographic cluster

- INFN Bari and INFN Napoli (ReCaS sites)



- Functionality test
- Custom policies test

INFN Bari (pre)production cluster



The screenshot shows the Hadoop NameNode web interface for the cluster 'pccms61.ba.infn.it:50070'. The page title is 'NameNode 'pccms61.ba.infn.it:9000' (active)'. It displays various status metrics and a cluster summary.

NameNode 'pccms61.ba.infn.it:9000' (active)

Started: Tue Nov 06 13:50:19 CET 2012
Version: 2.0.0-cdh4.1.1, 581959ba23e4af85afd8db98b7687662fe9c5f20
Compiled: Tue Oct 16 10:39:59 PDT 2012 by jenkins from Unknown
Upgrades: There are no upgrades in progress.
Cluster ID: CID-9b734b6d-3611-4eb7-ab5e-49419f75dc3a
Block Pool ID: BP-1130807058-212.189.205.51-1340275038748

[Browse the filesystem](#)
[NameNode Logs](#)

Cluster Summary

Security is OFF
370150 files and directories, 394102 blocks = 764252 total.
Heap Memory used 3.28 GB is 85% of Committed Heap Memory 3.84 GB. Max Heap Memory is 8.89 GB.
Non Heap Memory used 42.13 MB is 63% of Committed Non Heap Memory 66 MB. Max Non Heap Memory is 130 MB.

Configured Capacity	:	135.75 TB			
DFS Used	:	5.4 TB			
Non DFS Used	:	0 KB			
DFS Remaining	:	130.36 TB			
DFS Used%	:	3.98 %			
DFS Remaining%	:	96.02 %			
Block Pool Used	:	5.4 TB			
Block Pool Used%	:	3.98 %			
DataNodes usages	:	Min %	Median %	Max %	stdev %
	:	0.99 %	17.46 %	100 %	43.42 %

[Live Nodes](#) : 227 (Decommissioned: 0)
[Dead Nodes](#) : 5 (Decommissioned: 0)
[Decommissioning Nodes](#) : 0
Number of Under-Replicated Blocks : 0

NameNode Journal Status:
Current transaction ID: 4397350

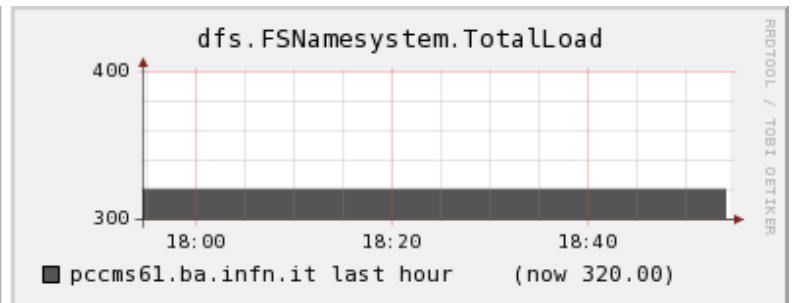
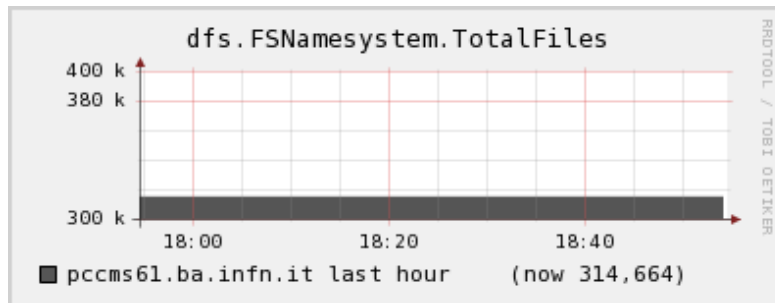
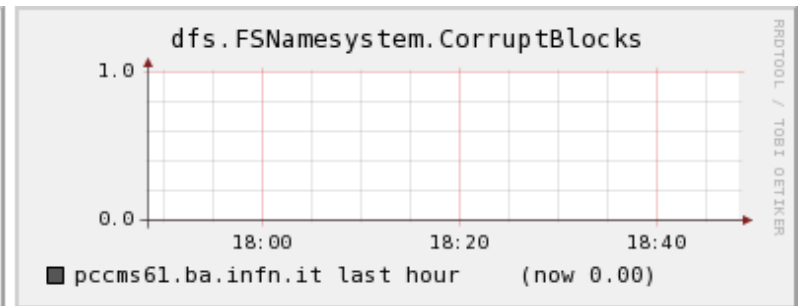
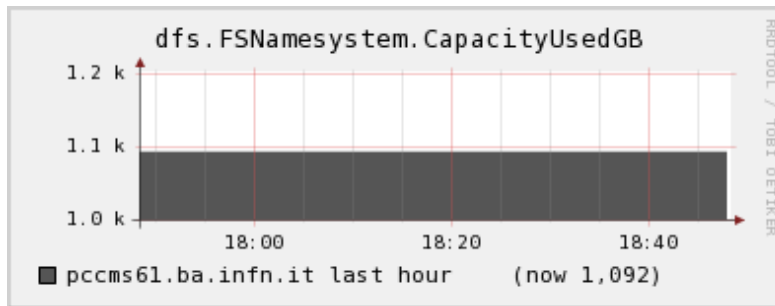


System monitoring

Ganglia monitoring

- Configured in the INFN-Bari cluster

Some metrics



Custom monitoring

- We developed a monitoring system in order to track:
 - Locations of blocks placements
 - Recent blocks history
 - Corrupted or missing blocks
 - Blocks operations
- Stored in a MySQL database

Automatic node installation and configuration

- We developed a parameterized script procedure to run on each node, that provide:
 - Software installation
 - Packages repository
 - Configuration based on nodetype
 - formatting, mounting and assigning unused disks/partitions to HDFS
 - Process restart if node falls
- Reusable in other sites



Performance test

Performance test

- We measured writing and reading mean rates on 3000 file operations
 - One client running

Test settings

Parameters	Values
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

Performance test: statistical results

Writing mean rates (MB/s)

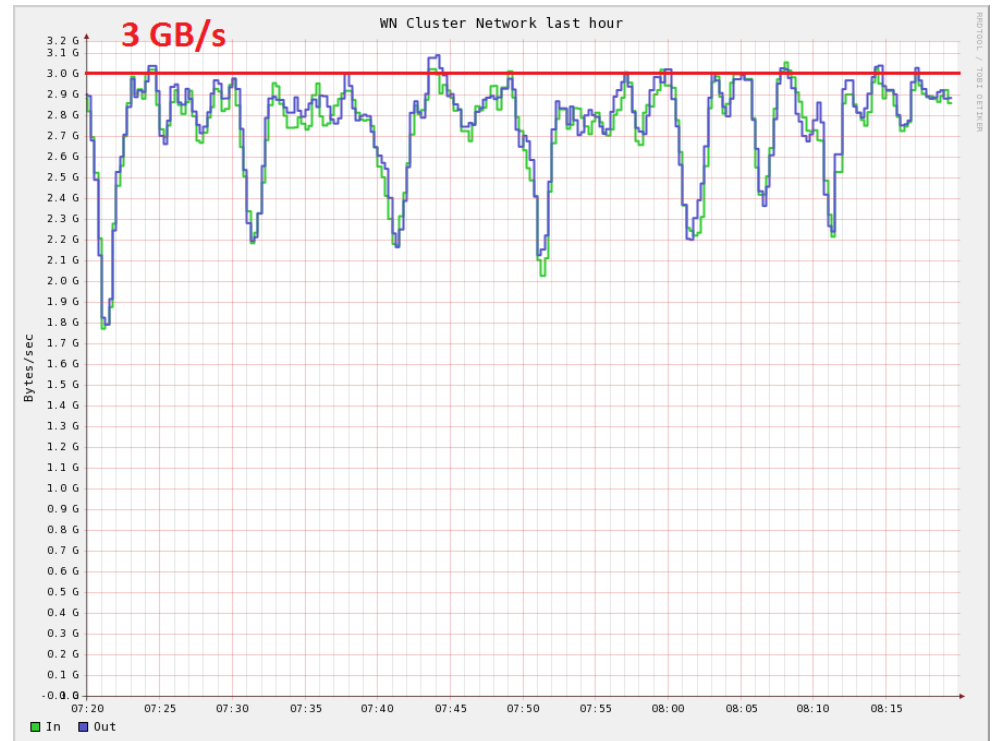
Block size	Replication Factor 1		Replication Factor 2	
	Passive datan.	Active datan.	Passive datan.	Active datan.
64MB	45,66	83,75	40,24	59,96
128MB	46,13	79,41	42,45	59,85
256MB	47,95	76,48	42,71	49,77

Reading mean rates (MB/s)

Block size	Replication Factor 2
64MB	62,29
128MB	60,87
256MB	61,84

Performance test: real case

- 600 jobs of Pamela simultaneously reading ROOT files from HDFS via Fuse-dfs
- we obtained good results: peaks of over 3GB/s





Future works & conclusions

Future works

- Infrastructure expansion
 - Scalability test
 - Long-run test on cluster up to 300 nodes and 500TB of disk space
 - Up to 4000 jobs simultaneously running
- Opening of pre-production cluster to other experiments/VOs (as CMS)
- Geographic test of 3 sites-cluster
 - Add another ReCaS site to the existing cluster
- Research of optimal configuration
 - Block size
 - Fuse-dfs

Conclusions

- Strength of data reliability
- Strength of automatic recovery behavior
- Optimization of data placement in order to increase reliability and performance
- Positive feedback by first real users



Thanks