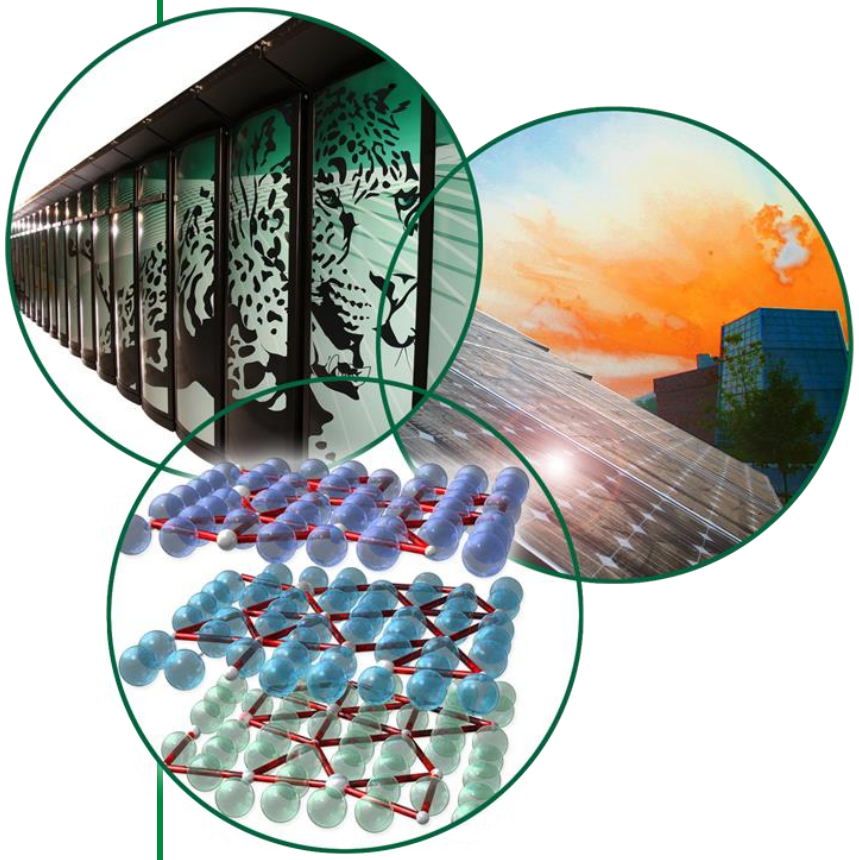
Analytics and Fusion for Distributed Big Data



Arjun Shankar, Ph.D.

Thanks also to: James Horey and Arvind Ramanathan

ORNL Computational Sciences and Engineering Division

SOS, Jekyll Island, Georgia
March 2013

U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Outline

- Context
- Examples of Distributed Big Data Analytics
- Emerging needs as Big Compute and Big Data analytics converge

Context

Volume and rates

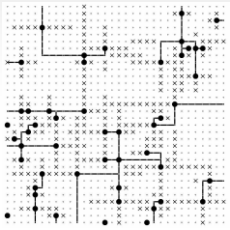
Twitter Updates	• 400 M/d
Facebook	• Likes/Comments: 2.7 B/d • Shared Contents: 30 B/m
World Emails	• 419 B/d
YouTube	• Storage: 76 PB/yr • Traffic: 16.2 EB/yr
World Social Media	• 1.8 ZB (x2 every 2 years)

Social Media

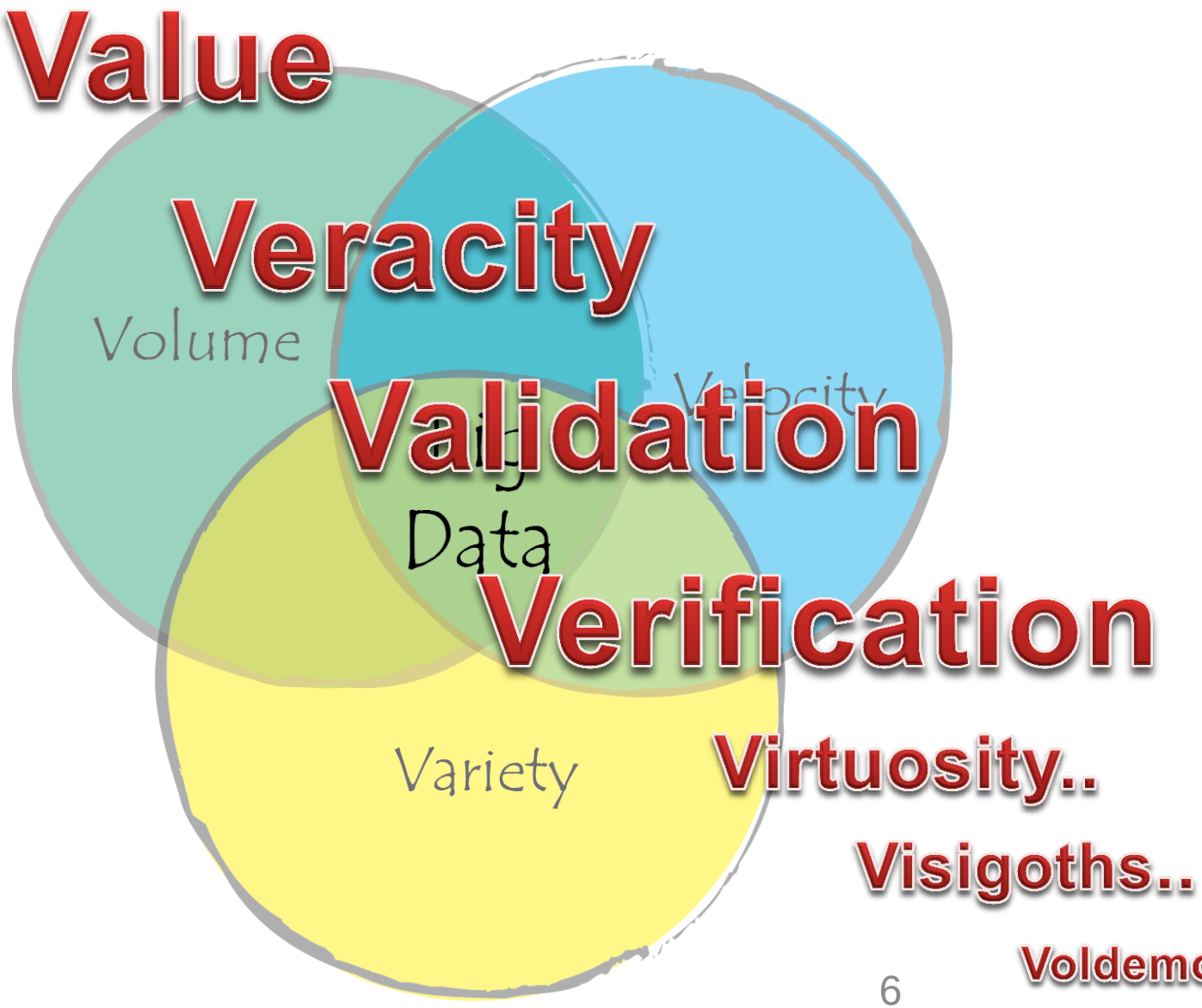
Volume and rates

2.5 m Telescope	200 GB/d
Ion Mobility Spectroscopy	10 TB/d
3D X-ray Diffraction Microscopy	24 TB/d
Boeing 737 cross-country flight	240 TB
Personal Location Data	1 PB/yr
Astrophysics Data	10 PB (2014)
Square Kilometer Array	480 PB/d

Sensors

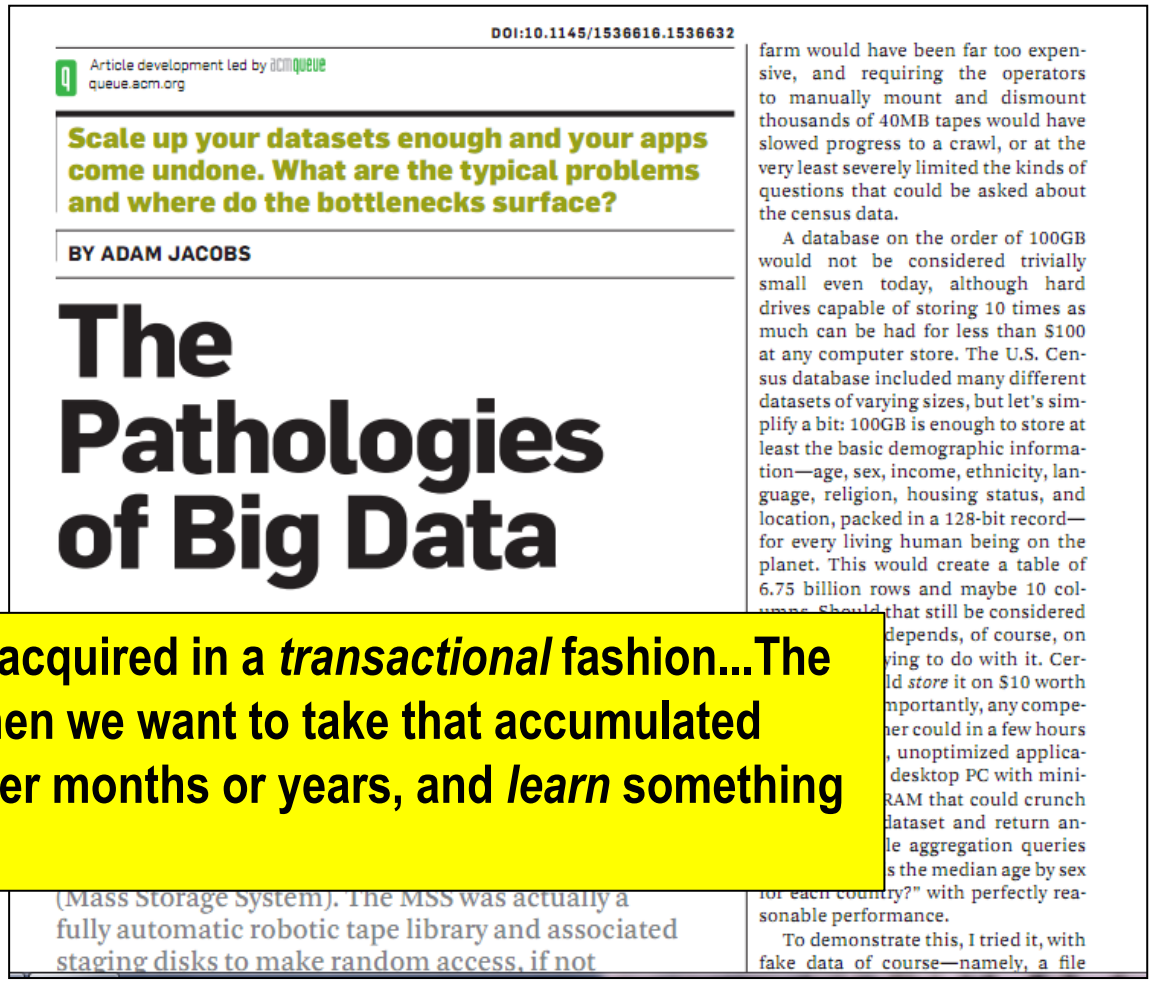


Big Data = Volume, Variety, Velocity



Learning from data is a major problem

How we harness our infrastructure to do data management and data analytics?



DOI:10.1145/1536616.1536632

Article development led by acmqueue
queue.acm.org

Scale up your datasets enough and your apps come undone. What are the typical problems and where do the bottlenecks surface?

BY ADAM JACOBS

The Pathologies of Big Data

farm would have been far too expensive, and requiring the operators to manually mount and dismount thousands of 40MB tapes would have slowed progress to a crawl, or at the very least severely limited the kinds of questions that could be asked about the census data.

A database on the order of 100GB would not be considered trivially small even today, although hard drives capable of storing 10 times as much can be had for less than $100 at any computer store. The U.S. Census database included many different datasets of varying sizes, but let's simplify a bit: 100GB is enough to store at least the basic demographic information—age, sex, income, ethnicity, language, religion, housing status, and location, packed in a 128-bit record—for every living human being on the planet. This would create a table of 6.75 billion rows and maybe 10 columns. Should that still be considered
depends, of course, on
ying to do with it. Cer-
ld *store* it on $10 worth
nportantly, any compe-
her could in a few hours
, unoptimized applica-
desktop PC with mini-
RAM that could crunch
dataset and return an-
le aggregation queries
s the median age by sex
for each country?" with perfectly reasonable performance.

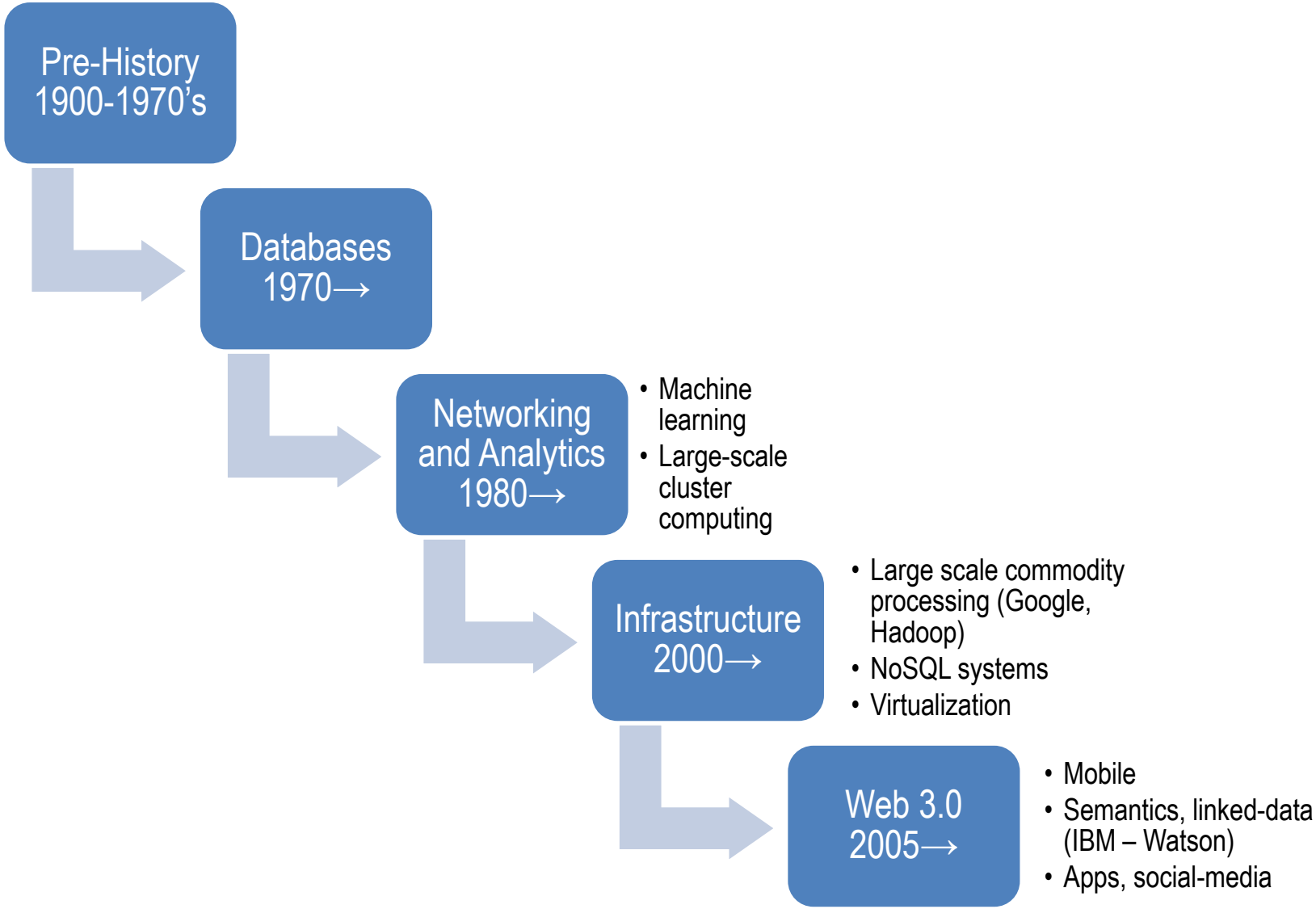
To demonstrate this, I tried it, with fake data of course—namely, a file

(Mass Storage System). The MSS was actually a fully automatic robotic tape library and associated staging disks to make random access, if not

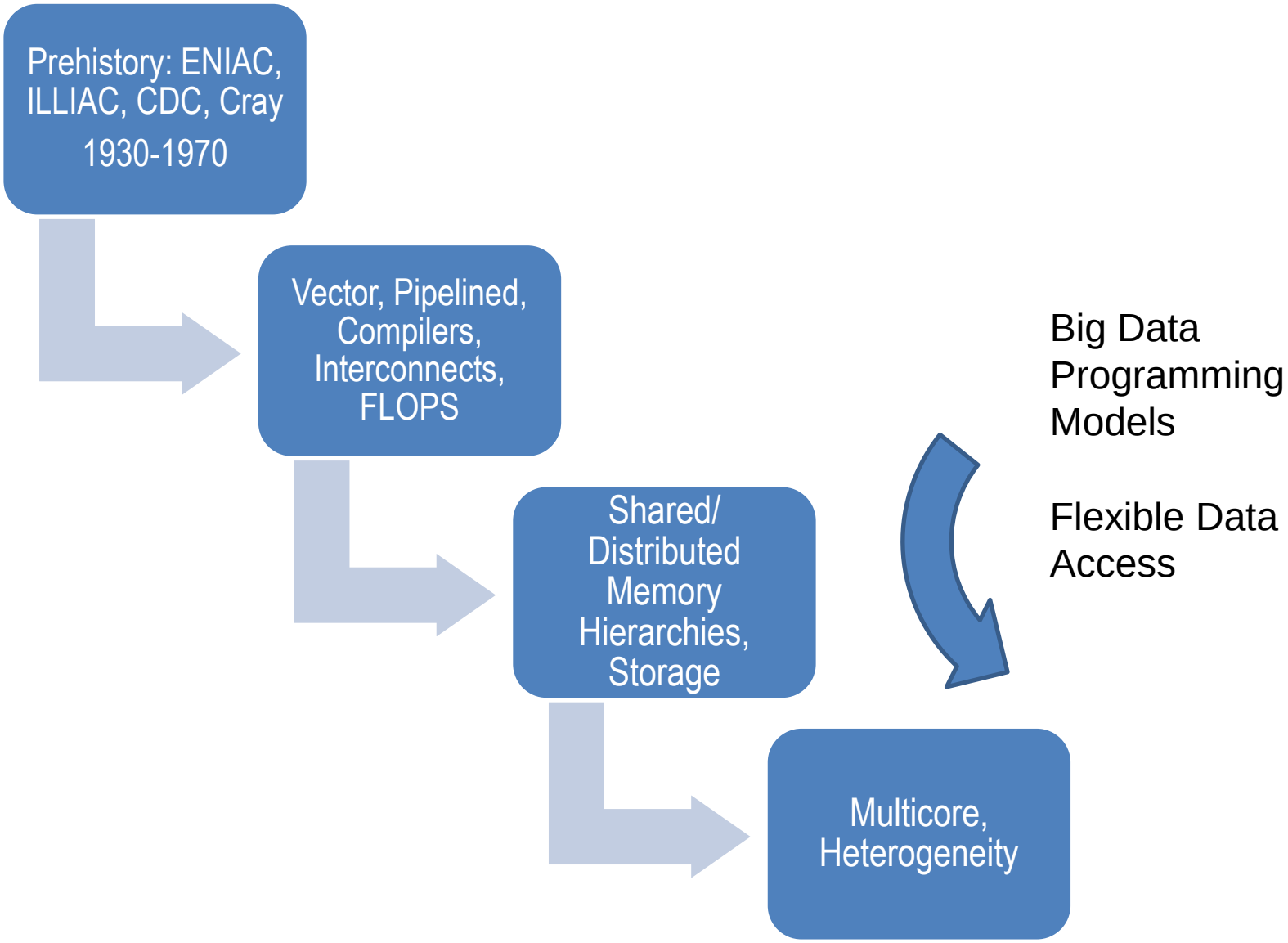
"Data is typically acquired in a *transactional* fashion...The trouble comes when we want to take that accumulated data, collected over months or years, and *learn* something from it."

Adam Jacobs, "The Pathologies of Big Data" 2009

Data Systems and Analytics



Big Computing Developments



Examples of Distributed Big Data Analytics

Large Scale Infrastructures for Analytics

- Centralized (aka Big Compute) HPC
  - Compute core – multicore, heterogeneous (cell, GPU, etc.)
  - Innovative Memory Hierarchies, Storage, and Interconnects
  - Specialized software stack (Fortran, OpenMP, XMT, CUDA,)
- Centralized Big Compute/Data platforms
  - MPP Databases (Netezza, Greenplum, Vertica), Column oriented
  - Document stores (MarkLogic)
  - Graph Engines (URIKA)
- Distributed Big Compute/Data platforms
  - Private and public cloud - virtualized storage and compute – Hadoop Ecosystem
  - EC2/Dryad/AppEngine/etc.
  - SciDB

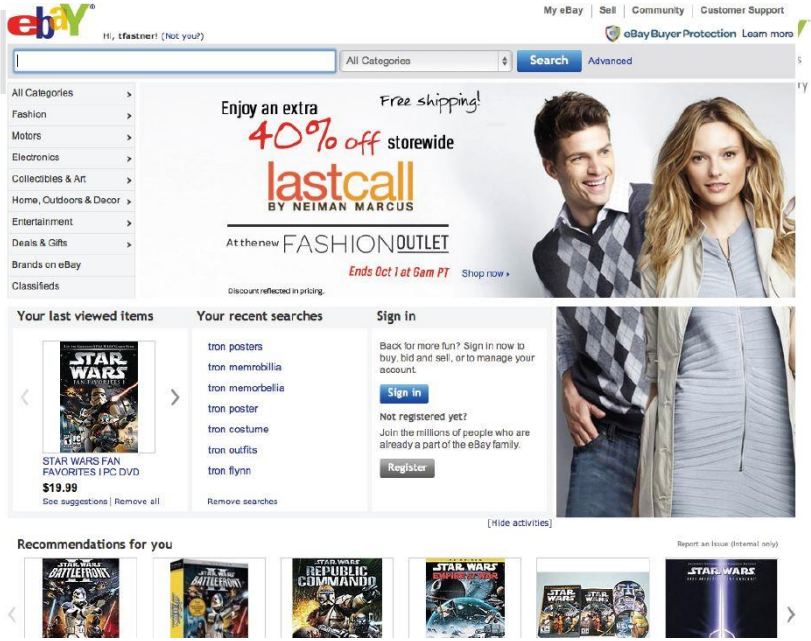
CAP Theorem Target

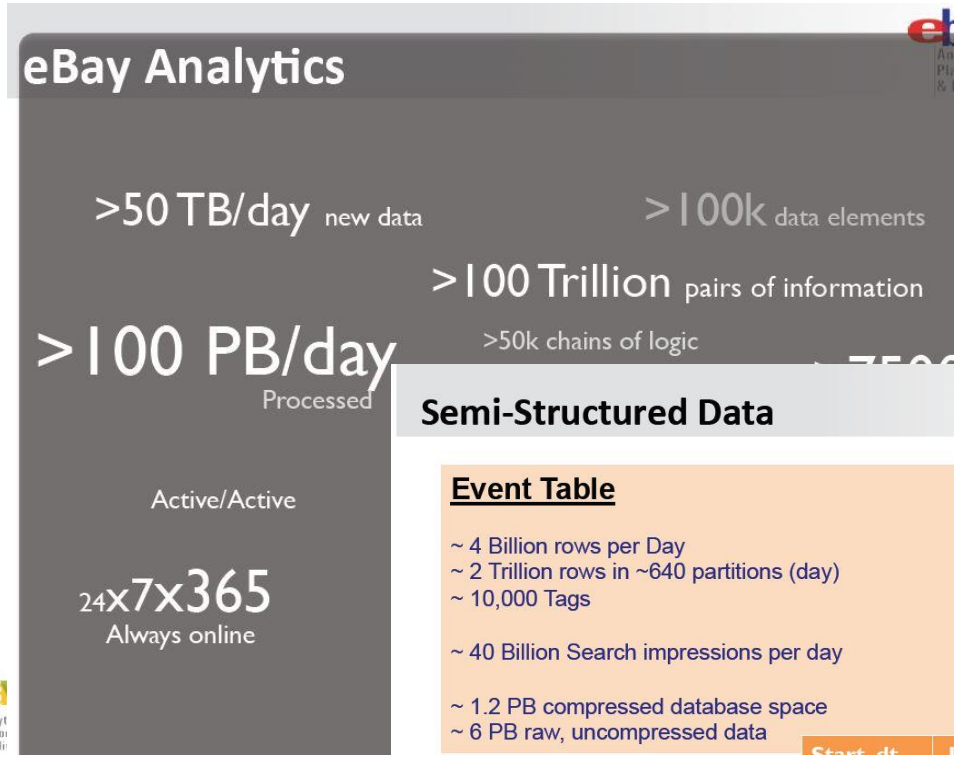
- Wide-Area Distributed Big Data platforms
  - Teragrid
  - Sensor networks
  - Web-services-based workflows

Big Data Analytics

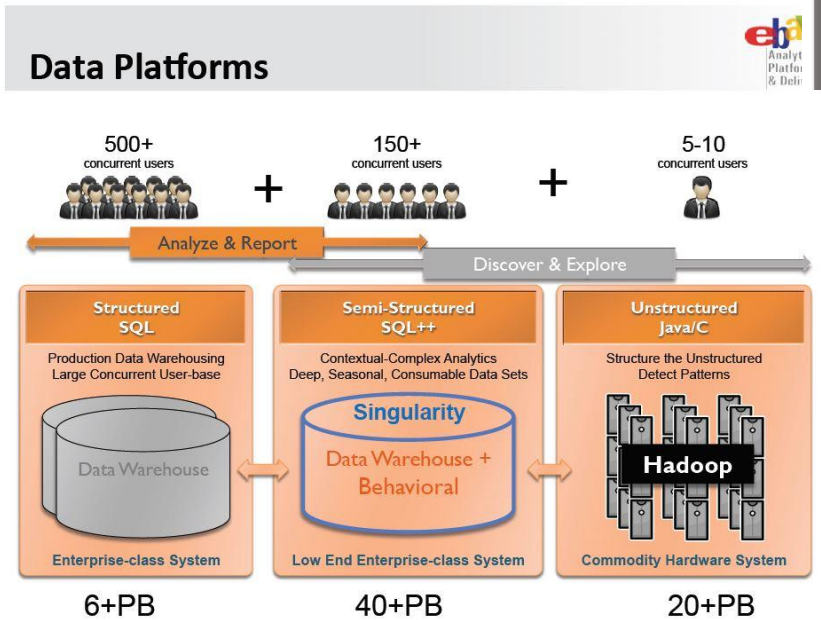
1. Centralized (aka Big Compute) HPC
   - First Principles, Floating Point Solvers, Prospective Data Generation
2. Centralized Big Compute/Data platforms
   - Discrete/Combinatorial, Retrieval, Indexing Lookup, Query, Graph-lookups, Data Ingest (ETL)
3. Distributed Big Compute/Data platforms
   - Virtualization and Utility Computing, Machine Learning Stack
4. Wide-Area Distributed Big Data platforms
   - Distributed Sensing, Correlate, and Actuate – Real-time, HPC Resource Use

(#2) Ebay – Big Data Analytics Example



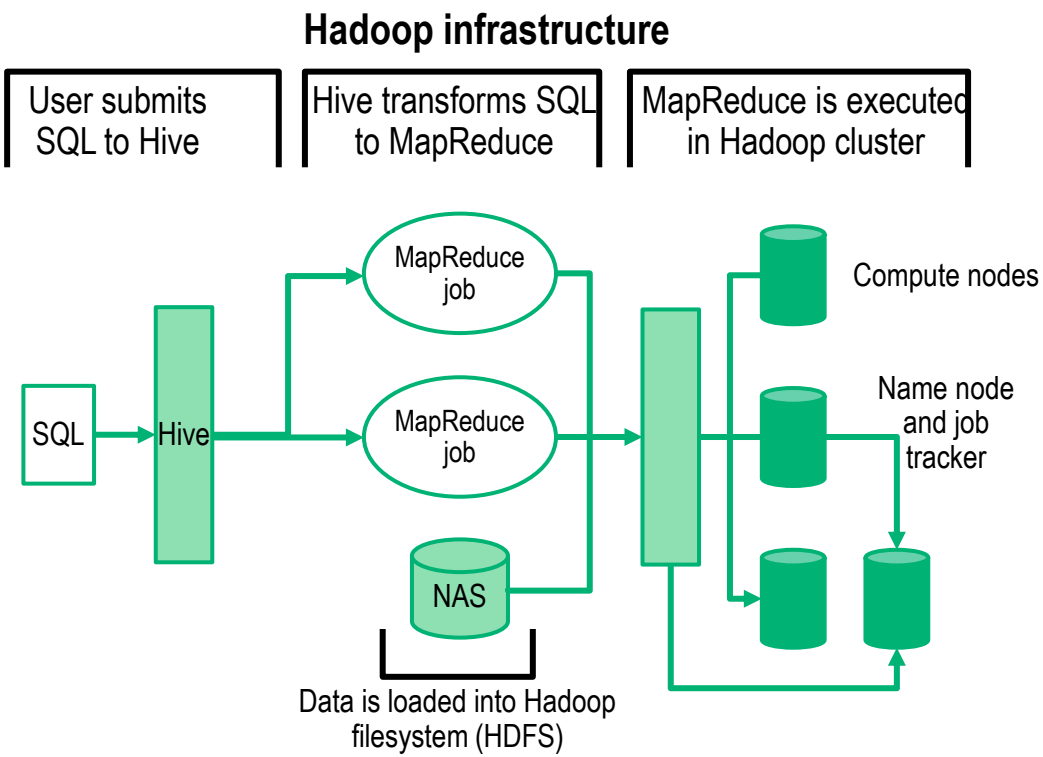


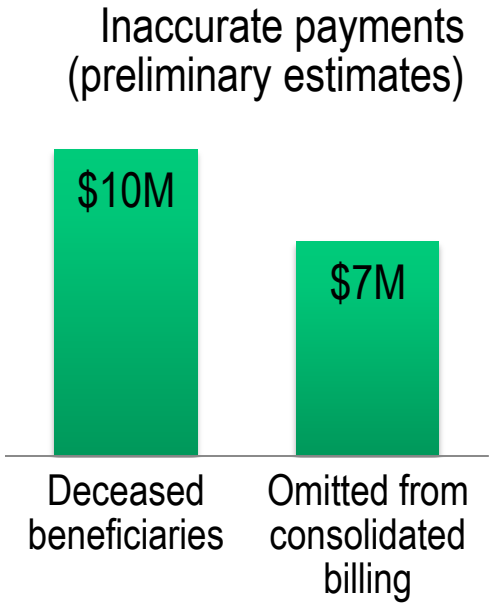
c. 2011

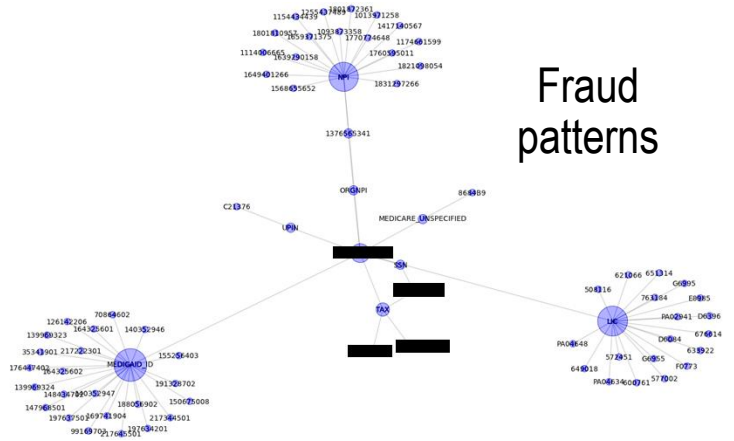


Slides from:Tom Fastener, 2011 Ebay Principal Architect, @ High-Performance Transaction Systems, Asilomar

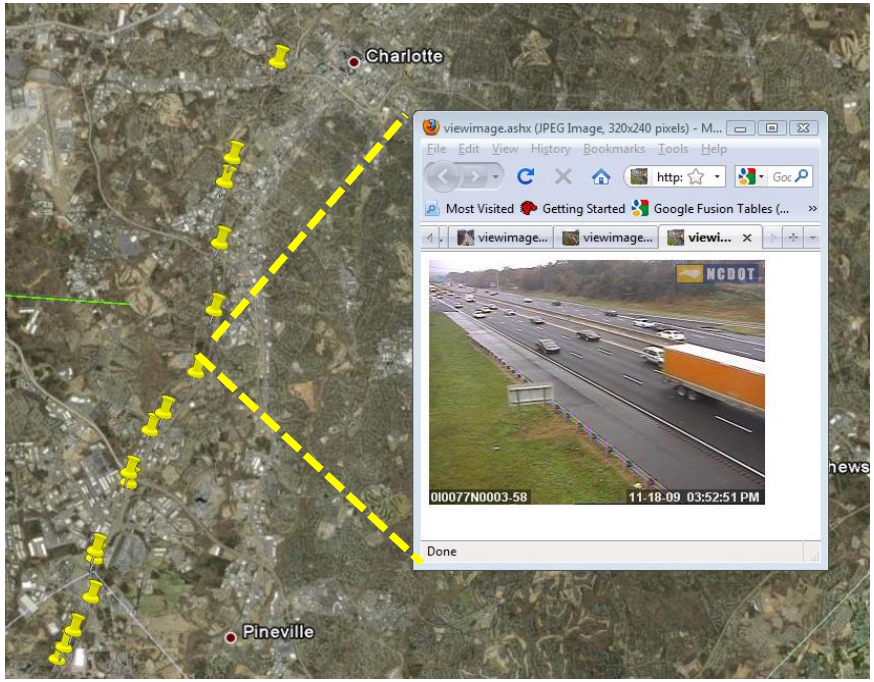
(#3) Healthcare Overpayments Analysis (@ORNL)

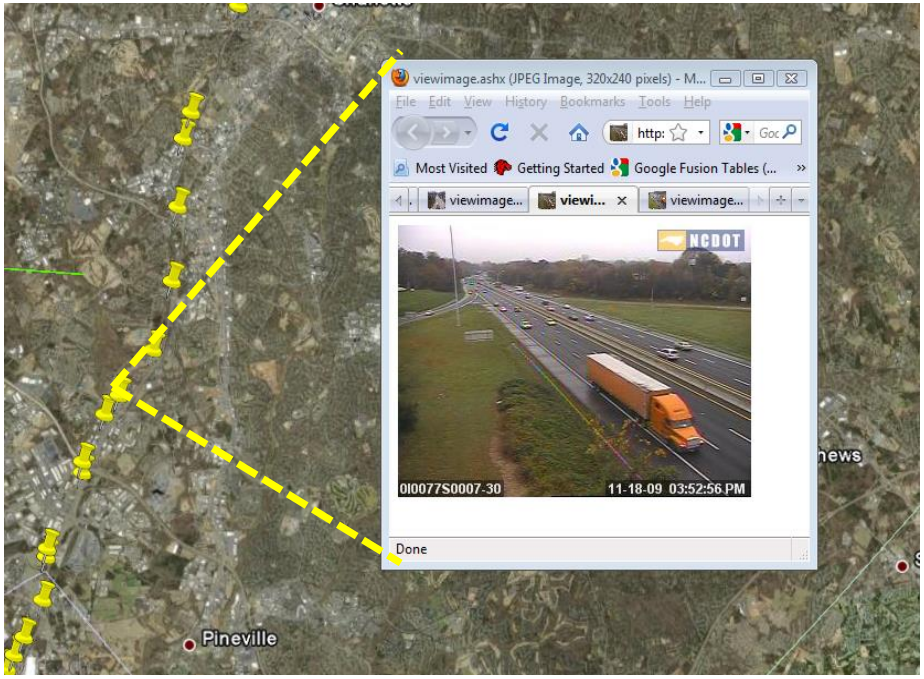






(#4) Real-Time Distributed Analytics



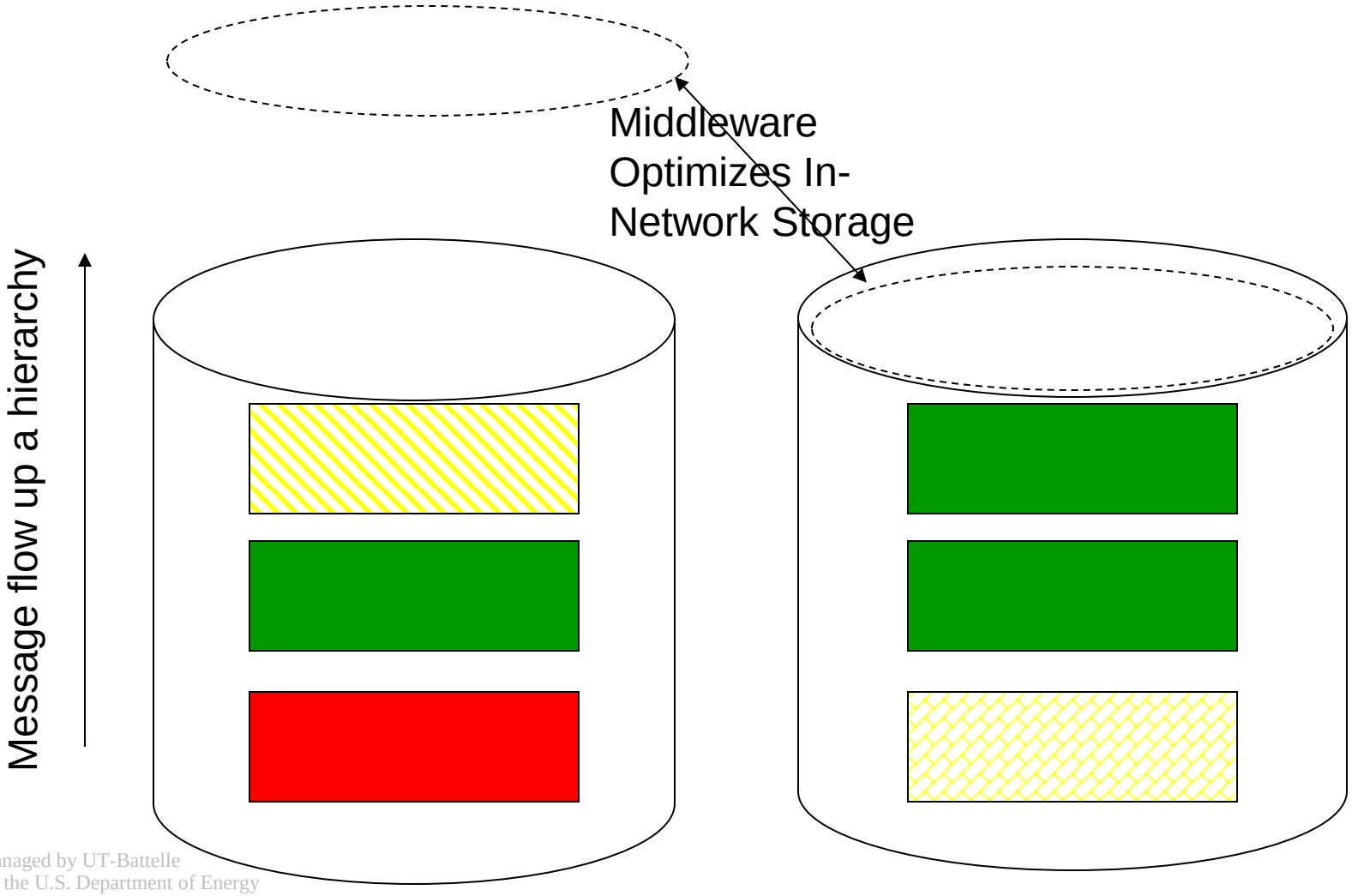


Centralized data analysis across infrastructure and data modalities is prohibitive (and often too late)!

Distributed Pattern Storage and Correlation

```
If (Sensor-noticed-X && Report-said-Y && Within-Last-Day)

        Notify!
```



Processing in the Network

Application specific open questions

- Value of Information
  - What to keep and what to drop (or send later)?
  - When to take notice?
- Infrastructure
  - Where in the hierarchy to compute?
  - How to set up the infrastructure to enable the forward joins?

Big Data Analytics on Big Compute - Emerging Applications

1. Graph analytics
2. Hypothesis driven to data driven analytics
3. Compute-analyze-compute paradigm

(1) Data Analytics Beyond Data-Parallelism

Data-Parallel → Graph-Parallel

Map Reduce

Feature Extraction

Cross Validation

Computing Sufficient Statistics

Graphical Models
Gibbs Sampling
Belief Propagation
Variational Opt.

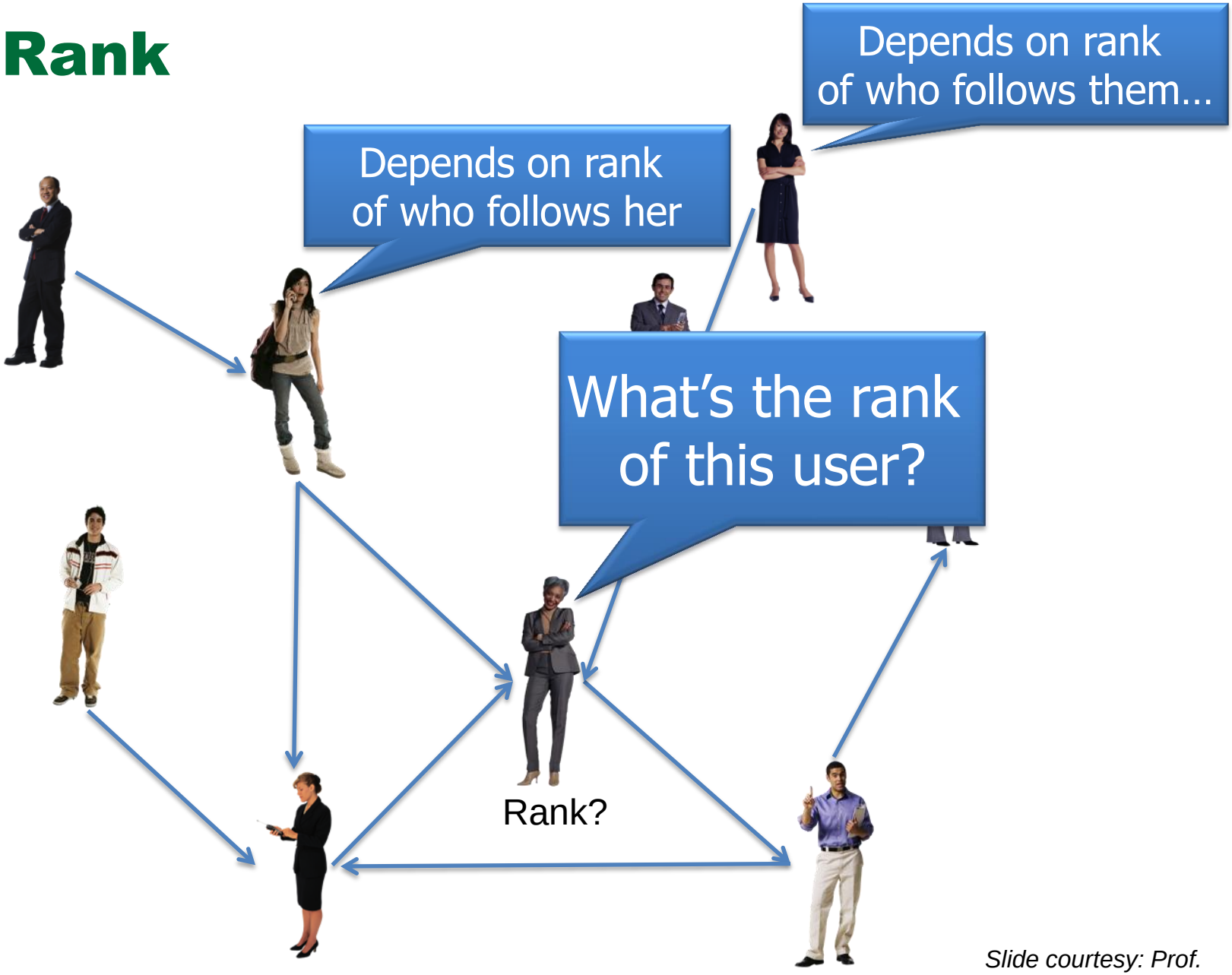
Semi-Supervised Learning
Label Propagation
CoEM

Collaborative Filtering
Tensor Factorization

Graph Analysis
PageRank
Triangle Counting

Slide courtesy: Prof. Carlos Guestrin's GraphLab Workshop, July 2012

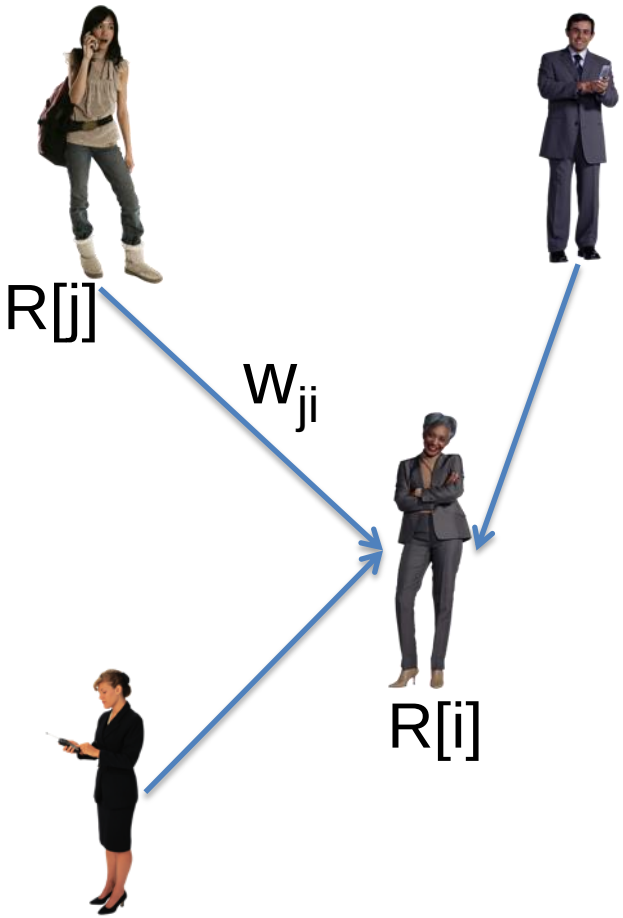
PageRank



Loops in graph → Must iterate!

Slide courtesy: Prof. Carlos Guestrin's GraphLab Workshop, July 2012

PageRank Iteration



Iterate until convergence:

"My rank is weighted average of my friends' ranks"

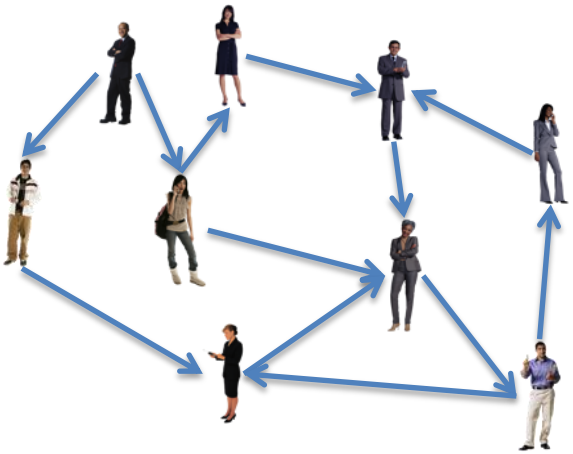
$$R[i] = \alpha + (1-\alpha) \sum_{(j,i)\in E} w_{ji} R[j]$$

- α is the random reset probability
- w_{ji} is the prob. transitioning (similarity) from j to i

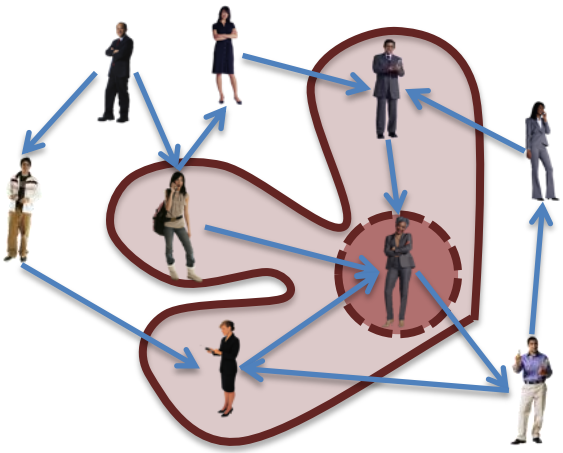
Slide courtesy: Prof. Carlos Guestrin's GraphLab Workshop, July 2012

Properties of Graph Parallel Algorithms

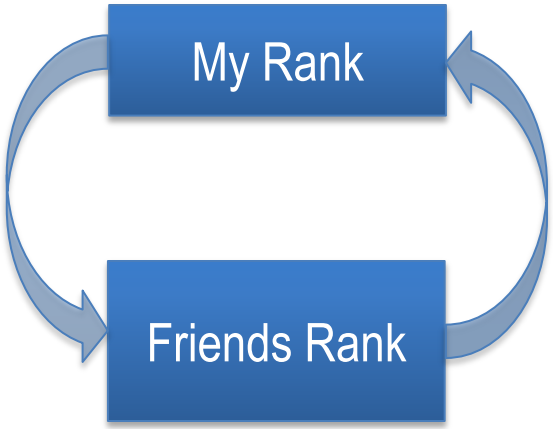
Dependency Graph



Local Updates



Iterative Computation



Slide courtesy: Prof. Carlos Guestrin's GraphLab Workshop, July 2012

Addressing Graph-Parallel ML

Data-Parallel ⟷ Graph-Parallel

Map Reduce

Feature Extraction

Cross Validation

Computing Sufficient Statistics

Graph-Parallel Abstraction

Graphical Models
Gibbs Sampling
Belief Propagation
Variational Opt.

Semi-Supervised Learning
Label Propagation
CoEM

Collaborative Filtering
Tensor Factorization

Data-Mining
PageRank
Triangle Counting

GraphLab software tailors abstractions for asynchronous updates, "natural" graphs, in-memory computations, etc.

Slide courtesy: Prof. Carlos Guestrin's GraphLab Workshop, July 2012

(2) Scaling Analytics: Let a Million Hypotheses Bloom …

- Classic scientific discovery scenario:
  - *Choose relevant D dimensions, evaluate on N samples*

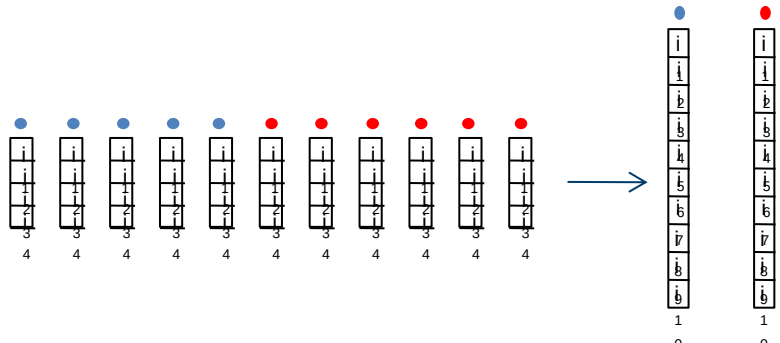
$$D < N \qquad N \to \infty \qquad D := Dimensionality \qquad N := \# Samples$$

- Modern data-driven analysis:
  - *Measure everything, hope to discover relevant D using N samples*

$$D > N \qquad N := fixed \qquad \varepsilon := Error \qquad \boxed{\left(1/\varepsilon\right)^{D}}$$

- The progression to ultrascale:
  - *Characterized by interdependency (not necessarily redundancy); many inter-related subsystems*

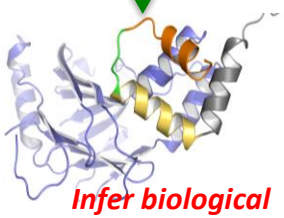
$$\boxed{D >> N}$$



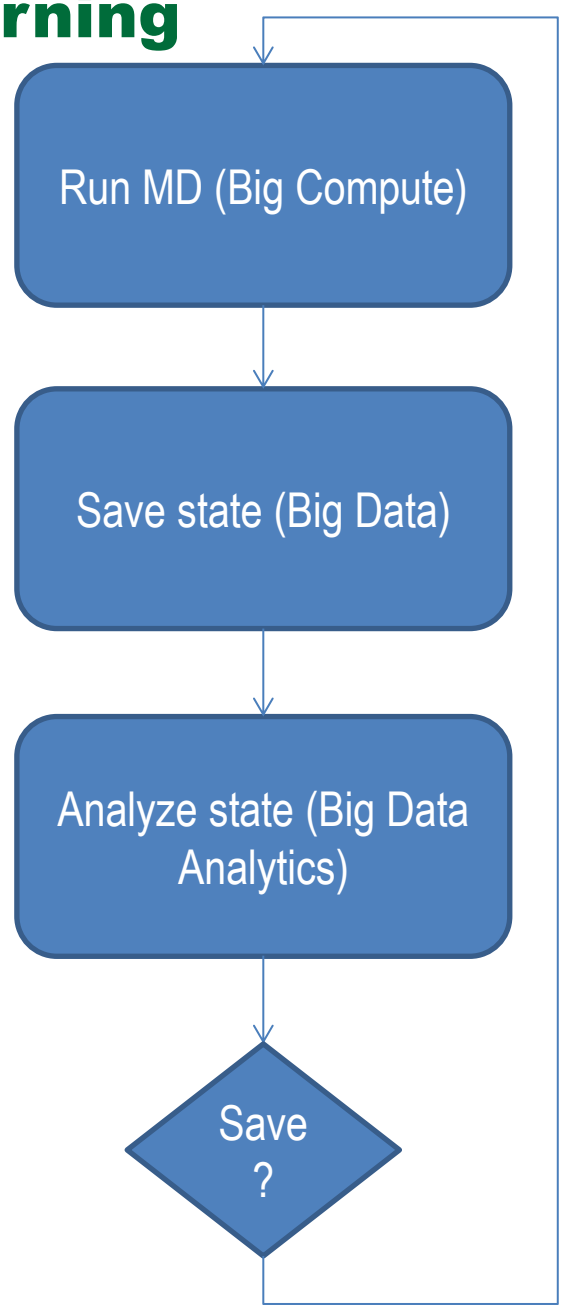
(3) Big Data Analytics Integrated with Big Compute: In Situ Machine Learning

Biological Data

High resolution all-atom simulations
(Jaguar/Titan)
> 1 petabyte



Infer biological function?



Big-Data Analytic Needs from Big-Compute

- Better programming abstractions and environments for
  - Machine learning suites
  - Automatic memory hierarchies management libraries
  - Automatic storage management libraries
  - Simplify communication and synchronization abstractions

Special Notice

Probabilistic Programming for Advancing Machine Learning (PPAML)

DARPA-SN-13-30

March 19, 2013

Probabilistic Programming is a new programming paradigm for managing uncertain information. The goal of PPAML is to advance machine learning by using probabilistic programming to: (1) dramatically increase the number of people who can successfully build machine learning applications, (2) make machine learning experts radically more effective, and (3) enable new applications that are impossible to conceive of using today's technology. In support of this overarching goal, PPAML has a number of sub-goals. Specifically, the sub-goals are: (1) to make machine learning model code shorter, (2) to reduce development time, (3) to facilitate the construction of richer models, (4) to require lower levels of expertise in building machine learning applications, and (5) to support the construction of integrated models.

those needing lots of data at the high end. Advances that are likely necessary to achieve this improvement include: (1) developing analyses that select the most appropriate solver or set of solvers given a particular model, query, and set of prior data; (2) improving the performance of existing solvers by incorporating ideas from the compiler optimization community; (3) compiling specific solvers to diverse hardware platforms in ways that optimize the resources of the hardware including multi-core machines, GPUs, cloud infrastructures, and potentially custom hardware; (4) developing new solvers; and (5) developing an API so that new solvers can be slotted into the solving infrastructure easily.

DARPA

Defense Advanced Research Projects Agency
675 North Randolph Street
Arlington, VA 22203-2114

DARPA is asking for programming techniques to simplify big data analytics… (19th March 2013)

Thank you!

Discussion, Questions?