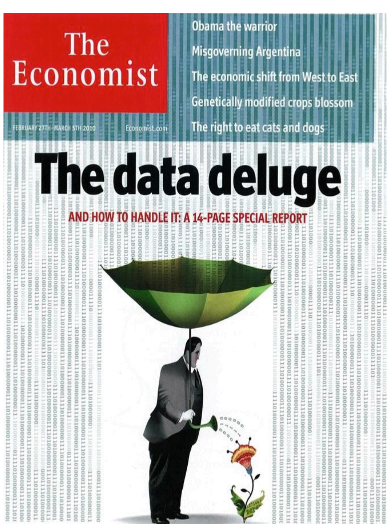
CSE 710 Seminar

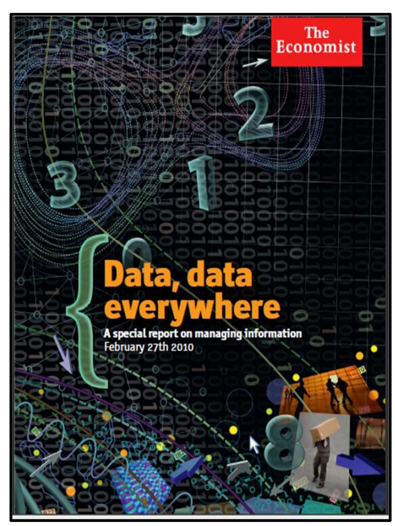
Wide Area Distributed File Systems

Tevfik Kosar, Ph.D.

Week 1: January 23, 2012

Data Deluge



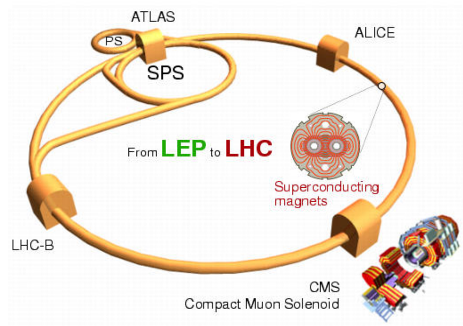


Big Data in Science

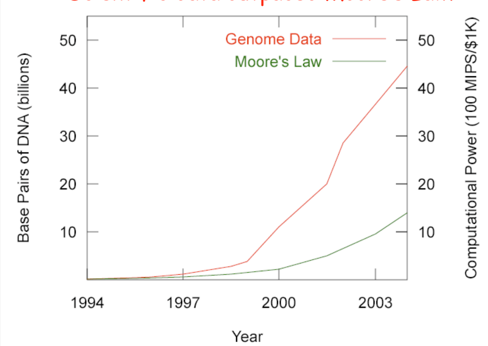
Demand for data in all areas of science!

Application	Area	Data Volume
VISTA	Astronomy	100 TB/year
LIGO	Astrophysics	250 TB/year
WCER EVP	Educational Technology	500 TB/year
LSST	Astronomy	1000 TB/year
BLAST	Bioinformatics	1000 TB/year
ATLAS/CMS	High Energy Physics	5000 TB/year

The Large Hadron Collider (LHC)



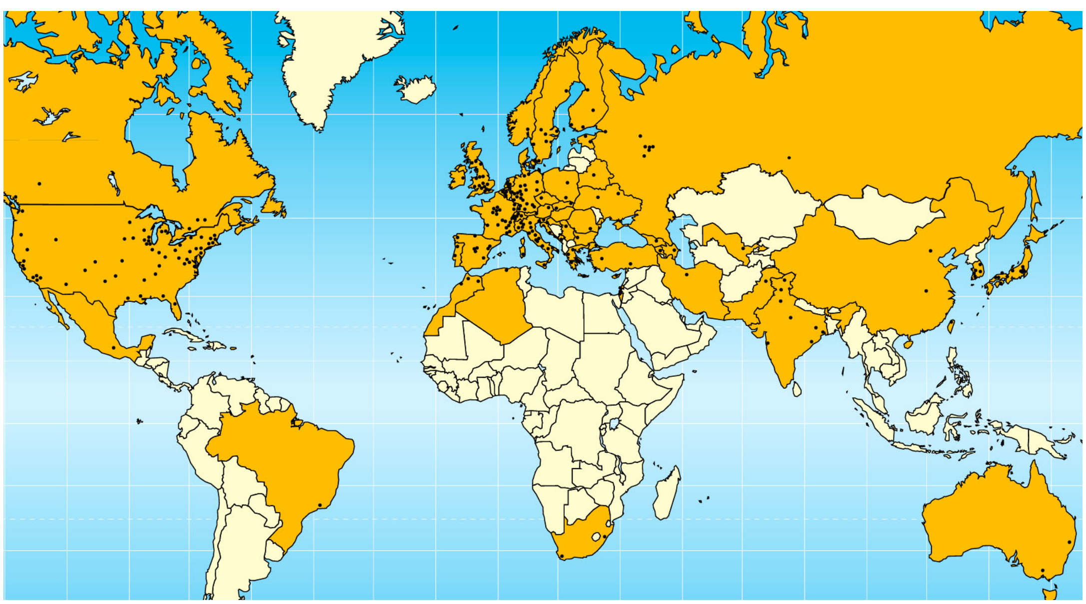
Scientific data outpaced Moore's Law!



Demand for data brings demand for computational power:

ATLAS and CMS applications alone require more than 100,000 CPUs!

ATLAS Participating Sites



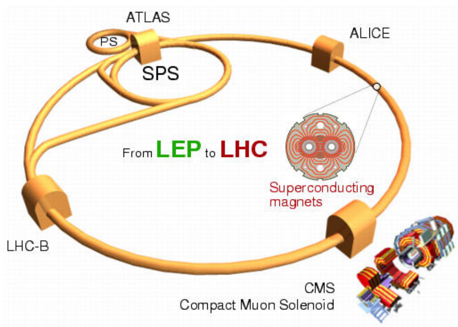
ATLAS: High Energy Physics project

*Generates **10 PB** data/year --> distributed to and processed by 1000s of researchers at **200 institutions** in **50 countries**.*

Big Data Everywhere

Science

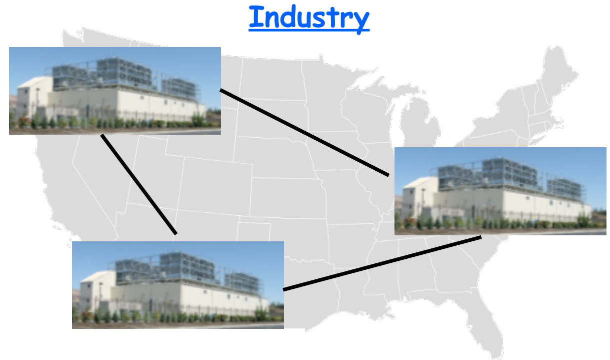
The Large Hadron Collider (LHC)



- 1 PB is now considered "small" for many science applications today

- For most, their data is distributed across several sites

Industry



A survey among 106 organizations operating two or more data centers:

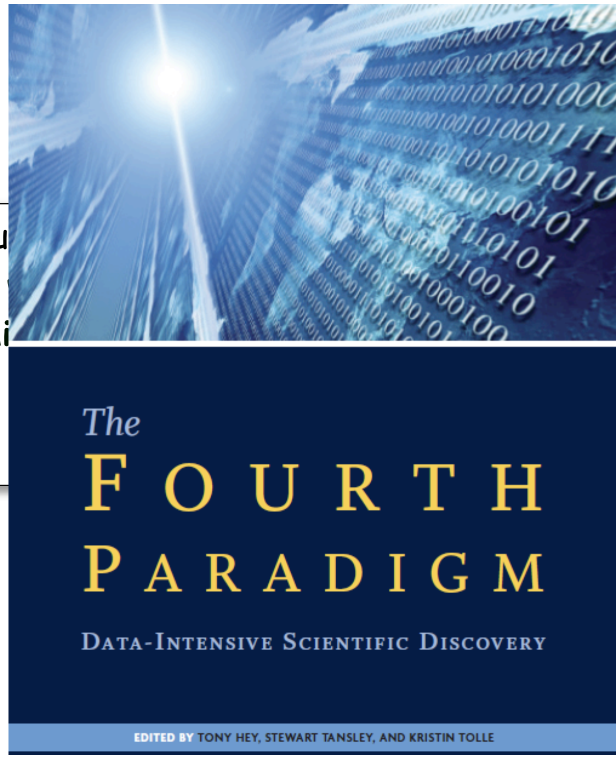
- 50% has more than 1 PB in their primary data center

- 77% run replication among three or more sites

Particle Physics Large Hadron Collider (15PB)	Human Genomics (7000PB) 1GB / person 200PB+ captured	World Wide Web (10PB)	Wikipedia 400K Articles/ Year
Annual Email Traffic, no spam (300PB+)	Internet Archive (1PB+)	Estimated On-line RAM in Google (8PB)	Personal Digital Photos (1000PB+)
200 of London's Traffic Cams (8TB/day)	Walmart Transaction DB (500TB)	Typical Oil Company (350TB+)	Merck Bio Research DB (1.5TB/qtr)
UPMC Hospitals Imaging Data (500TB/yr)	MIT Babytalk Speech Experiment (1.4PB)	Terashake Earthquake Model of LA Basin (1PB)	One Day of Instant Messaging (1TB)

Total digital data to be created this year **270,000PB** (IDC)

Future Trends



"In the futu ence and engineering ty to **leverage thi** ed in digital form."

yberinfrastructure

Emergence of a **Fourth Research Paradigm**

Thousand years ago – **Experimental Science**

- Description of natural phenomena

Last few hundred years – **Theoretical Science**

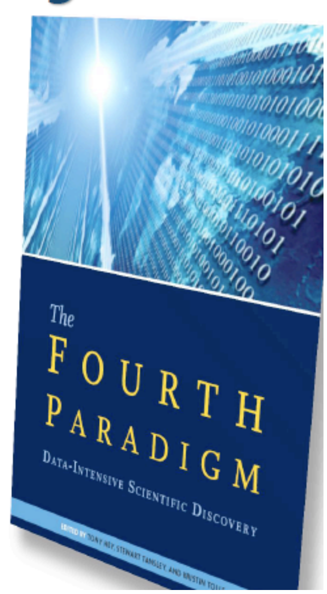
- Newton's Laws, Maxwell's Equations…

Last few decades – **Computational Science**

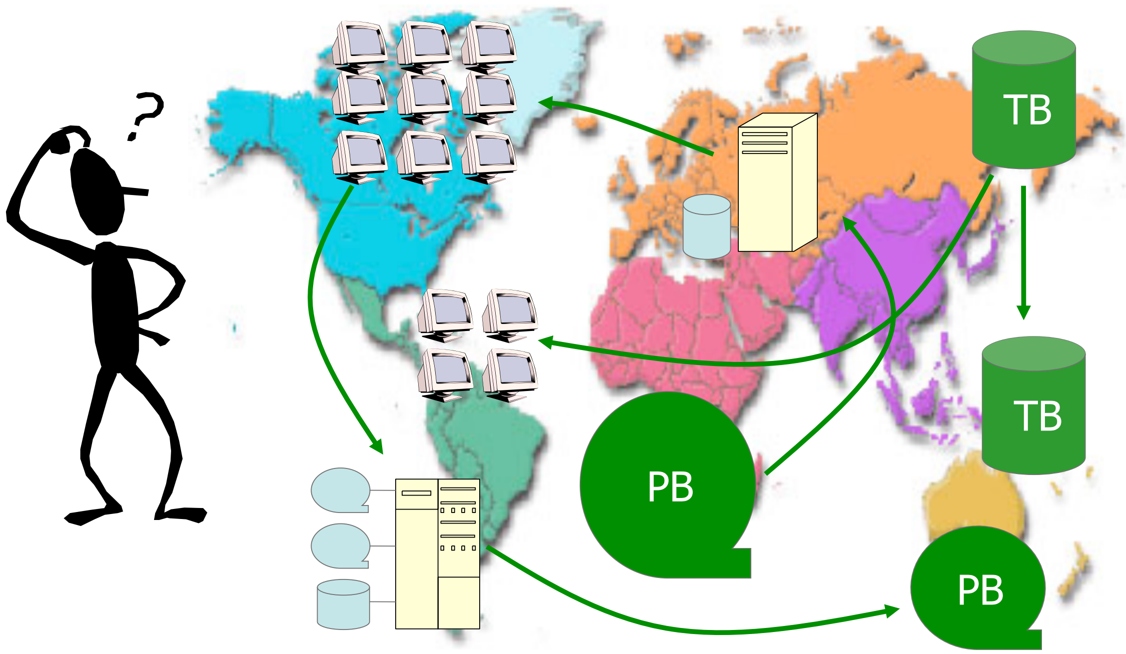
- Simulation of complex phenomena

Today – **Data-Intensive Science**

- Large-scale data analysis and data mining; visualization and exploration; scholarly communication and dissemination



How to Access and Process Distributed Data?

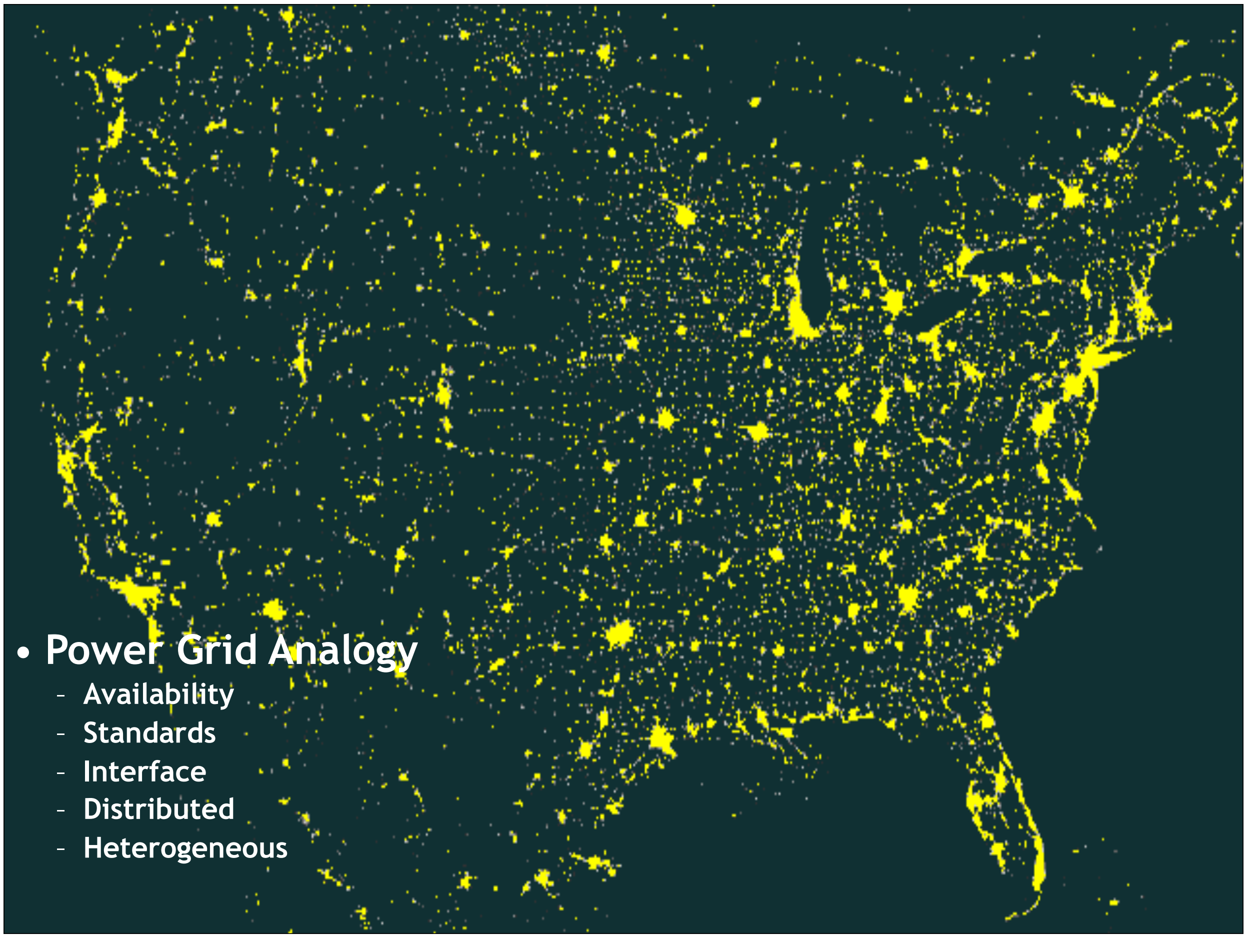


Ian Foster
UChicago/Argonne

Carl Kesselman
ISI/USC

They have coined the term "Grid Computing" in 1996!

In 2002, "Grid Computing" selected one of the Top 10 Emerging Technologies that will change the world!

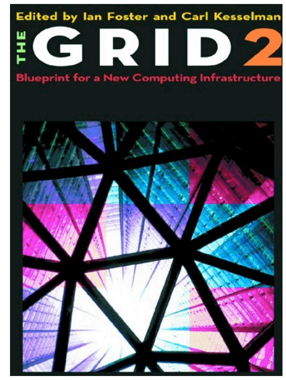


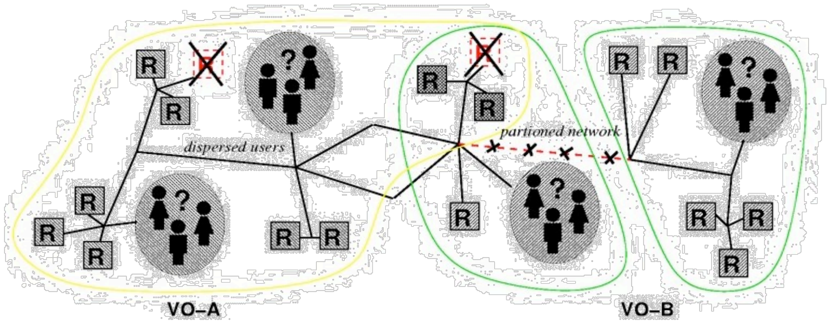
Defining Grid Computing

- There are several competing definitions for "The Grid" and Grid computing
- These definitions tend to focus on:
  - Implementation of Distributed computing
  - A common set of interfaces, tools and APIs
  - inter-institutional, spanning multiple administrative domains
  - "The Virtualization of Resources" abstraction of resources

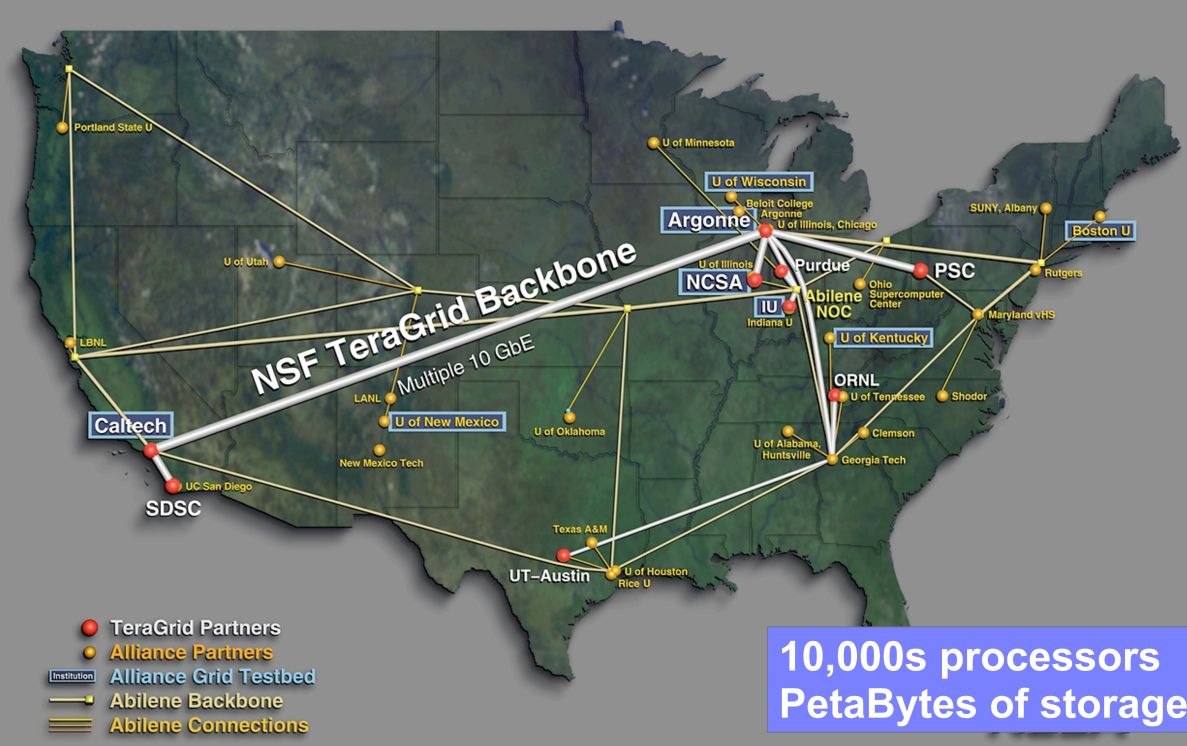
According to Foster & Kesselman:

"coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations" (The Anatomy of the Grid, 2001)





TeraGrid and the Alliance



10,000s processors
PetaBytes of storage

Desktop Grids

SETI@home:
- Detect any alien signals received through Arecibo radio telescope
- Uses the idle cycles of computers to analyze the data generated from the telescope

Others: Folding@home, FightAids@home

- Over 2,000,000 active participants, most of whom run screensaver on home PC
- Over a cumulative 20 TeraFlop/sec
  - TeraGrid: 40 TeraFlop/src
- Cost: $700K!!
  - TeraGrid: > $100M

Emergence of Cloud Computing

Grid Computing
- Solving large problems with parallel computing
- Made mainstream by Globus Alliance

Utility Computing
- Offering computing resources as a metered service
- Introduced in late 1990s

Software as a Service

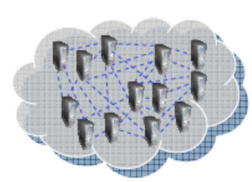
Network-based subscriptions to applications

Gained momentum in 2001

Cloud Computing

Next-Generation Internet computing

Next-Generation Data Centers



Commercial clouds

amazon webservices
Amazon Elastic Compute Cloud (Amazon EC2) - Beta
Appistry
hp
vmware
Joyent
GOGRID beta
A ServePath Company
citrix
3tera
Q-layer
MOSSO
the hosting cloud
b-hive
rPath
CYCLECOMPUTING
Microsoft
VERIO
An NTT Communications Company
Engine Yard
Sun
TAP INTO THE
POWER OF NETWORK.COM
VMOps

Commercial Clouds Growing...

- Microsoft [NYTimes, 2008]
  - 150,000 machines
  - Growth rate of 10,000 per month
  - Largest datacenter: 48,000 machines
  - 80,000 total running Bing
- Yahoo! [Hadoop Summit, 2009]
  - 25,000 machines
  - Split into clusters of 4000
- AWS EC2 (Oct 2009)
  - 40,000 machines
  - 8 cores/machine
- Google
  - (Rumored) several hundreds of thousands of machines

Distributed File Systems

- Data sharing of multiple users
- User mobility
- Data location transparency
- Data location independence
- Replications and increased availability

- Not all DFS are the same:
  - Local-area vs Wide area DFS
  - Fully Distributed FS vs DFS requiring central coordinator

Issues in Distributed File Systems

- Naming (global name space)
- Performance (Caching, data access)
- Consistency (when/how to update/synch?)
- Reliability (replication, recovery)
- Security (user privacy, access controls)
- Virtualization

Naming of Distributed Files

- *Naming* – mapping between logical and physical objects.
- A *transparent* DFS hides the location where in the network the file is stored.
- **Location transparency** – file name does not reveal the file's physical storage location.
  - File name denotes a specific, hidden, set of physical disk blocks.
  - Convenient way to share data.
  - Could expose correspondence between component units and machines.
- **Location independence** – file name does not need to be changed when the file's physical storage location changes.
  - Better file abstraction.
  - Promotes sharing the storage space itself.
  - Separates the naming hierarchy from the storage-devices hierarchy.

DFS - File Access Performance

- Reduce network traffic by retaining recently accessed disk blocks in local *cache*
- Repeated accesses to the same information can be handled locally.
  - All accesses are performed on the cached copy.
- If needed data not already cached, copy of data brought from the server to the local cache.
  - Copies of parts of file may be scattered in different caches.
- *Cache-consistency* problem – keeping the cached copies consistent with the master file.
  - Especially on write operations

DFS - File Caches

- In client memory
  - Performance speed up; faster access
  - Good when local usage is transient
  - Enables diskless workstations
- On client disk
  - Good when local usage dominates (e.g., AFS)
  - Caches larger files
  - Helps protect clients from server crashes

DFS - Cache Update Policies

- When does the client update the master file?
  - I.e. when is cached data written from the cache to the file?
- *Write-through* – write data through to disk ASAP
  - I.e., following *write*() or *put*(), same as on local disks.
  - Reliable, but poor performance.
- *Delayed-write* – cache and then write to the server later.
  - Write operations complete quickly; some data may be overwritten in cache, saving needless network I/O.
  - Poor reliability
    - unwritten data may be lost when client machine crashes
    - Inconsistent data
  - Variation – scan cache at regular intervals and flush *dirty* blocks.

DFS - File Consistency

- Is locally cached copy of the data consistent with the master copy?
- *Client*-initiated approach
  - Client initiates a validity check with server.
  - Server verifies local data with the master copy
    - E.g., time stamps, etc.
- *Server*-initiated approach
  - Server records (parts of) files cached in each client.
  - When server detects a potential inconsistency, it reacts

DFS - File Server Semantics

- *Stateful* Service
- Client *opens* a file (as in Unix & Windows).
- Server fetches information about file from disk, stores in server memory,
  - Returns to client a *connection identifier* unique to client and open file.
  - Identifier used for subsequent accesses until session ends.
- Server must reclaim space used by no longer active clients.
- Increased performance; fewer disk accesses.
- Server retains knowledge about file
  - E.g., read ahead next blocks for sequential access
  - E.g., file locking for managing writes
    - Windows

DFS - File Server Semantics

- *Stateless* Service
  - Avoids *state* information in server by making each request self-contained.
  - Each request identifies the file and position in the file.
  - No need to establish and terminate a connection by open and close operations.

  - Poor support for locking or synchronization among concurrent accesses

DFS - Server Semantics Comparison

- Failure Recovery: S*tateful server* loses all volatile state in a crash.
  - Restore state by recovery protocol based on a dialog with clients.
  - Server needs to be aware of crashed client processes
    - orphan detection and elimination.
- Failure Recovery: *Stateless server* failure and recovery are almost unnoticeable.
  - Newly restarted server responds to self-contained requests without difficulty.

DFS - Replication

- *Replicas* of the same file reside on failure-independent machines.
- Improves availability and can shorten service time.
- Naming scheme maps a replicated file name to a particular replica.
  - Existence of replicas should be invisible to higher levels.
  - Replicas must be distinguished from one another by different lower-level names.
- Updates
  - Replicas of a file denote the same logical entity
  - Update to any replica *must* be reflected on all other replicas.

CSE 710 Seminar

- State-of-the-art research, development, and deployment efforts in wide-area distributed file systems on clustered, grid, and cloud infrastructures.

- We will review 28 papers on topics such as:
  - File System Design Decisions
  - Performance, Scalability, and Consistency issues in File Systems
  - Traditional Distributed File Systems
  - Parallel Cluster File Systems
  - Wide Area Distributed File Systems
  - Cloud File Systems
  - Commercial vs Open Source File System Solutions

CSE 710 Seminar *(cont.)*

- Early Distributed File Systems
  - NFS (Sun)
  - AFS (CMU)
  - Coda (CMU)
  - xFS (UC Berkeley)
- Parallel Cluster File Systems
  - GPFS (IBM)
  - Panasas (CMU/Panasas)
  - PVFS (Clemson/Argonne)
  - Lustre (Cluster Inc)
  - Nache (IBM)
  - Panache (IBM)

CSE 710 Seminar *(cont.)*

- Wide Area File Systems
  - OceanStore (UC Berkeley)
  - WheelFS (MIT)
  - Shark (NYU)
  - XUFS (UT-Austin)
  - Ceph (UC-Santa Cruz)

  - Google FS (Google)
  - Hadoop DFS (Yahoo!)
  - Pangea (HPLabs)
  - zFS (IBM)

CSE 710 Seminar *(cont.)*

- Distributed Storage Management
  - Bigtable (Google)
  - Dynamo (Amazon)
  - PNUTS (Yahoo!)
  - Cassandra (Facebook)
  - Spyglass (NetApp)
  - Megastore (Google)

- File Systems for Mobile/Portable Computing
  - Coda (CMU)
  - BlueFS (UMich)
  - ...

Reading List

- The list of papers to be discussed is available at:

http://www.cse.buffalo.edu/faculty/tkosar/cse710/reading_list.htm

- Each student will be responsible for:
  - Presenting 1 paper
  - Writing reviews for 2 other papers
  - Reading and contributing the discussion of all the other papers (ask questions, make comments etc)
- We will be discussing 2 papers each class

Paper Presentations

- Each student will present 1 paper:
- 25-30 minutes each + 20-25 minutes Q&A/discussion
- No more than 10 slides
- Presenters should meet with me on Friday before their presentation to show their slides!
- Office hours: Fri 11:30am - 1:00pm

Paper Reviews

- 1 paragraph executive summary *(what are the authors trying to achieve? potential contributions of the paper?)*
- 2-3 paragraphs of details *(key ideas? motivation & justification? strengths and weaknesses? technical flaws? supported with results? comparison with other systems? future work? anything you disagree with authors?)*
- 1-2 paragraphs summarizing the discussions in the class.
- Reviews are due two days after the presentation (Wednesday night)
- Recommended Readings:
  - How to Read a Paper, by S. Keshav.
  - Reviewing a Technical Paper, by M. Ernst

Participation

- Post at least one question to the seminar blog by Friday night before the presentation:
- http://cse710.blogspot.com/
- In class participation is required as well
- (Attendance will be taken each class)

Grading

- **Grading will be S/U**

1. If a student fails to attend any class without any prior notification to me with a valid excuse, he/she will loose 1 point.
2. Each student should post at least one question/comment every week to the course blog on one of the papers we discuss that week. Any student failing to do so, will loose 1 point.
3. If a student fails to do a good job in the presentation or in the paper reviews, will loose 1 point.
4. Any student who looses 5 points or more throughout the semester will get a U
5. If a student completely misses a presentation or a review, the student will get a U.

Contact Information

- Prof. Tevfik Kosar
- Office: 338J Davis Hall
- Phone: 645-2323
- Email: tkosar@buffalo.edu
- Web: www.cse.buffalo.edu/~tkosar

- Office hours: Fri 11:30am – 1:00pm
- Course web page: http://www.cse.buffalo.edu/faculty/tkosar/cse710

Any Questions?