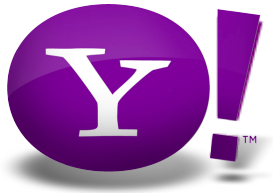
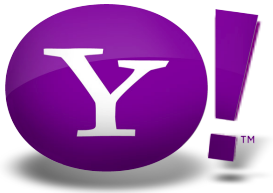
Hadoop – A platform for Grid Computing

Devaraj Das
Grid Computing, Yahoo! Bangalore



What is this talk about

- The challenge of large scale data processing
- Grid computing as a solution
- Components of a Grid
- Introduction to Hadoop
- Hadoop ecosystem



The Challenge:
Very large scale data processing

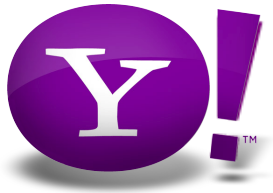
Internet Scale Generates *BigData*

- Yahoo is the most Visited Site on the Internet
  - 600M+ Unique Visitors per Month
  - Billions of Page Views per Day
  - Billions of Searches per Month
  - Billions of Emails per Month
  - IMs, Address Books, Buddy Lists, …
  - **Terabytes of Data per Day!**
- And we crawl the Web
  - 100+ Billion Pages
  - 5+ Trillion Links
  - **Petabytes of data**

…terabytes, 100s of terabytes, petabytes, 10s of petabytes…

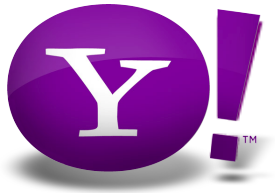
A huge amount of work is needed to process that data…

- Search queries
  - Searching on billions of docs
- Ads and Display
  - Figure out what ads to show, based on user preferences, demographic and geographic data
  - Customize content
  - Model ad performance and query patterns
  - Trace user activity to discover short- and long-term behavioral trends
  - Personalize pages via models (e.g., MyYahoo)



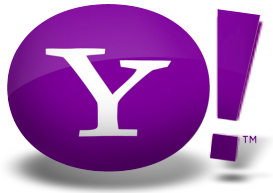
How to Process BigData?

- Remember, we're dealing with 100s of terabytes.
- Just reading 100 terabytes of data can be overwhelming
  - On a standard computer (100 MBPS):
    - ~11 days
  - Across a 10Gbit link (very high end storage solution):
    - 1 day
  - On 1000 standard computers:
    - 15 minutes!



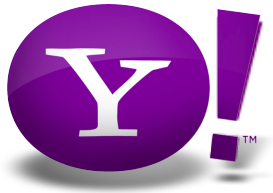
A little detour – getting the most out of computers

- Parallel computing:
  - vector processing (SIMD)
  - multiple cores/processors
  - thread-based parallelism
  - Problems: physical limits, heat, power
- Massively parallel processing
  - Lots of networked processors
  - bandwidth becomes a problem
  - HW is expensive
- Distributed/grid/cloud/utility:
  - Lots of networked machines
  - Usually, commodity machines



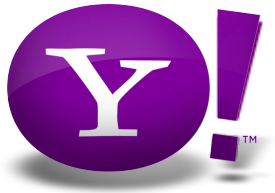
Yahoo!’s approach: Grid computing

- 1000s of commodity machines
  - no special hardware, storage
- Software layer (Hadoop); this is the hard part
- Similar model deployed by many others: Google, Amazon, IBM.



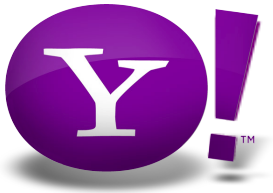
What is hard in Grid Computing

- Reliability problems: in large clusters, computers fail every day, and in different ways
  - For a single machine: MTBF is ~3 years
  - On 1000 machines, MTBF is ~1 day
  - Data is corrupted or lost (disk, memory, network)
  - Computations are disrupted
- Without a good framework, programming clusters is **very hard**
  - newer programming paradigms? languages?
  - Traditional debugging and performance tools don't apply



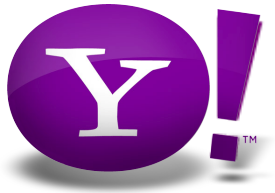
What is hard in Grid Computing

- Storage
  - how do I store petabytes of data, reliably?
  - How is data available to all machines?
- Resource management / scheduling
  - How are multiple jobs run?
  - How do I make sure all resources are utilized
- Others (communication, security, monitoring, etc) on a much larger scale



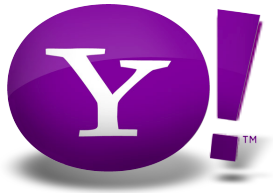
Recap

- Our challenge is large scale data processing (petabytes of data)
- We use grids/clusters of thousands of networked commodity machines
- Hadoop is the software that ties this together (more later)
- Lots of hard problems, made harder by scale and heterogeneous machines



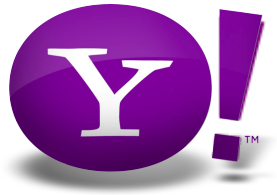
Agenda

- The challenge of large scale data processing
- Grid computing as a solution
- Components of a Grid
- Introduction to Hadoop
- Hadoop ecosystem

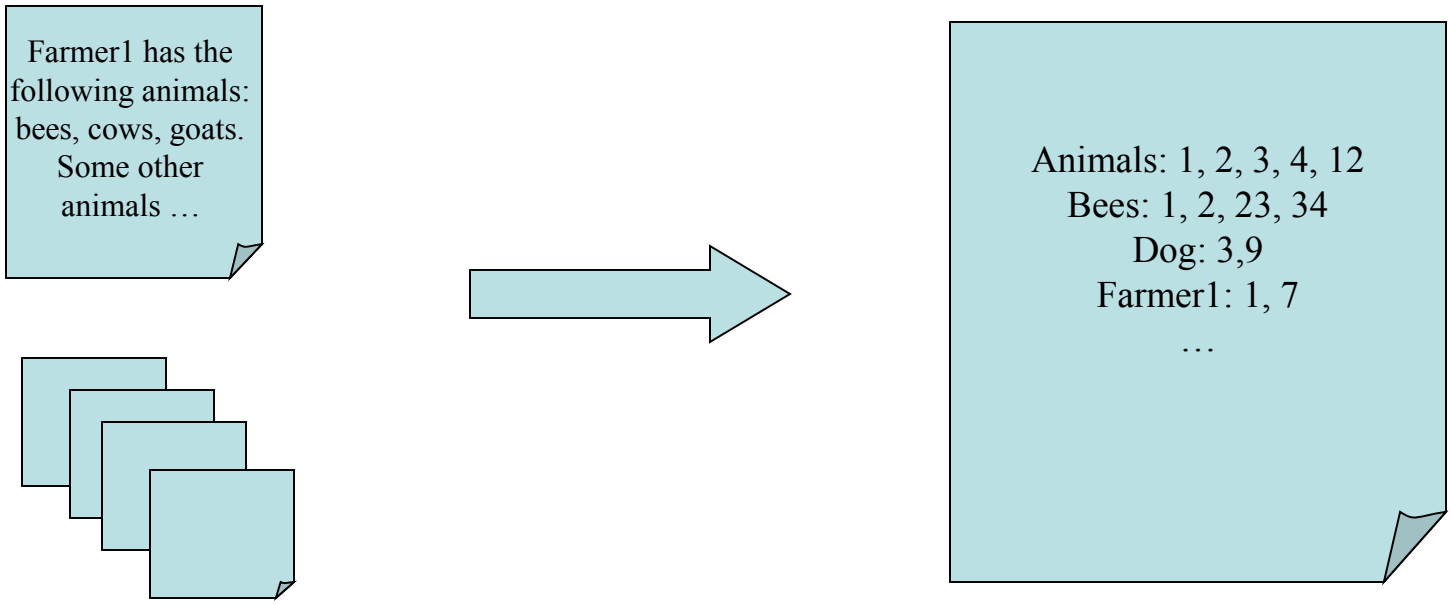


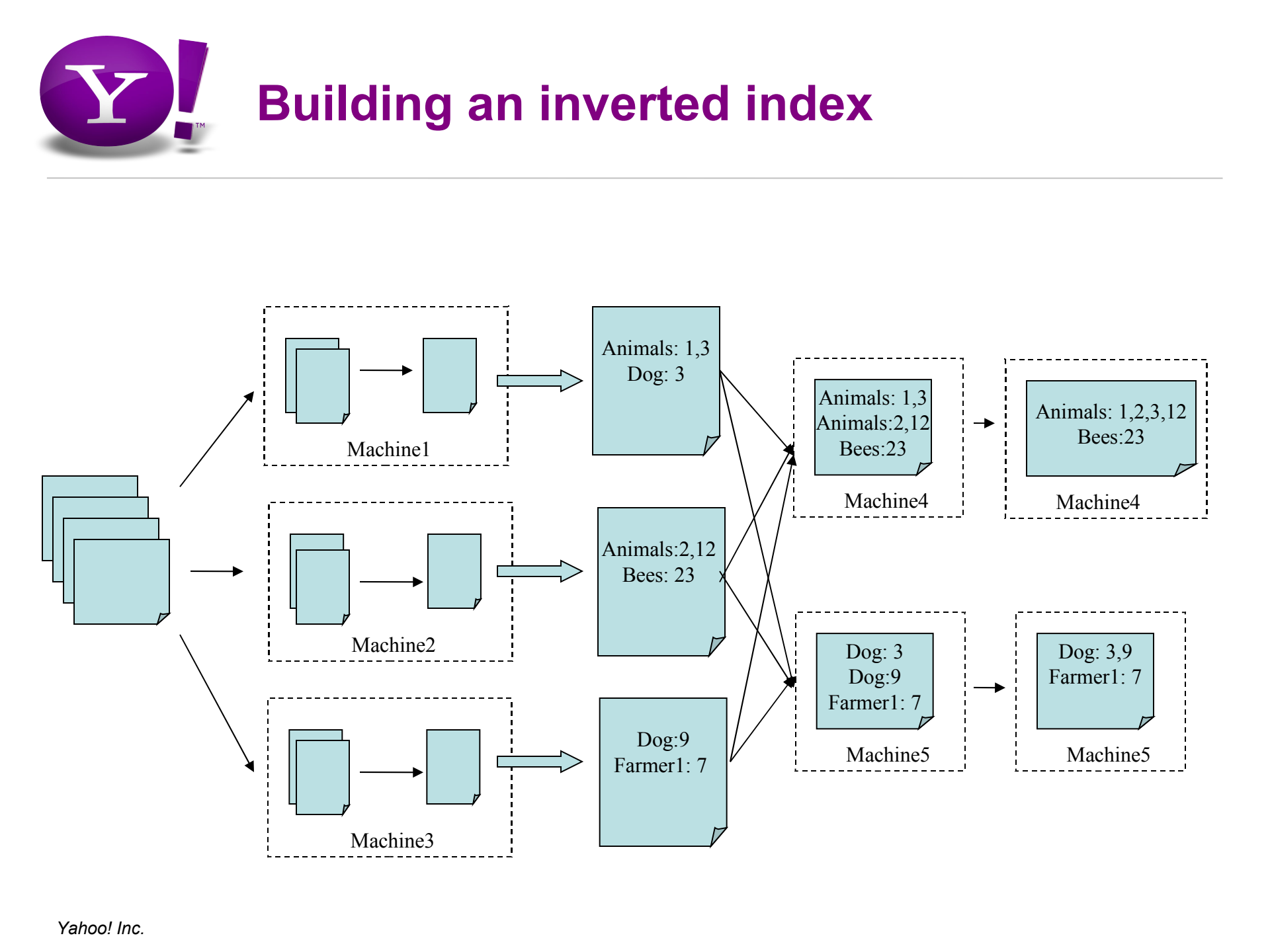
Programming on the Grid

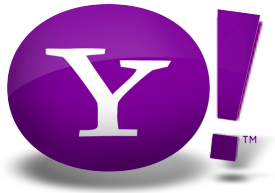
- So we have 100s of machines available to us. How do we write applications on them?
- As an example, consider the problem of creating an index for search.
  - I have hundreds of documents
  - I want to build an index so I can search on them
  - I really want to build an inverted index (a map of words to their location)
  - I have a few machines at my disposal



The problem: inverted index



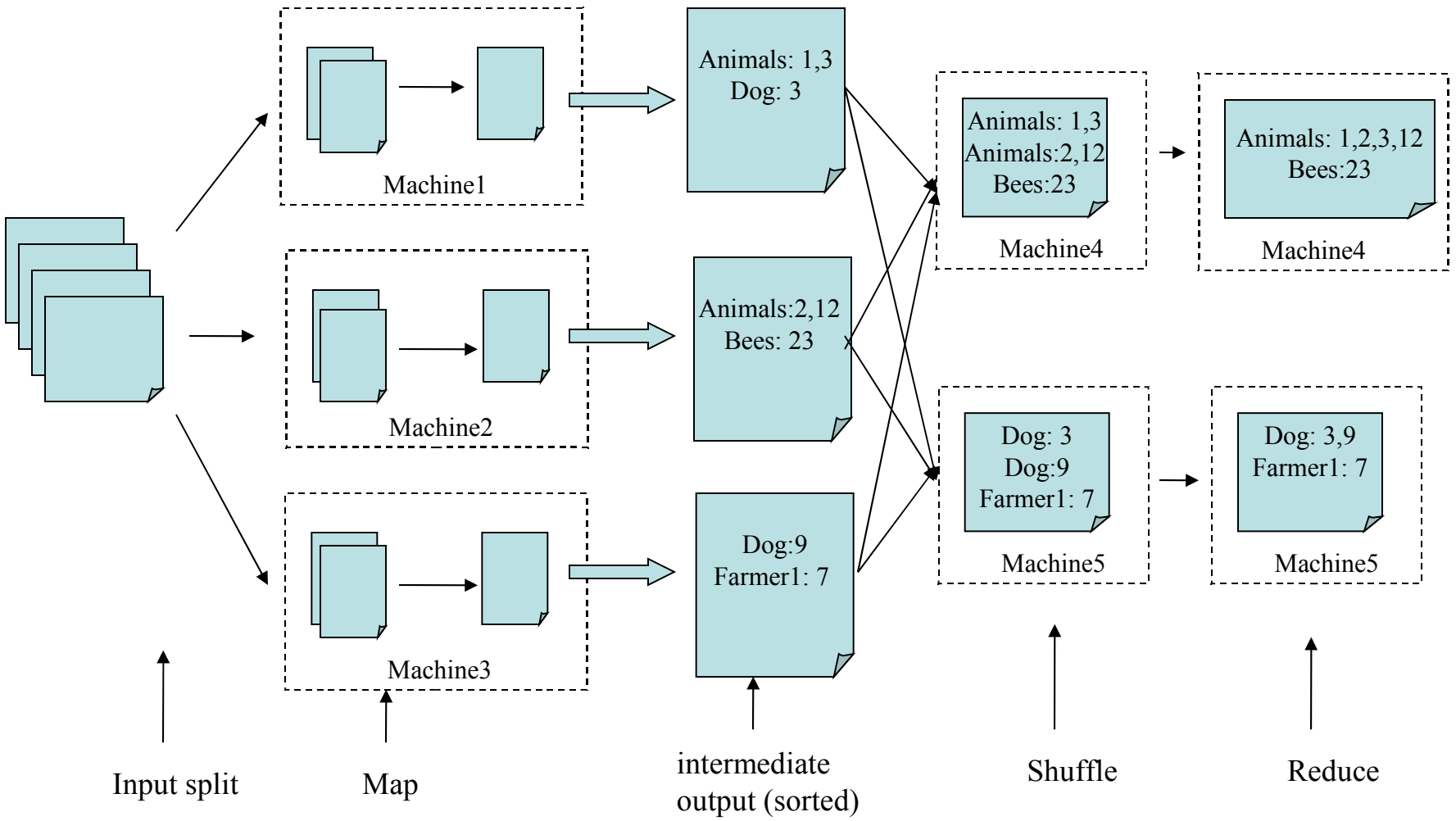

Building an inverted index
Machine1
Machine2
Machine3
Animals: 1,3
Dog: 3
Animals:2,12
Bees: 23
Dog:9
Farmer1: 7
Animals: 1,3
Animals:2,12
Bees:23
Machine4
Animals: 1,2,3,12
Bees:23
Machine4
Dog: 3
Dog:9
Farmer1: 7
Machine5
Dog: 3,9
Farmer1: 7
Machine5

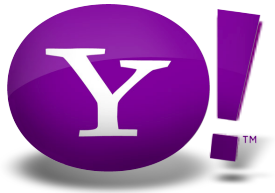


This is Map-Reduce

- General form:
  - Map: (K1, V1) ➞ list(K2, V2)
  - Reduce: (K2, list(V2)) ➞ list(K3, V3)
- In our example
  - Map: (doc#, word) ➞ [(word, doc-num)]
  - Reduce: (word, [doc1, doc3, ...]) ➞ [(word, “doc1, doc3, …”)]

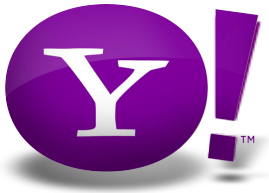
Mapping our example to Map-Reduce





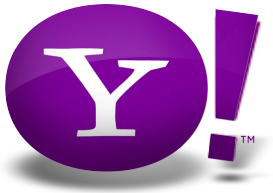
Map-Reduce on a larger scale

- Take the previous example and expand it
  - billions of web pages
  - index can reach a few petabytes
  - Thousands of machines
  - Run multiple jobs/programs
- We need a platform that can let us do this. What should the platform support?



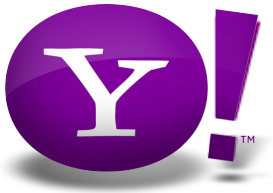
Grid components

- Storage (file system)
  - highly available (machines can fail)
  - replicated (large amounts of data, can't store copies on every machine)
  - data consistency (due to replication)
- Framework (Map-Reduce)
  - user should just say where the data is, how to split it, and what to do with each data set
  - Framework should start/stop tasks, move data/computation, manage execution



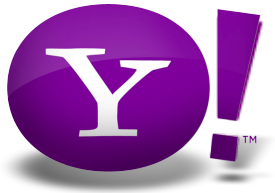
Grid components

- Communication / data exchange
  - Serialize/deserialize data (into files, across networks) in a generic manner
  - Language/OS neutral.
  - What transport mechanism? Sockets?
- Provisioning
  - To utilize machines, we need to run many applications. How do we share resources across many apps?
- Monitoring
  - error detection, logging, etc



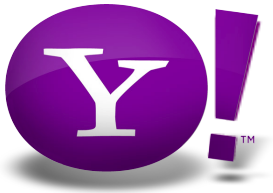
Recap

- We saw an example of Map-Reduce
- What do we need in a Grid:
  - Storage
  - Programming model (Map-Reduce)
  - Communication
  - Provisioning
  - Monitoring, error handling



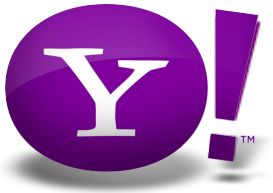
Agenda

- The challenge of large scale data processing
- Grid computing as a solution
- Components of a Grid
- Introduction to Hadoop
- Hadoop ecosystem



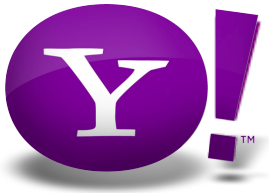
Hadoop

- Framework for running applications on large clusters built of commodity hardware
- Lets one easily write and run applications that process vast amounts of data (petabytes).
- Distributed File System
  - Modeled on GFS
- Distributed Processing Framework
  - Using Map-Reduce metaphor
- Scheduler/Resource Management



Hadoop is

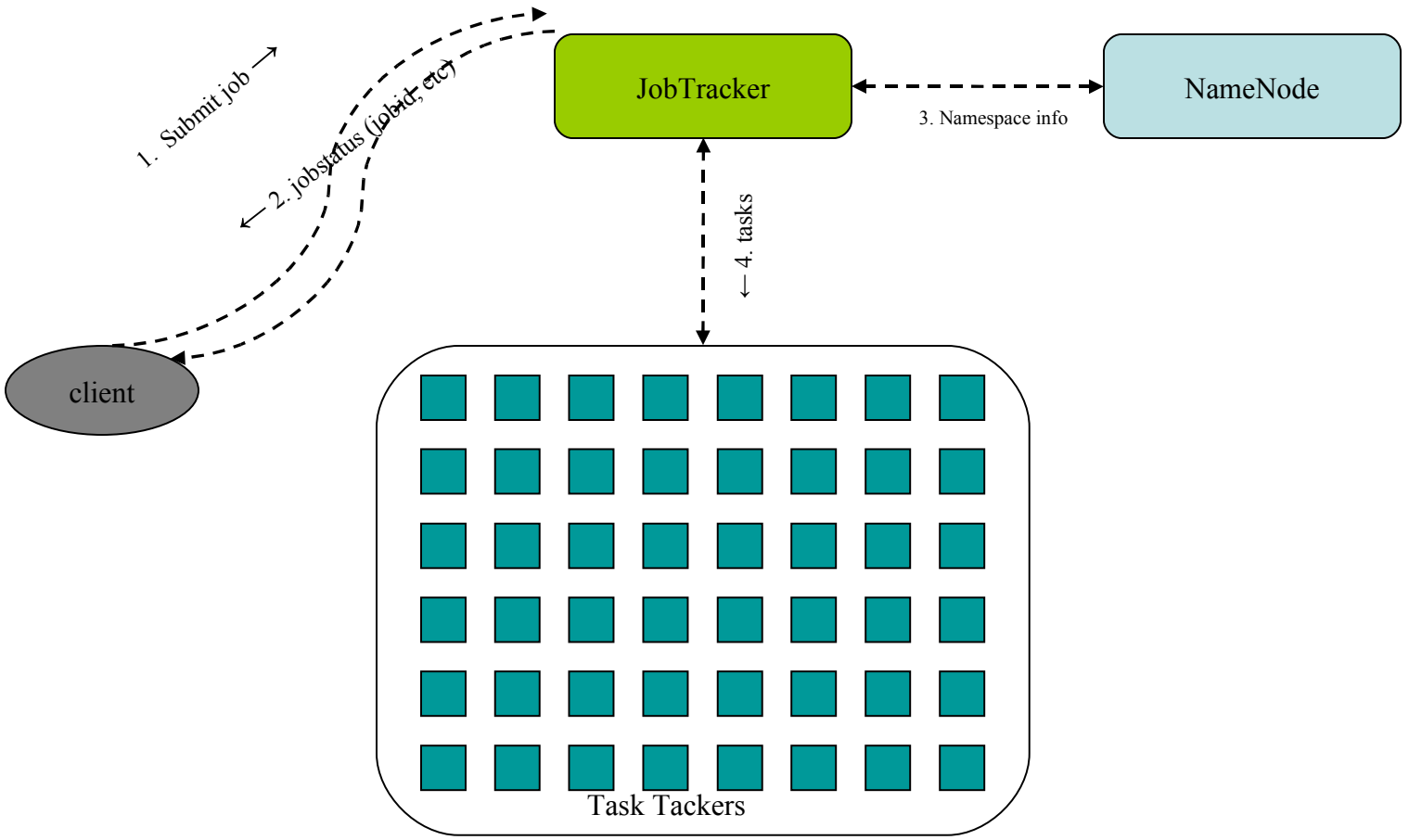
- Completely Open Source
  - Top level Apache project
- Written in Java
  - Runs on Linux, Mac OS/X, Windows, and Solaris
  - Commodity hardware
  - Client apps can be written in various languages

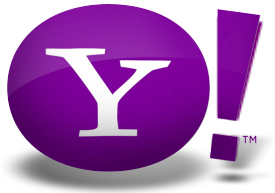


Apache Hadoop – Open Source

- Multi-Organization Development Community
  - Yahoo! is primary contributor so far
    - Bangalore team drives Map-Reduce and Scheduling work
  - Also IBM, Amazon, Facebook, Powerset, …
  - And various independent programmers
- Anyone can use Hadoop, for free!
  - 33+ organization have registered their Hadoop usage / clusters
  - Hadoop is used in Universities on several continents
  - We've started hiring employees with Hadoop experience!
- Anyone can enhance Hadoop!
  - Fix a bug, submit a test case, write some docs, starting is easy
  - Publish a new Hadoop application or two
  - Hadoop's direction is determined by those who invest in it

Hadoop Job


1. Submit job
2. jobstatus (jobid, etc)
JobTracker
NameNode
3. Namespace info
4. tasks
client
Task Tackers

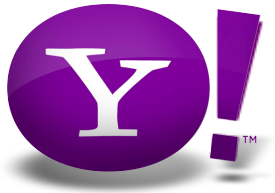


Agenda

- The challenge of large scale data processing
- Grid computing as a solution
- Components of a Grid
- Introduction to Hadoop
- Hadoop ecosystem

Hadoop Software Ecosystem

- Pig – Yahoo!
  - Parallel Programming Language and Runtime
- Zookeeper – Yahoo!
  - High-Availability Directory and Configuration Service
- HBase – Powerset.com
  - TableStore layered on HDFS
- JACL – IBM
  - JSON / SQL inspired programming Hadoop language
- Mahout – Individual apache members
  - Machine learning libraries for Hadoop
- Tashi – Intel, CMU
  - Virtual machine provisioning service (soon)
- Hive – Facebook.com
  - Data warehousing framework on Hadoop (soon)



Yahoo! Grid Services

- We operate multiple grid clusters within Yahoo!
- 10,000s nodes, 100,000s cores, TBs RAM, PBs disk
- Support large internal user community
- Manage data needs (Ingest TBs per day)
- Deploy and manage software (Hadoop, Pig, etc)

Example: Yahoo! WebMap

- Process which builds a database of all known Web pages and sites on the internet and a vast array of data about every page and site
- Includes a directed graph of the web
- The WebMap data feeds the Machine Learned Ranking algorithms at the heart of Yahoo! Search
- Build process consists of ~100 applications

- Ported to Hadoop MapReduce, entered production Q1 2008
- 33% time savings over previous version (non-MapReduce) on the same hardware
- Dramatically more maintainable and extensible
- Over 10,000 processors used in build
- Over 5PB of raw disk in cluster
- Output size >300TB (compressed)
- Largest job runs for ~70 hours

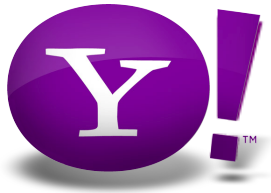
The largest production Hadoop job we know of!

Search & Advertising Sciences

An example application: Search Assist™

- Database for **Search Assist™** is built using Hadoop.
- 3 years of log-data
- 20-steps of map-reduce

	Before Hadoop	After Hadoop
Time	26 days	***20 minutes***
Language	C++	Python
Development Time	2-3 weeks	2-3 days

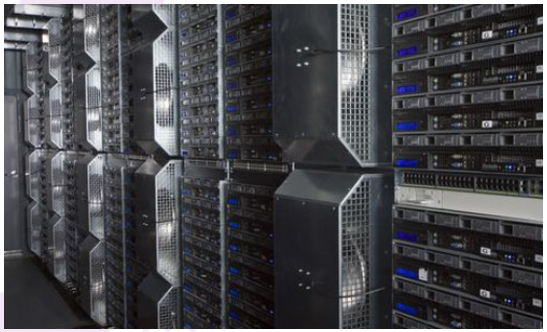


Hadoop usage outside Yahoo

- Amazon
  - Build product search index, process millions of sessions daily
  - Clusters vary from 1-100 nodes
- Facebook
  - log file processing for analytics/reporting, machine learning
  - 320 machine cluster, 2560 cores, 1.3PB of raw storage
- Lots of users of Nutch
- Intel/CMU
  - Building Ground Models of Southern California
- The New York Times uses it to process their archives

M45 : Open Academic Clusters

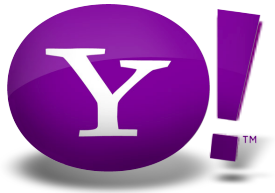
- Collaboration with Major Research Universities
  - Foster open research
  - Focus on large-scale, highly parallel computing
- Seed Facility: Datacenter in a Box (DiB)
  - 500 nodes, 4000 cores, 3TB RAM, 1.5PB disk
  - High bandwidth connection to Internet
  - Located on Yahoo! corporate campus
- Runs Yahoo! / Apache Grid Stack
- Carnegie Mellon University is Initial Partner
- Public Announcement 11/12/07



What's ahead

- Improved scalability
  - E.g. 10s K nodes
  - Federated applications across clusters and data centers
- Improved performance
- Enhanced features
- Further extensions
  - on-line service grid?
- More applications, from many fields!

The Hadoop eco-system is growing and has the potential to have a lot of impact across the internet industry, and many others!



Reference links

- http://hadoop.apache.org/ - The main Apache site
  - Mailing lists, the code, documentation and more
- http://developer.yahoo.com/blogs/hadoop - Our blog
  - Reports, videos,… More on the way
- http://wiki.apache.org/hadoop/PoweredBy
  - A list of users, please add yourself!
- http://wiki.apache.org/hadoop/ProjectSuggestions
  - Ideas for folks who want to get started
- http://developer.yahoo.net/blog/archives/2007/07/yahoo-h
  - Our first timeline post…