Hortonworks

HDFS Federation

Sanjay Radia

Founder and Architect @ Hortonworks

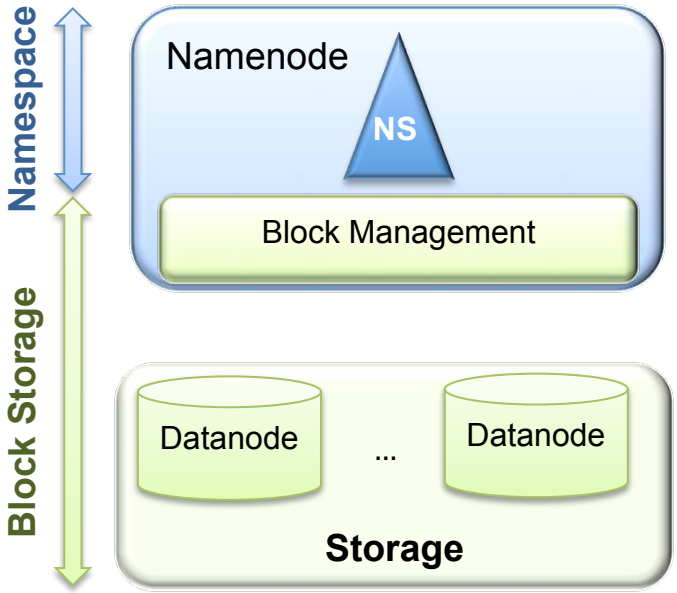
About Me

- Apache Hadoop Committer and Member of Hadoop PMC
- Architect of core-Hadoop @ Yahoo
  - Focusing on HDFS, MapReduce scheduler, Compatibility, etc.
- PhD in Computer Science from the University of Waterloo, Canada

Agenda

- HDFS Background
- Current Limitations
- Federation Architecture
- Federation Details
- Next Steps
- Q&A

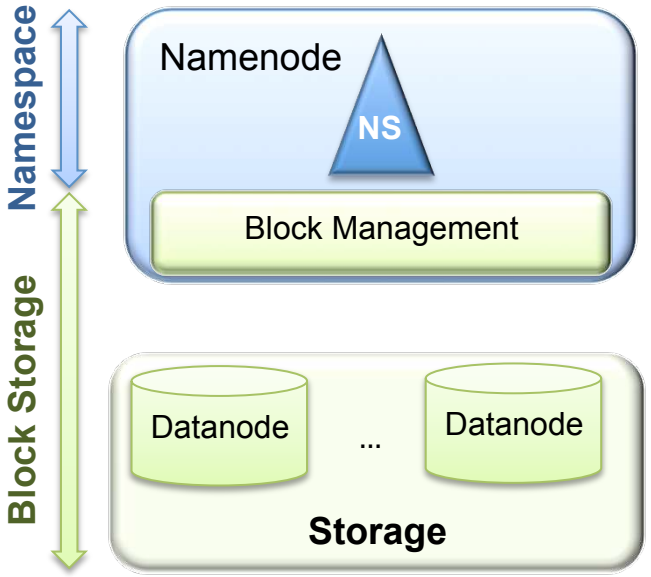
HDFS Architecture



Two main layers

- **Namespace**
  - Consists of dirs, files and blocks
  - Supports create, delete, modify and list files or dirs operations
- **Block Storage**
  - Block Management
    - Datanode cluster membership
    - Supports create/delete/modify/get block location operations
    - Manages replication and replica placement
  - Storage - provides read and write access to blocks

HDFS Architecture



Implemented as

- Single ***Namespace Volume***
  - Namespace Volume = Namespace + Blocks
- Single namenode with a namespace
  - Entire namespace is in memory
  - Provides Block Management
- Datanodes store block replicas
  - Block files stored on local file system

Limitation - Isolation

Poor Isolation

- All the tenants share a single namespace
  - Separate volume for tenants is desirable
- Lacks separate namespace for different application categories or application requirements
  - Experimental apps can affect production apps
  - Example - HBase could use its own namespace

Limitation - Scalability

Scalability

- Storage scales horizontally - namespace doesn't
- Limited number of files, dirs and blocks
  - 250 million files and blocks at 64GB Namenode heap size
    - Still **a very large cluster**
    - Facebook clusters are sized at ~70 PB storage

Performance

- File system operations throughput limited by a single node
  - 120K read ops/sec and 6000 write ops/sec
    - Support 4K clusters easily
    - Easily scalable to 20K write ops/sec by code improvements

Limitation – Tight Coupling

Namespace and Block Management are distinct layers

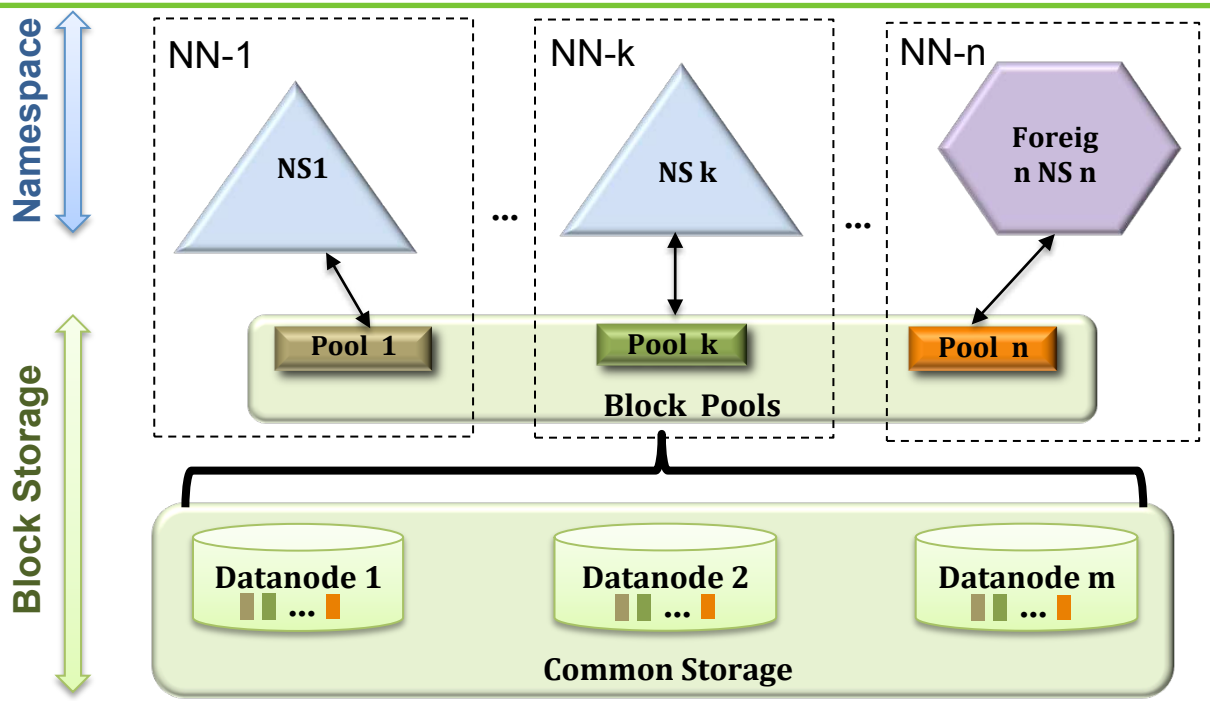
- Tightly coupled due to co-location
- Separating the layers makes it easier to evolve each layer
- Separating services
  - Scaling block management independent of namespace is simpler
  - Simplifies Namespace and scaling it,

Block Storage could be a generic service

- Namespace is one of the applications to use the service
- Other services can be built directly on Block Storage
  - HBase
  - MR Tmp
  - Foreign namespaces

Stated Problem

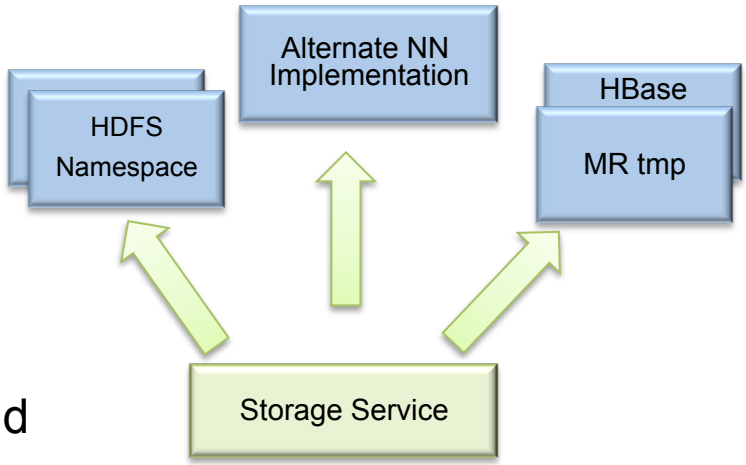
HDFS Federation



- Multiple **independent** Namenodes and Namespace Volumes in a cluster
  - Namespace Volume = Namespace + Block Pool
- Block Storage as generic storage service
  - Set of blocks for a Namespace Volume is called a ***Block Pool***
  - DNs store blocks for all the Namespace Volumes – no partitioning

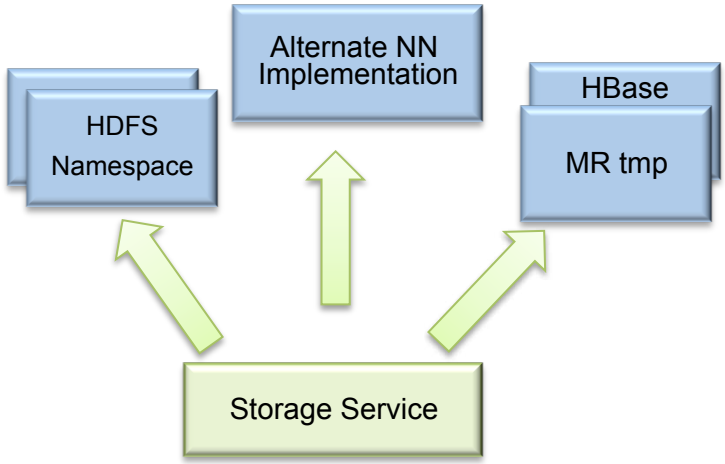
Key Ideas & Benefits

- Distributed Namespace: Partitioned across namenodes
  - Simple and Robust due to independent masters
    - Each master serves a namespace volume
    - Preserves namenode stability – little namenode code change
  - Scalability – 6K nodes, 100K tasks, 200PB and 1 billion files



Key Ideas & Benefits

- Block Pools enable generic storage service
  - Enables Namespace Volumes to be independent of each other
  - Fuels innovation and Rapid development
    - New implementations of file systems and Applications on top of block storage possible
    - New block pool categories – tmp storage, distributed cache, small object storage



- In future, move Block Management out of namenode to separate set of nodes
  - Simplifies namespace/application implementation
    - Distributed namenode becomes significantly simpler

HDFS Federation Details

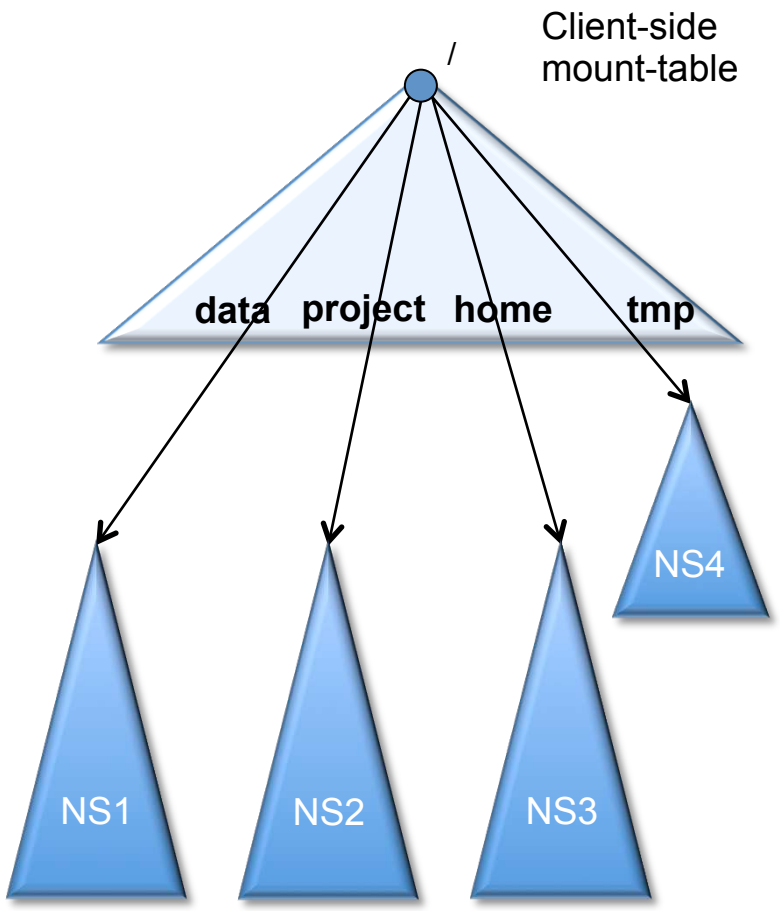
- Simple design
  - Little change to the Namenode, most changes in Datanode, Config and Tools
  - Core development in 4 months
  - Namespace and Block Management remain in Namenode
    - Block Management could be moved out of namenode in the future
- Little impact on existing deployments
  - Single namenode configuration runs as is
- Datanodes provide storage services for all the namenodes
  - Register with all the namenodes
  - Send periodic heartbeats and block reports to all the namenodes
  - Send block received/deleted for a block pool to corresponding namenode

HDFS Federation Details

- Cluster Web UI for better manageability
  - Provides cluster summary
  - Includes namenode list and summary of namenode status
  - Decommissioning status
- Tools
  - Decommissioning works with multiple namespace
  - Balancer works with multiple namespaces
    - Both Datanode storage or Block Pool storage can be balanced
- Namenode can be added/deleted in Federated cluster
  - No need to restart the cluster
- Single configuration for all the nodes in the cluster

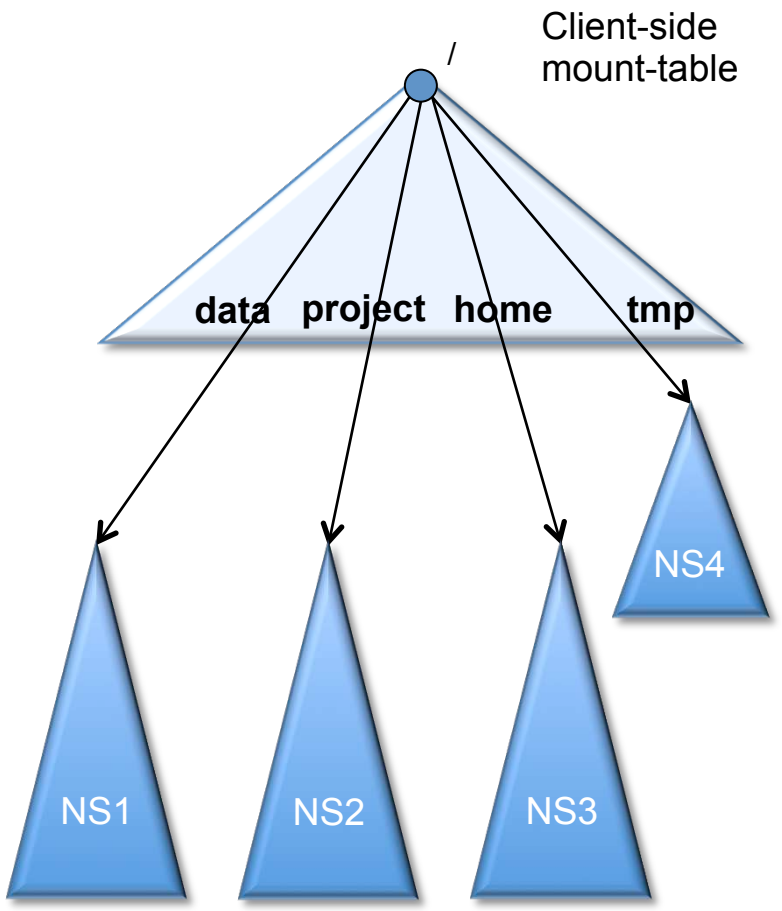
Managing Namespaces

- Federation has multiple namespaces
  - *Don't you need a single global namespace?*
  - Some tenants want private namespace
  - Global? Key is to share the data and the names used to access the data
- A single global namespace is one way share



Managing Namespaces

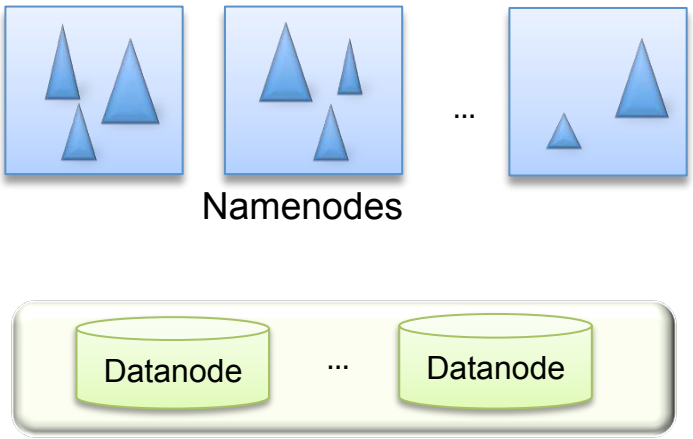
- Client-side mount table is another way to share
  - Shared mount-table => “global” shared view
  - Personalized mount-table => per-application view
    - Share the data that matter by mounting it
- Client-side implementation of mount tables
  - No single point of failure
  - No hotspot for root and top level directories



Next Steps

- Complete separation of namespace and block management layers
  - Block storage as generic service
- Partial namespace in memory for further scalability
- Move partial namespace from one namenode to another
  - Namespace operation - no data copy

Next Steps



- Namenode as a container for namespaces
  - Lots of small namespace volumes
    - Chosen per user/tenant/data feed
    - Mount tables for unified namespace
      - Can be managed by a central volume server
  - Move namespace from one container to another for balancing
- Combined with partial namespace
  - Choose number of namenodes to match
    - Sum of (Namespace working set)
    - Sum of (Namespace throughput)

Thank You

More information

1. HDFS-1052: HDFS Scalability with multiple namenodes
2. Hadoop – 7426: user guide for how to use viewfs with federation
3. An Introduction to HDFS Federation – https://hortonworks.com/an-introduction-to-hdfs-federation/

Other Resources

- **Next webinar: Improve Hive and HBase Integration**
  - May 2, 2012 @ 10am PST
  - Register now :http://hortonworks.com/webinars/

- **Hadoop Summit**
  - June 13-14
  - San Jose, California
  - www.Hadoopsummit.org

- **Hadoop Training and Certification**
  - Developing Solutions Using Apache Hadoop
  - Administering Apache Hadoop
  - http://hortonworks.com/training/

Backup slides

HDFS Federation Across Clusters