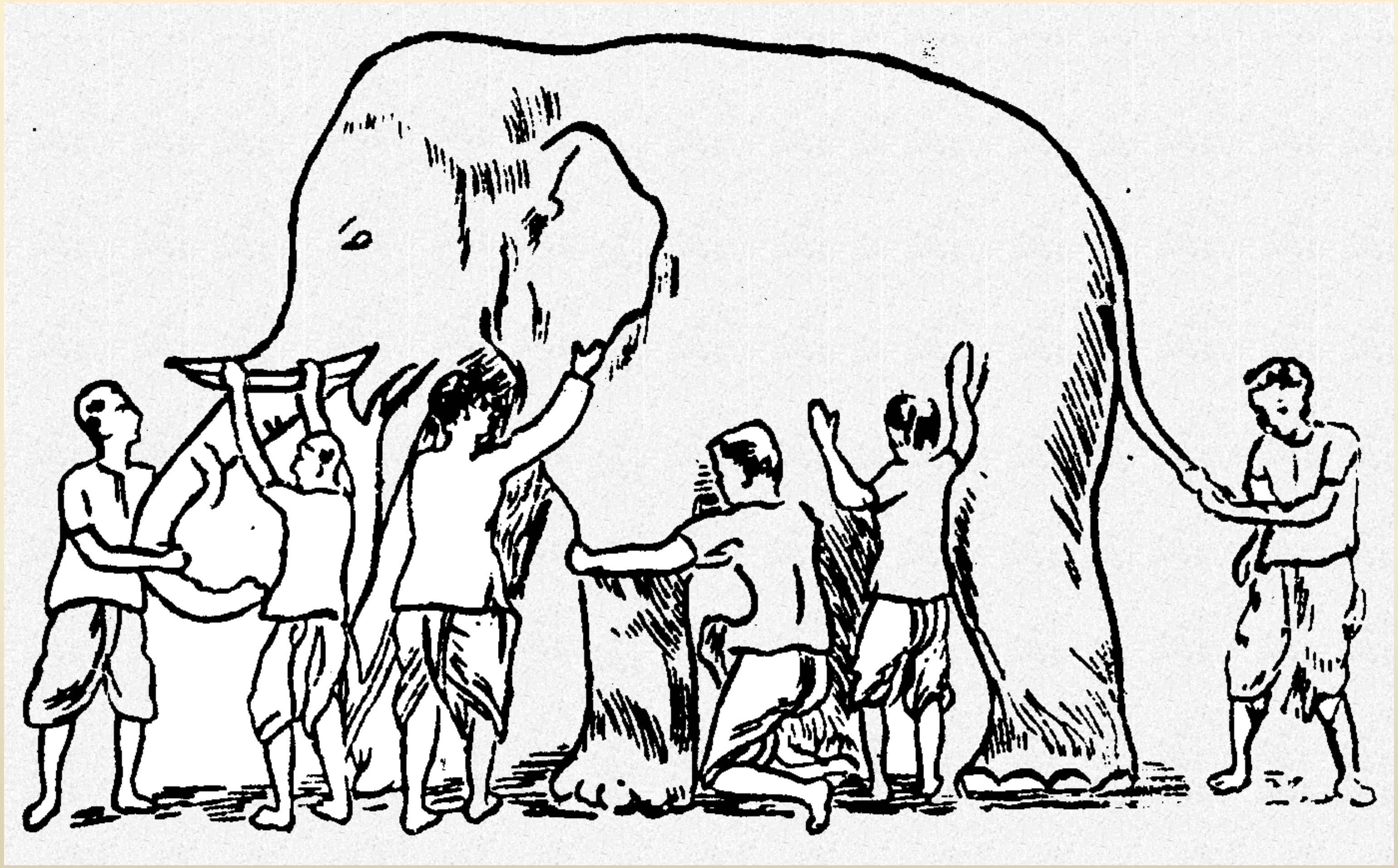
Data Intensive Computing

B. Ramamurthy

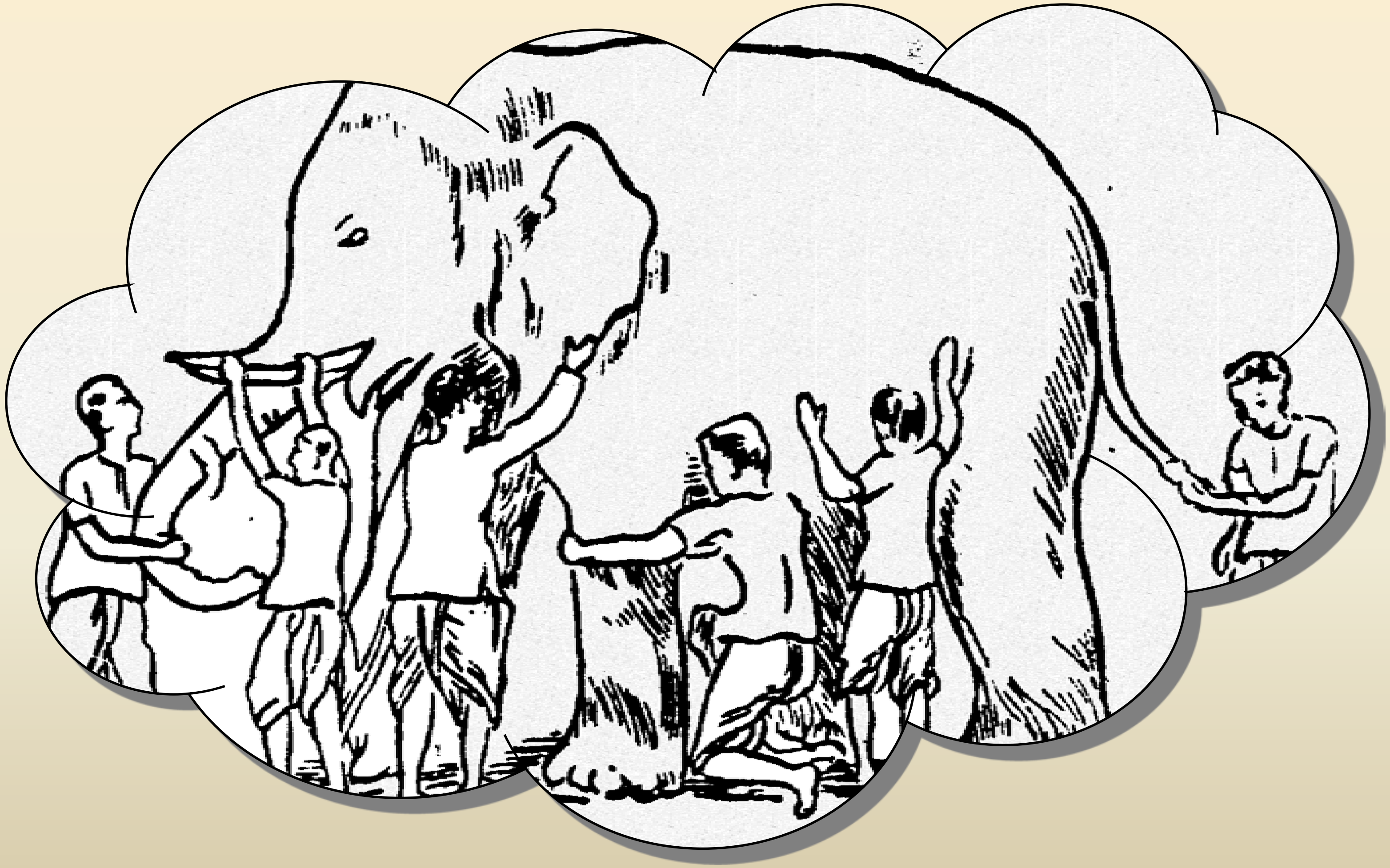
This work is Partially Supported by
NSF DUE Grant#: 0737243, 0920335

bina@buffalo.edu

Indian Parable: Elephant and the Blind men



Cloud Computing

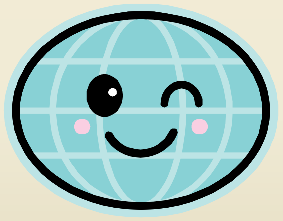


Goals of this talk

- Why is data-intensive computing relevant to cloud computing?
- Why is MapReduce programming model important for data-intensive computing?
- What is MapReduce?
- How is its support structure different from traditional structures?

Relevance to WIC

- Data-intensiveness is the main driving force behind the growth of the cloud concept
- Cloud computing is necessary to address the scale and other issues of data-intensive computing
- Cloud is turning computing into an everyday gadget
- Women are indeed experts at managing and effectively using gadgets!!??
- They can play an critical role in transforming computing at this momentous time in computing history.



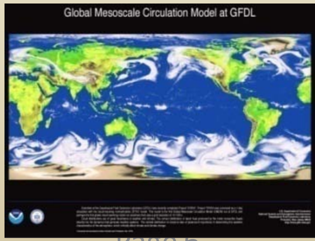
Definition

- Computational models that focus on data: large scale and/or complex data
- Example1: web log

```
fcrawler.looksmart.com - - [26/Apr/2000:00:00:12 -0400] "GET /contacts.html HTTP/1.0" 200 4595 "-" "FAST-WebCrawler/2.1-pre2 (ashen@looksmart.net)"
fcrawler.looksmart.com - - [26/Apr/2000:00:17:19 -0400] "GET /news/news.html HTTP/1.0" 200 16716 "-" "FAST-WebCrawler/2.1-pre2 (ashen@looksmart.net)"
ppp931.on.bellglobal.com - - [26/Apr/2000:00:16:12 -0400] "GET /download/windows/asctab31.zip HTTP/1.0" 200 1540096
        "http://www.htmlgoodies.com/downloads/freeware/webdevelopment/15.html" "Mozilla/4.7 [en]C-SYMPA  (Win95; U)"

123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET /pics/wpaper.gif HTTP/1.0" 200 6248 "http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)"
123.123.123.123 - - [26/Apr/2000:00:23:47 -0400] "GET /asctortf/ HTTP/1.0" 200 8130 "http://search.netscape.com/Computers/Data_Formats/Document/Text/RTF" "Mozilla/4.05
        (Macintosh; I; PPC)"
123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET /pics/5star2000.gif HTTP/1.0" 200 4005 "http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)"
123.123.123.123 - - [26/Apr/2000:00:23:50 -0400] "GET /pics/5star.gif HTTP/1.0" 200 1031 "http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)"
123.123.123.123 - - [26/Apr/2000:00:23:51 -0400] "GET /pics/a2hlogo.jpg HTTP/1.0" 200 4282 "http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)"
123.123.123.123 - - [26/Apr/2000:00:23:51 -0400] "GET /cgi-bin/newcount?jafsof3&width=4&font=digital&noshow HTTP/1.0" 200 36 "http://www.jafsoft.com/asctortf/" "Mozilla/4.05
        (Macintosh; I; PPC)"
```

- Example 2: Climate/weather data modeling

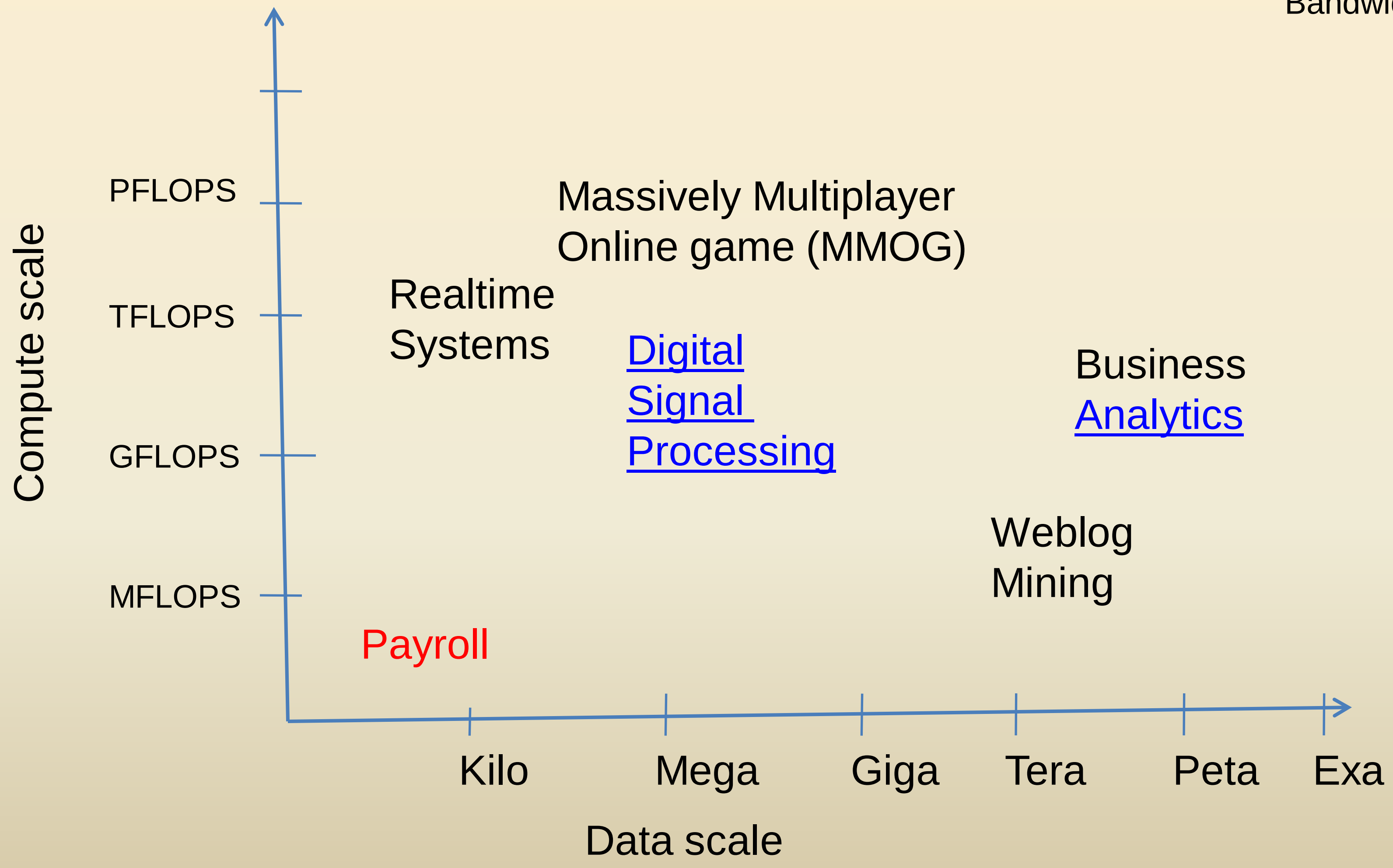


Background

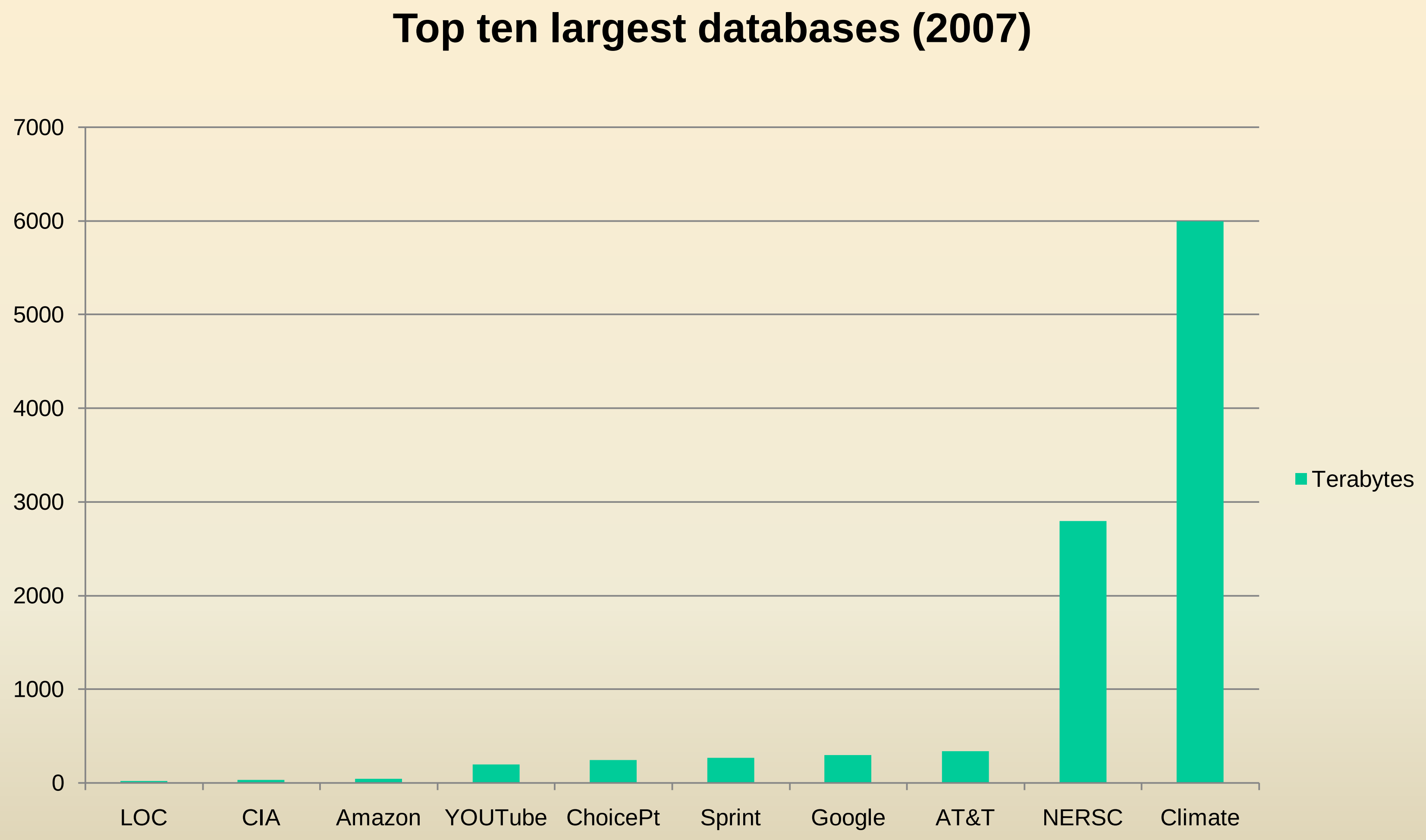
- Problem Space: explosion of data
- Solution space: emergence of multi-core, virtualization, cloud computing
- Inability of traditional file system to handle data deluge
- The Big-data Computing Model
  - MapReduce Programming Model (Algorithm)
  - Google File System; Hadoop Distributed File System (Data Structure)
  - Microsoft Dryad
- Cloud Computing and its Relevance to Big-data and Data-intensive computing –Plenary on 6/24

Problem Space

Other variables:
Communication
Bandwidth, ?



Top Ten Largest Databases



Ref: http://www.businessintelligencelowdown.com/2007/02/top_10_largest_.html

Processing Granularity

Data size: small

Pipelined Instruction level

Concurrent Thread level

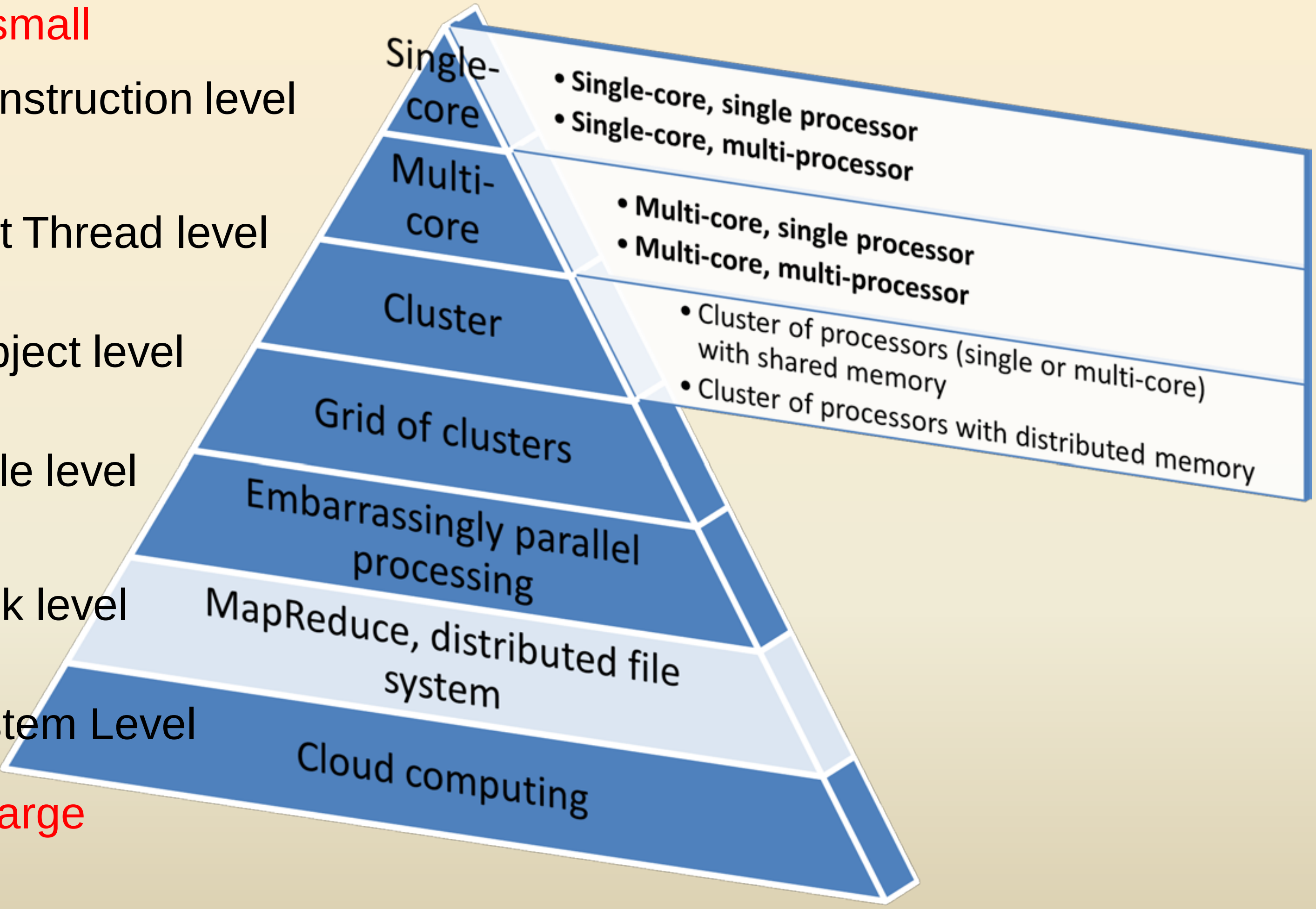
Service Object level

Indexed File level

Mega Block level

Virtual System Level

Data size: large


Single-core
Multi-core
Cluster
Grid of clusters
Embarrassingly parallel processing
MapReduce, distributed file system
Cloud computing
• Single-core, single processor
• Single-core, multi-processor
• Multi-core, single processor
• Multi-core, multi-processor
• Cluster of processors (single or multi-core) with shared memory
• Cluster of processors with distributed memory

Traditional Storage Solutions

- Off system/online storage/ secondary memory
- File system abstraction/ Databases
- Offline/ tertiary memory/ DFS
- RAID: Redundant Array of Inexpensive Disks
- NAS: Network Accessible Storage
- SAN: Storage area networks

Solution Space

Google File

- Internet introduced a new challenge in the form web logs, web crawler's data: large scale "peta scale"
- But observe that this type of data has an uniquely different characteristic than your transactional or the "customer order" data : "write once read many (WORM)" ;
  - Privacy protected healthcare and patient information;
  - Historical financial data;
  - Other historical data
- Google exploited this characteristics in its Google file system (GFS)

Data Characteristics

- Streaming data access
- Applications need streaming access to data
- Batch processing rather than interactive user access.
- Large data sets and files: gigabytes, terabytes, petabytes, exabytes size
- High aggregate data bandwidth
- Scale to hundreds of nodes in a cluster
- Tens of millions of files in a single instance
- Write-once-read-many: a file once created, written and closed need not be changed – this assumption simplifies coherency
- WORM inspired a new programming model called the MapReduce programming model
- Multiple-readers can work on the read-only data concurrently

The Big-data Computing System

The Context: Big-data

- Man on the moon with 32KB (1969); my laptop had 2GB RAM (2009)
- Google collects 270PB data in a month (2007), 20000PB a day (2008)
- 2010 census data is expected to be a huge gold mine of information
- Data mining huge amounts of data collected in a wide range of domains from astronomy to healthcare has become essential for planning and performance.
- We are in a knowledge economy.
  - Data is an important asset to any organization
  - Discovery of knowledge; Enabling discovery; annotation of data
  - Complex computational models
  - No single environment is good enough: need elastic, on-demand capacities
- We are looking at newer
  - programming models, and
  - Supporting algorithms and data structures.

The Outline

- Introduction to MapReduce
- Hadoop Distributed File System
- Demo of MapReduce on Virtualized hardware
- Demo (Internet access needed)
- Our experience with the framework
- Relevance to Women-in-Computing
- Summary
- References

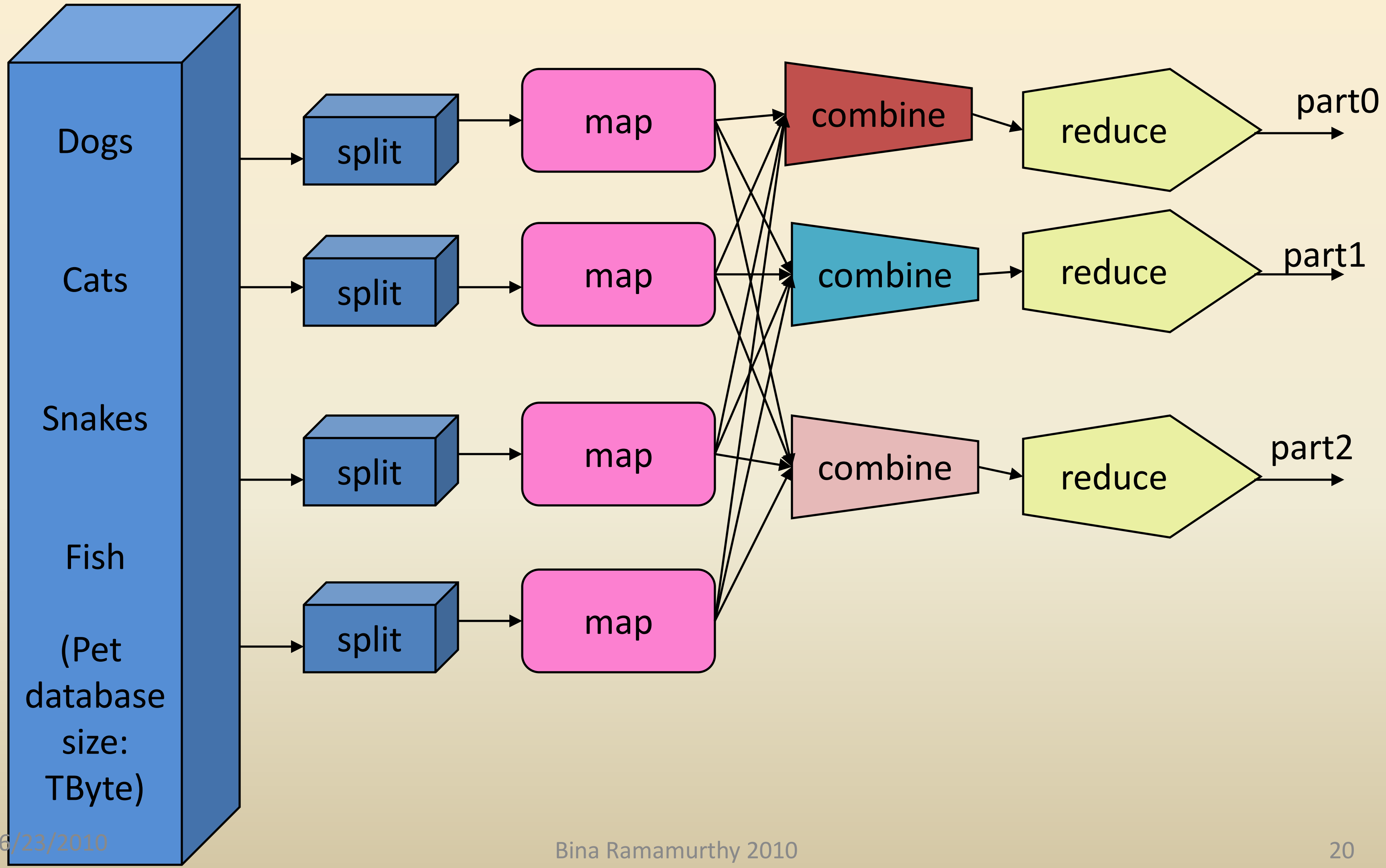
MAPREDUCE

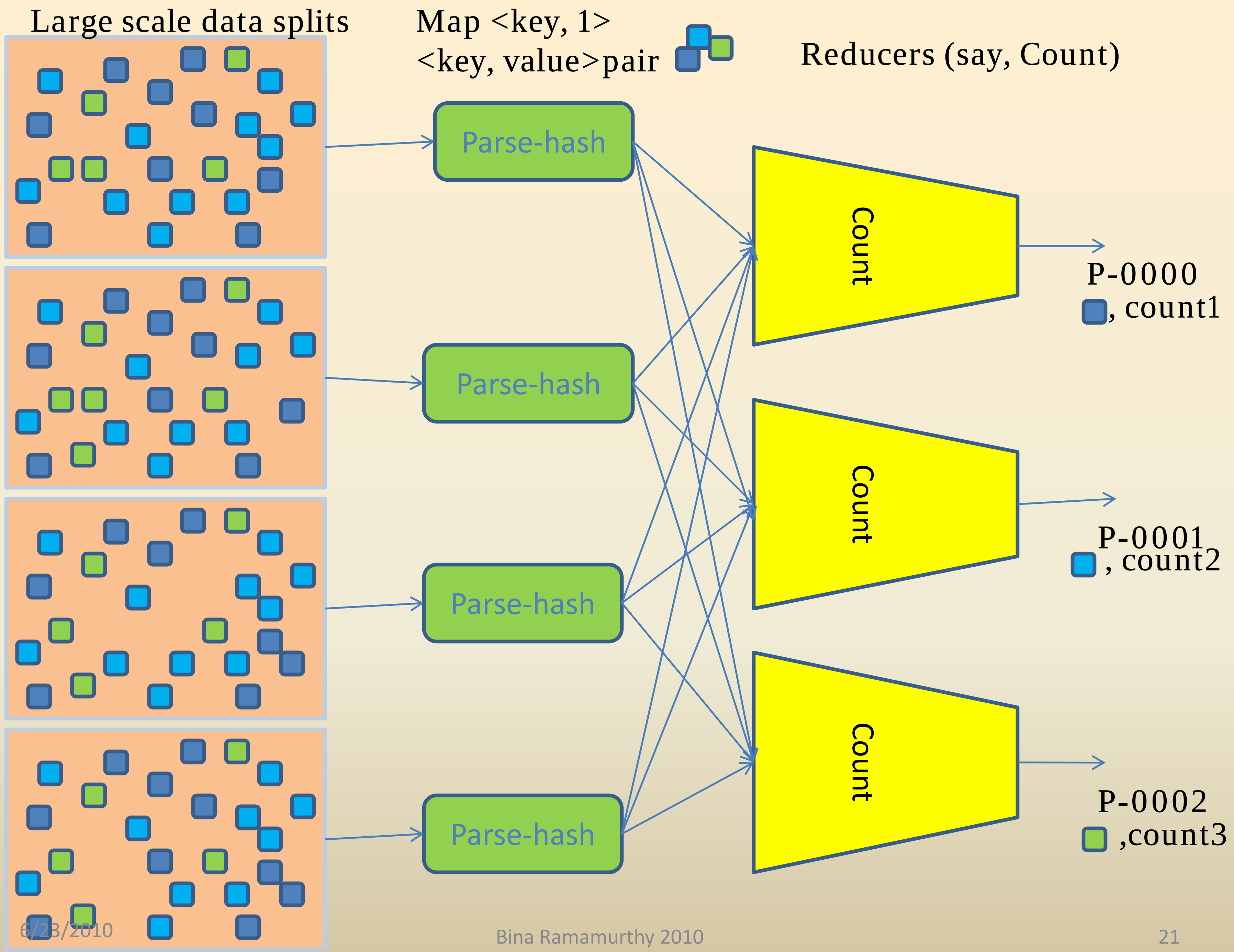
What is MapReduce?

- MapReduce is a programming model Google has used successfully is processing its “big-data” sets (~ 20000 peta bytes per day)
  - A map function extracts some intelligence from raw data.
  - A reduce function aggregates according to some guides the data output by the map.
  - Users specify the computation in terms of a *map* and a *reduce* function,
  - Underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, and
  - Underlying system also handles machine failures, efficient communications, and performance issues.

-- Reference: Dean, J. and Ghemawat, S. 2008. **MapReduce: simplified data processing on large clusters**. *Communication of ACM* 51, 1 (Jan. 2008), 107-113.

MapReduce Example in my Operating System Class




Large scale data splits
Map <key, 1>
<key, value>pair
Reducers (say, Count)
Parse-hash
Parse-hash
Parse-hash
Parse-hash
Count
Count
Count
P-0000
, count1
P-0001
, count2
P-0002
,count3

Classes of problems "mapreducable"

- Benchmark for comparing: Jim Gray's challenge on data-intensive computing. Ex: "Sort"
- Google uses it for wordcount, adwords, pagerank, indexing data.
- Simple algorithms such as grep, text-indexing, reverse indexing
- Bayesian classification: data mining domain
- Facebook uses it for various operations: demographics
- Financial services use it for analytics
- Astronomy: Gaussian analysis for locating extra-terrestrial objects.
- Expected to play a critical role in semantic web and web3.0

HADOOP

What is Hadoop?

- At Google MapReduce operation are run on a special file system called Google File System (GFS) that is highly optimized for this purpose.
- GFS is not open source.
- Doug Cutting and Yahoo! reverse engineered the GFS and called it Hadoop Distributed File System (HDFS).
- The software framework that supports HDFS, MapReduce and other related entities is called the project Hadoop or simply Hadoop.
- This is open source and distributed by Apache.

Basic Features: HDFS

- Highly fault-tolerant
- High throughput
- Suitable for applications with large data sets
- Streaming access to file system data
- Can be built out of commodity hardware
- HDFS provides Java API for applications to use.
- A HTTP browser can be used to browse the files of a HDFS instance.

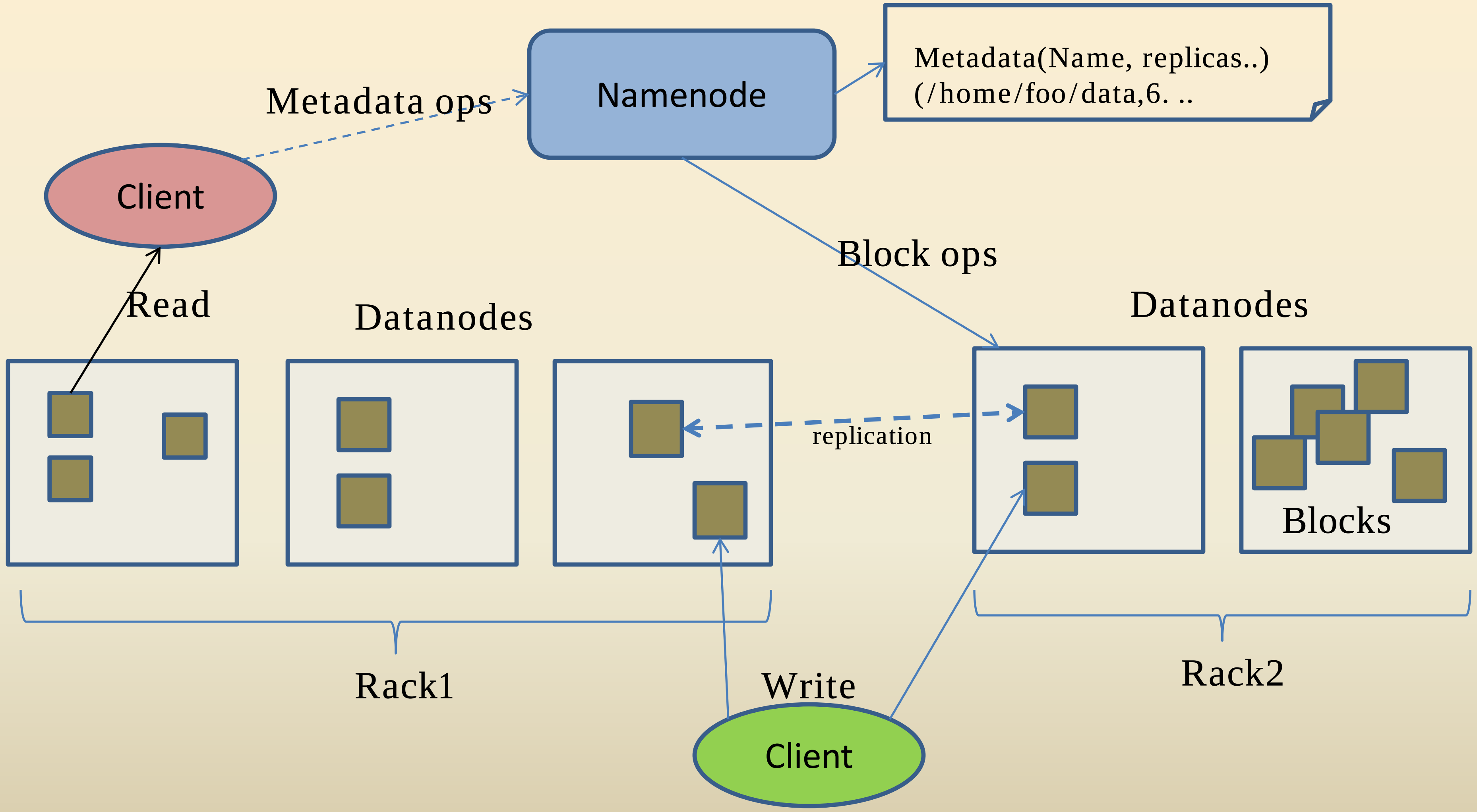
Fault tolerance

- Failure is the norm rather than exception
- A HDFS instance may consist of thousands of server machines, each storing part of the file system's data.
- Since we have huge number of components and that each component has non-trivial probability of failure means that there is always some component that is non-functional.
- Detection of faults and quick, automatic recovery from them is a core architectural goal of HDFS.

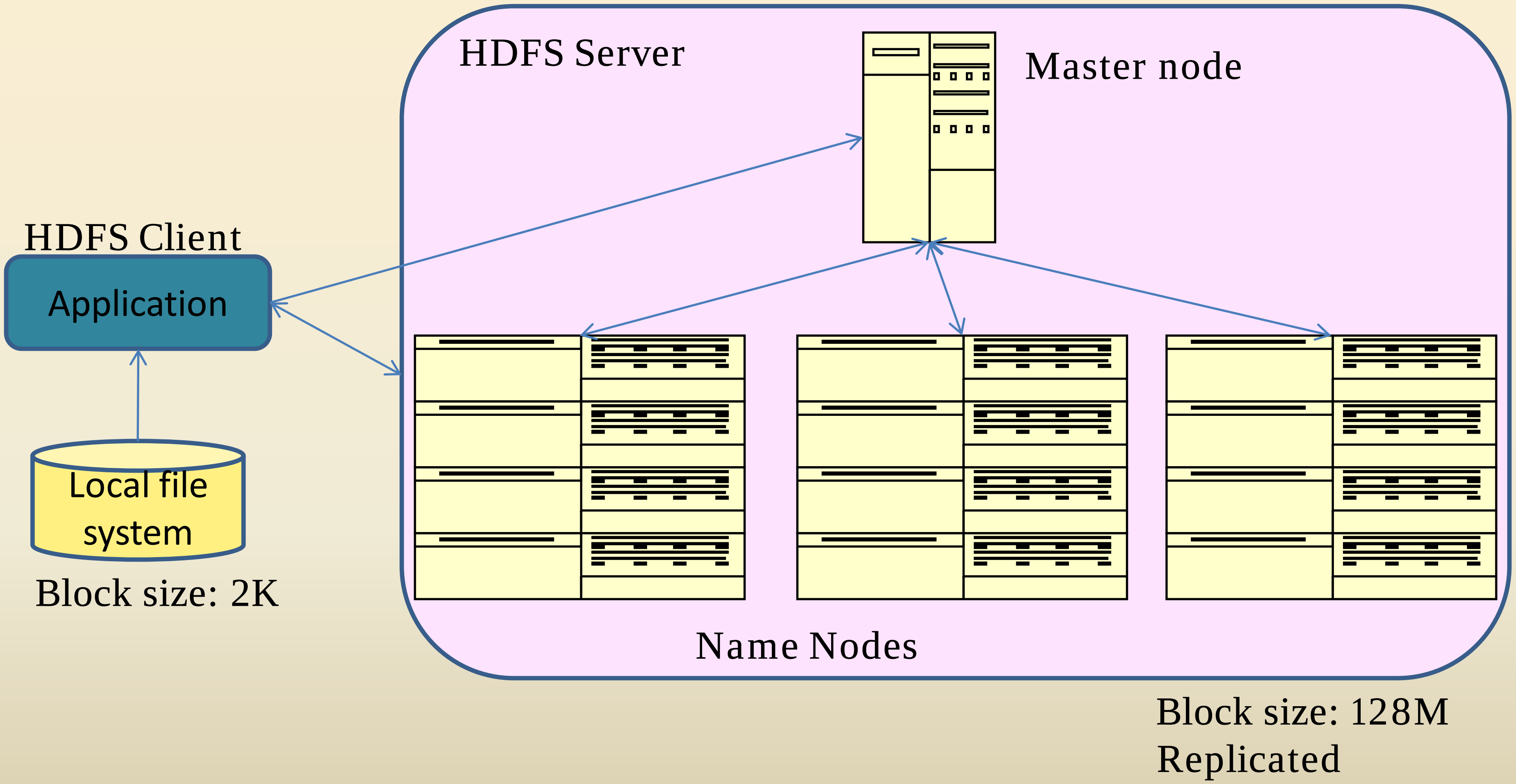
Namenode and Datanodes

- Master/slave architecture
- HDFS cluster consists of a single **Namenode**, a master server that manages the file system namespace and regulates access to files by clients.
- There are a number of **DataNodes** usually one per node in a cluster.
- The DataNodes manage storage attached to the nodes that they run on.
- HDFS exposes a file system namespace and allows user data to be stored in files.
- A file is split into one or more blocks and set of blocks are stored in DataNodes.
- DataNodes: serves read, write requests, performs block creation, deletion, and replication upon instruction from Namenode.

HDFS Architecture



Hadoop Distributed File System



File system Namespace

- Hierarchical file system with directories and files
- Create, remove, move, rename etc.
- Namenode maintains the file system
- Any meta information changes to the file system recorded by the Namenode.
- An application can specify the number of replicas of the file needed: replication factor of the file. This information is stored in the Namenode.

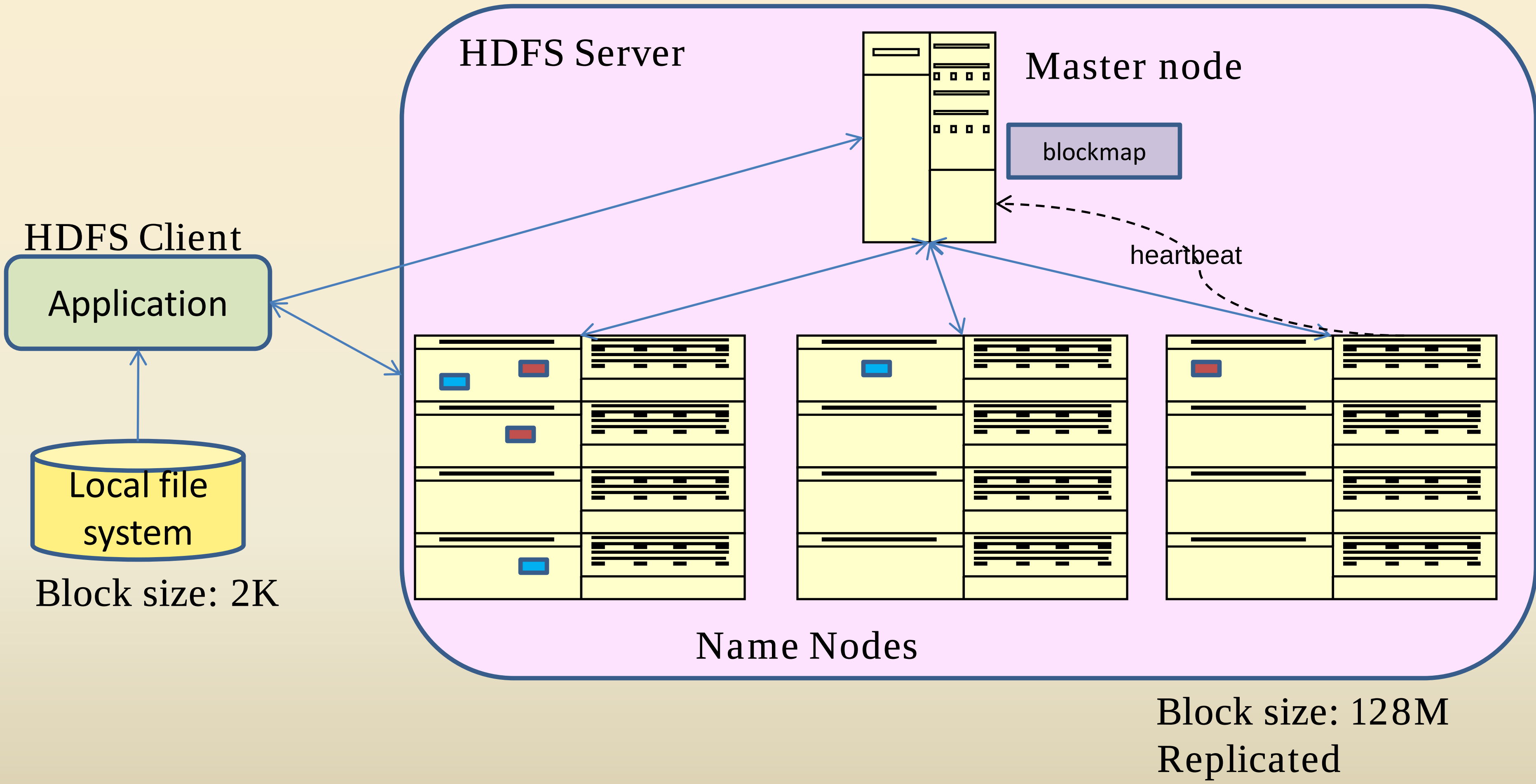
Data Replication

- HDFS is designed to store very large files across machines in a large cluster.
- Each file is a sequence of blocks.
- All blocks in the file except the last are of the same size.
- Blocks are replicated for fault tolerance.
- Block size and replicas are configurable per file.
- The Namenode receives a Heartbeat and a BlockReport from each DataNode in the cluster.
- BlockReport contains all the blocks on a Datanode.

Replica Placement

- The placement of the replicas is critical to HDFS reliability and performance.
- Optimizing replica placement distinguishes HDFS from other distributed file systems.
- Rack-aware replica placement:
  - Goal: improve reliability, availability and network bandwidth utilization
- Many racks, communication between racks are through switches
- Network bandwidth between machines on the same rack is greater than those in different racks.
- Namenode determines the rack id for each DataNode.
- Replicas are typically placed on unique racks
  - Simple but non-optimal
  - Writes are expensive
  - Typical replication factor is 3
- Replicas are placed: one on a node in a local rack, one on a different node in the local rack and one on a node in a different rack.
- 1/3 of the replica on a node, 2/3 on a rack and 1/3 distributed evenly across remaining racks.

Hadoop Distributed File System



Other features of HDFS

- Safe mode startup
- File system meta data
- Name node (master node)
- Datanode failure and heartbeat
- Namenode backup
- Re-replication
- Cluster rebalancing
- Meta-data disk failure

Summary

- We introduced MapReduce programming model for processing large scale data
- We discussed the supporting Hadoop Distributed File System
- Discussed big-data computing
- The concepts were illustrated using a simple example
- Relationship to Cloud Computing was also discussed

Take home points

- Data-intensive computing is becoming mainstream
- Newer programming models are addressing the data-deluge: mapreduce
- Newer infrastructures are being introduced: google and hadoop file systems; cloud computing
- Consider introducing it in your undergraduate curriculum and research

Demo

- VMware simulated Hadoop and MapReduce demo
- Remote access to NEXOS system at my Buffalo, NY lab (Inaccessible?)
  - 5-node HDFS running HDFS on Ubuntu 8.04
  - 1 –name node and 5 data-nodes
  - Each is an old commodity PC with 512 MB RAM, 120GB – 160GB external memory
  - Zeus (namenode), datanodes: hermes, dionysus, aphrodite, athena, theos
- Amazon Elastic Cloud Computing (EC2) supported MapReduce

References

1. Dean, J. and Ghemawat, S. 2008. **MapReduce: simplified data processing on large clusters.** *Communication of ACM* 51, 1 (Jan. 2008), 107-113.
2. Cloudera Videos by Aaron Kimball: http://www.cloudera.com/hadoop-training-basic
3. http://www.cse.buffalo.edu/faculty/bina/mapreduce.html
4. Apache Hadoop Tutorial: http://hadoop.apache.org http://hadoop.apache.org/core/docs/current/mapred_tutorial.html
5. The Hadoop Distributed File System: Architecture and Design by Apache Foundation Inc. http://hadoop.apache.org/core/docs/current/hdfs_design.html