Hortonworks

Hadoop: Embracing future hardware

Suresh Srinivas
@suresh_m_s

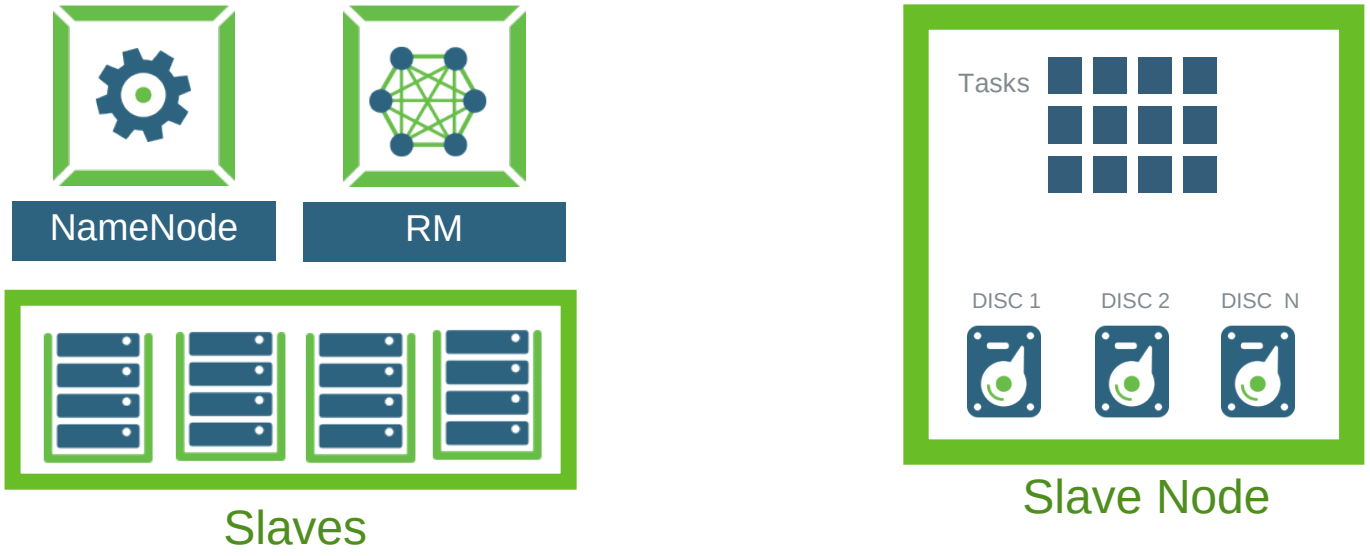
About Me

- **Architect & Founder at Hortonworks**
- **Long time Apache Hadoop committer and PMC member**
- **Designed and developed many key Hadoop features**
- **Experience in supporting many clusters**
  - Including some of the world's largest Hadoop clusters

Agenda

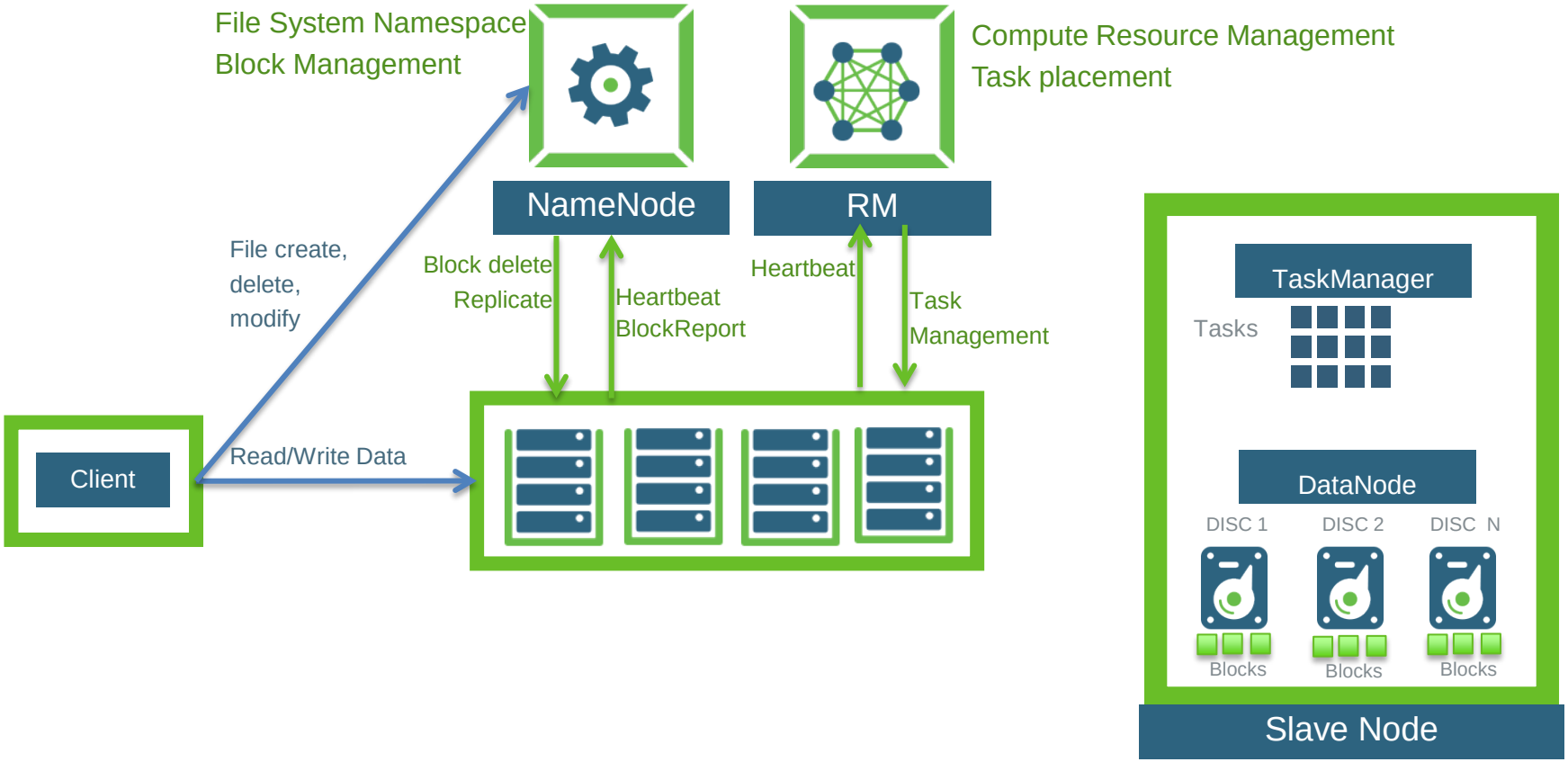
- **Hadoop Overview**
- **Changing Hardware Landscape**
- **Changing Use Cases**
- **Hierarchical Storage**
- **HDFS Cache**
- **Future**

Hadoop Quick Overview



- **Cluster of commodity servers - 3 to 1000s of nodes**
- **HDFS uses slave nodes for storage and large aggregated I/O bandwidth**
- **MapReduce uses same nodes for computation**
- ***Move computation to data***

Architecture



Commodity Hardware in 2008

- **Commodity 1U Server**
  - 2 quad core CPUs
  - 4x1TB SATA disks per node
  - 8G RAM per node
- **4000 node cluster at Yahoo**
  - 1 gigabit ethernet on each node
  - 40 nodes per rack
  - 4 gigabit ethernet uplinks from each rack to the core
  - Over 30,000 cores, 32 TB memory with nearly 16PB of raw disk!
- **10 gigabit ethernet was still expensive**

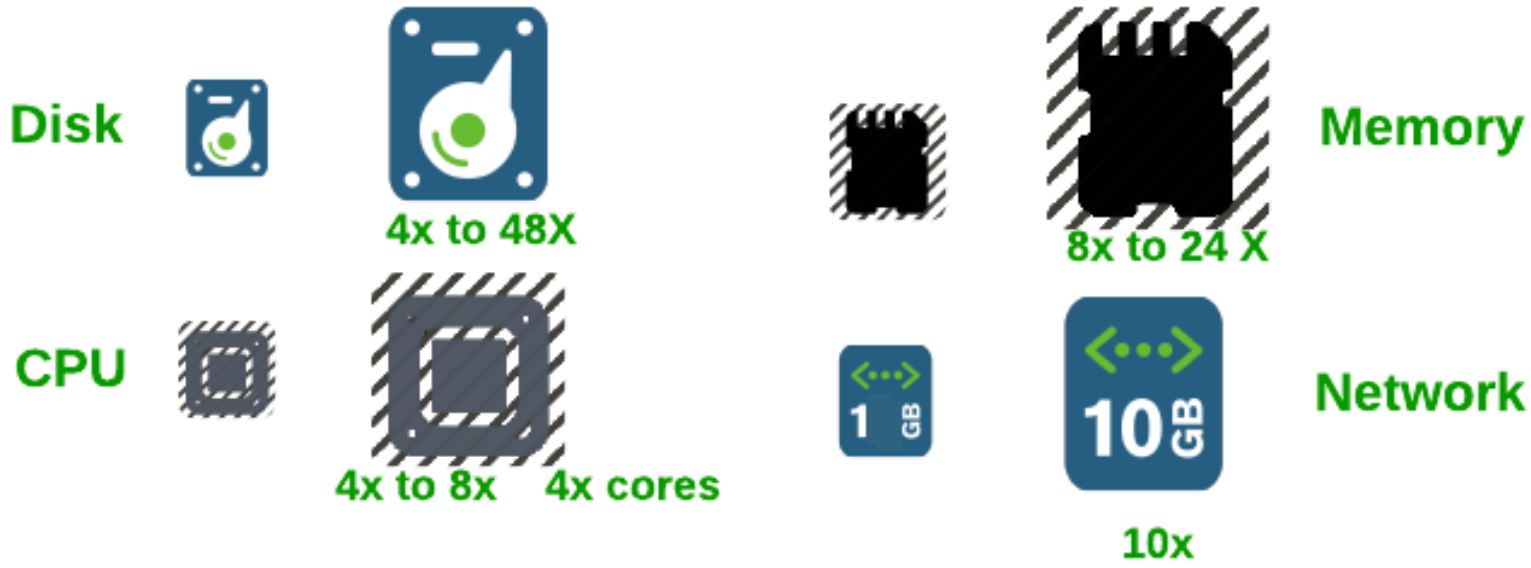
Applications in 2008

- **Predominantly batch oriented**
  - Only MapReduce for computation
  - Hive
  - Pig
- **Batch oriented access patterns**
  - Sequential reads and writes
  - Throughput over Latency
  - Low bar for durability
    - File deleted when not closed
- **HBase**
  - Random reads
  - Many features added
    - Durability using ***sync***

New Clusters

- **CPU**
  - Moore's law - CPU transistors/perf doubles every 2 years
  - More cores, Faster memory channels, I/O interfaces, Accelerators
- **Storage**
  - Increasing storage density - 3TB/4TB disks
  - But I/O throughput and latencies are the same
- **New Storage choices**
  - SSD, NVRAM
  - RAM as cache instead of just in-memory application data (100s of GBs)
- **Commodity 1U/2U/4U Server**
  - More performance: 2 CPUs– 8/12/16 cores
  - More storage: 4x4TB / 6x4TB / 12x4TB SATA disks per node
  - More memory: 72G/128G/196G RAM per node
- **Cluster network**
  - 10 gigabit intra-rack ethernet
  - Infiniband in some cases

New Clusters



- **More storage, compute, memory, network bandwidth per node**
- **2008 4000 1U node cluster = 500 – 1000 1U nodes cluster today**
  - Conversely 2013 4000 nodes = 2008 ~12000 nodes!
- **2008 software assumptions need a revisit**

Challenges

- **Software needs better scalability**
  - Better processors, more {cores, storage, memory}
- **HDFS**
  - More files and blocks
  - More blocks per DataNode
    - Bigger block reports
- **Processing**
  - Higher number of parallel tasks in a cluster
  - Higher network bandwidth
- **Current assumptions no longer valid**
  - storage is uniform
  - nodes are uniform

New Applications

- **New applications/Use cases**
  - Batch processing to Interactive query (Stinger/Tez)
    - Memory hungry
    - Latency sensitive
    - Random reads
    - Memory cache
  - New processing frameworks
    - Batch processing to Streaming
    - Graph processing
- **New deployment models**
  - Cloud/Hadoop as a service
    - Higher density nodes – 240x 2008 node size!
    - VMs instead of physical machines
  - Appliance
    - Higher density nodes
    - Infiniband

Scalabililty

- **HDFS Federation**
  - Multiple independent namespaces support
  - More files and blocks
  - More storage support

- **YARN**
  - Resource management separate from compute framework
  - Support for new types of compute framework
    - MapReduce is just one of the compute frameworks
  - More number of parallel tasks in the cluster
  - Better resource management and isolation with cgroups

Heterogeneous Storage

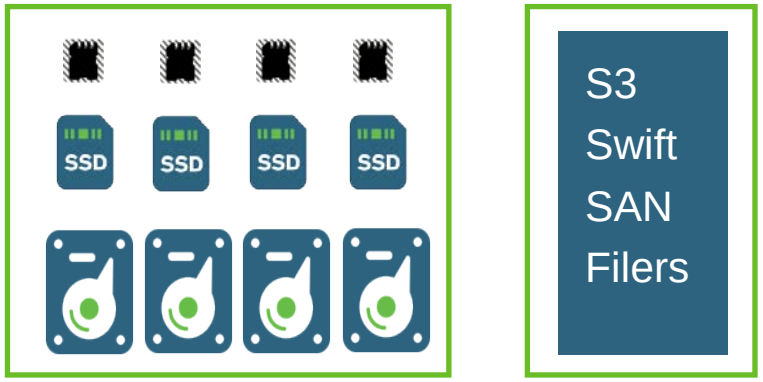
Current Architecture

- DataNode is a **single storage**
- Storage is uniform - Only storage type Disk
- Storage types hidden from the file system

All disks as a single storage

New Architecture

- DataNode is a **collection of storages**
- Support different types of storages
  - Disk, SSDs, Memory
- Support external storages
  - S3, OpenStack Swift



Collection of tiered storages

Heterogeneous Storage

- **Support new hardware/storage types**
  - Datanodes configured with per volume storage types
  - Block report per storage – better scalability
- **Applications choose storage tier appropriate for the work load**
  - SSD for random reads
    - HBase and other query loads
  - Memory only data
    - Intermediate data
- **Mixed storage type support**
  - Some copies in SSD and some on disk
  - Some copies on disk in HDFS and some on S3/OpenStack Swift
- **Storage type preference per file**
  - Tiered storages
- **Multi-tenant support – quotas per storage type**
- **Work in progress – HDFS-2832**

HDFS Cache

- **Co-ordinated caching of data in DataNodes**
  - Cache dimension tables for Hive
  - Cache data required for multiple queries
- **Mechanism**
  - Off heap memory
  - Fast zero copy access to data using DirectByteBuffer
  - Caching by pinning data in memory – mmap, mlock, munlock
  - Entire blocks are cached
- **Administrator/user issues commands to cache the data**
  - Cache done using paths to files or directories
  - Directory based cache adds newly created files to cache
  - After cache use admin issues command to uncache the data
- **Quota using hierarchical pools**
  - Pool to have min/max bytes
- **Work in progress – HDFS-4949**

Future

- **Dynamic caching**
  - Access pattern based caching of hot data
    - LRU, LRU2
  - Cache partial blocks
  - Dynamic migration of data between storage tiers
- **Multiple network interface support**
  - Better aggregated bandwidth utilization
  - Isolation of traffic
- **Support NVRAM**
  - Better durability without write performance cost
  - File system metadata to NVRAM for better throughput
- **Hardware Security Modules**
  - Better key management
  - Processing that requires higher security only on these nodes
  - Important requirement for Financials and Healthcare

Q & A

Thank You!