Dude, Where's My Data?

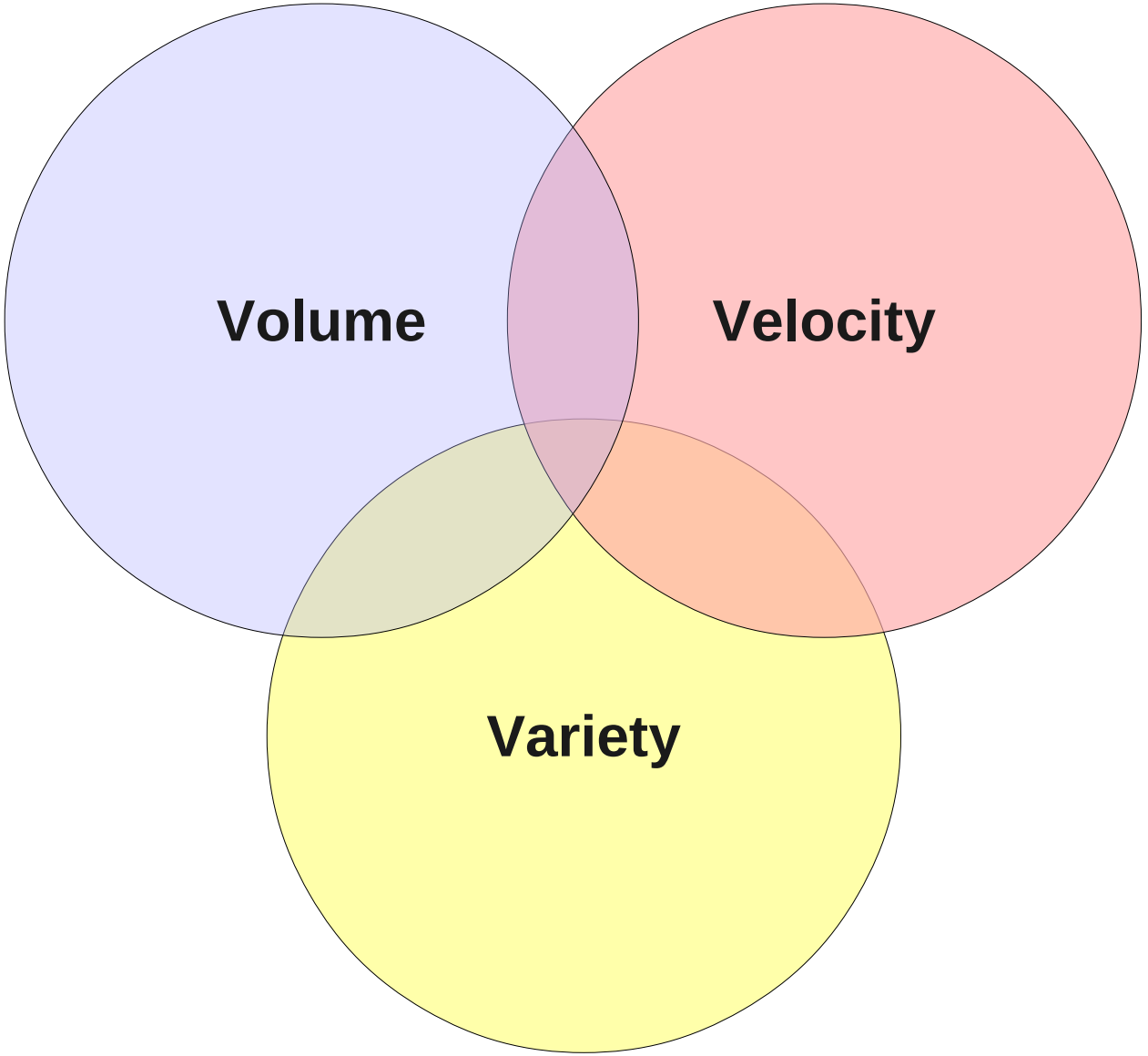
Jeff Darcy
gluster.org

The Problem

- Compute cycles are everywhere
- Your data isn't
- It's easy to move computation, then find that there's no data for it to work on

OOPS!

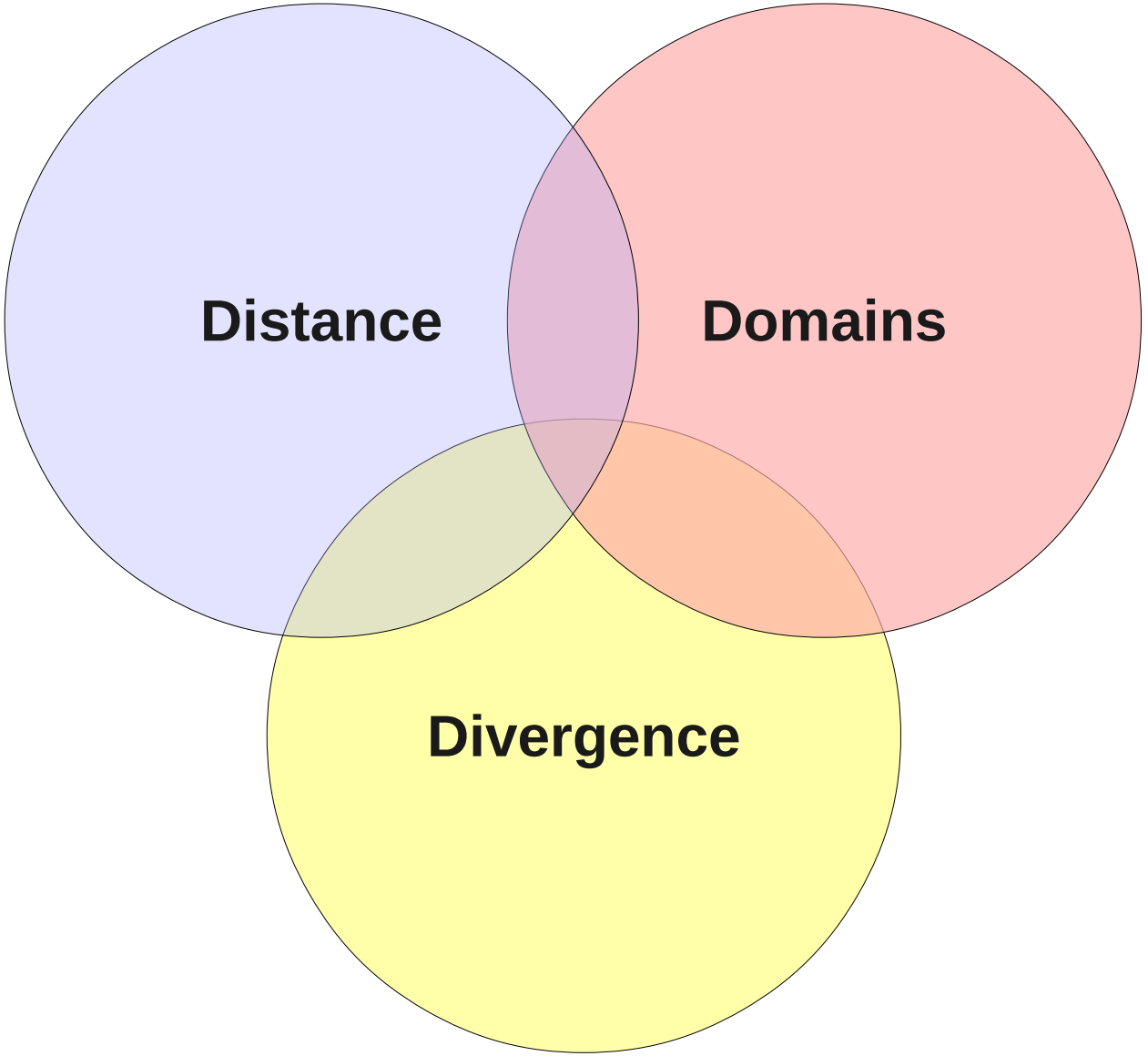
Big Data: V^3



Volume, Velocity, Variety

- Volume = total data (TB)
  - affects initial setup (or full-resync) time
  - bandwidth problem
- Velocity = rate of change (TB/hour, files/hour)
  - affects ongoing bandwidth need
  - bandwidth and latency problem
- Variety = data “shape”
  - file- and directory-size distribution, sparseness, extended attributes, even contents

Bigger Data: D^3



Distance, Domains, Divergence

- Distance: how far?
  - across the river vs. across the world
- Domains: how many?
  - two sites vs. four sites vs. hundreds of sites
  - also separate security perimeters and policies
- Divergence: how similar?
  - sync vs. async, ordered vs. unordered, conflict resolution, ...

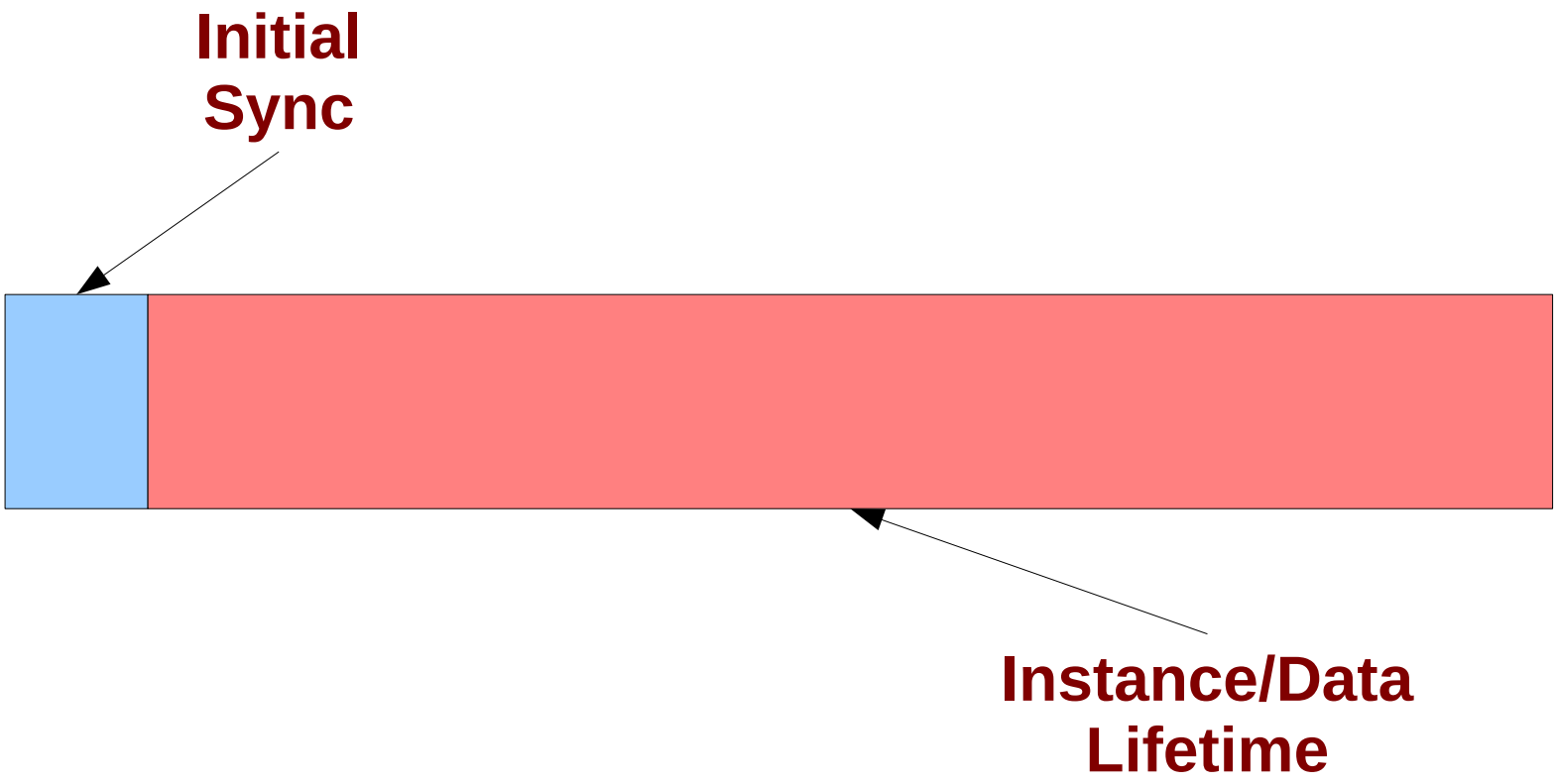
Example: rsync

- Variety affects scanning rate
  - delta comparison favors large files
- Sensitive to distance
  - still need network round trips to compare checksums
- Hard to manage with many domains
  - set up each connection separately, including parallel connections
- High divergence - scanning order, conflicts

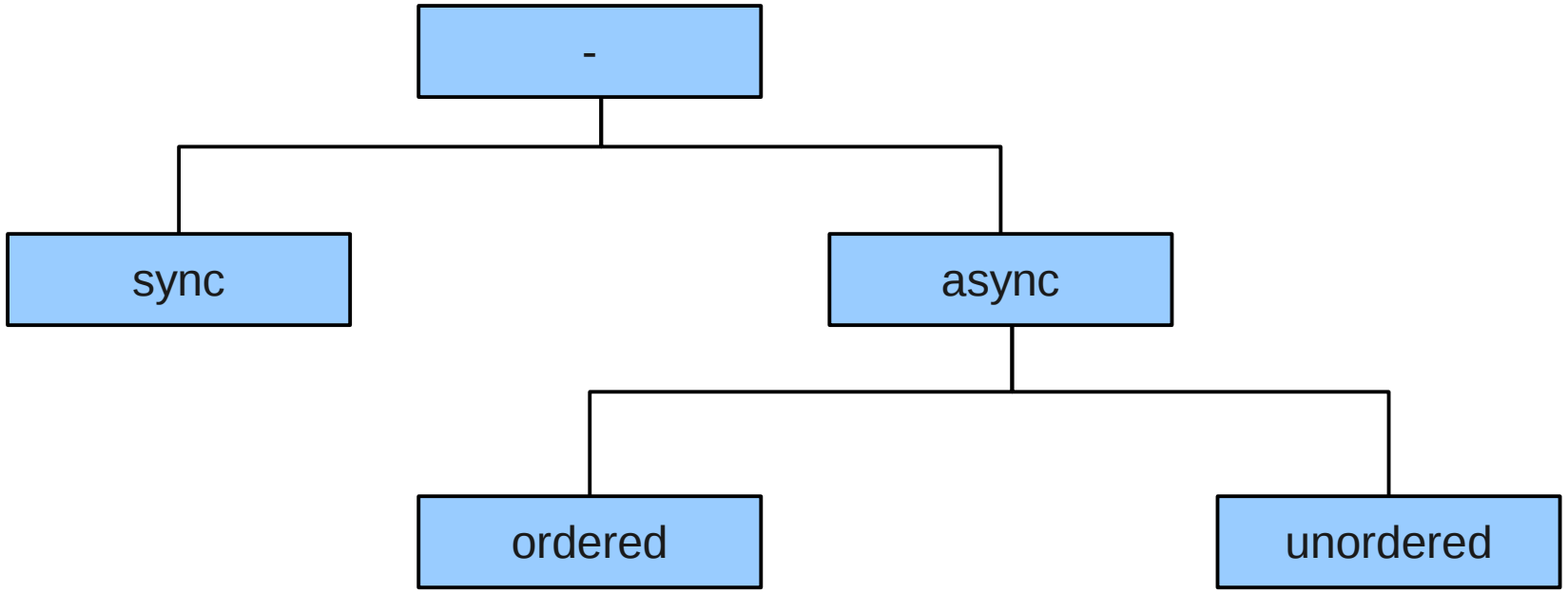
Initial Sync

- Transfer large files instead of small ones
  - 10 MB/s ~= 1 TB/day
  - 80ms RTT ~= 1M ops/day (fewer files/day)
  - copy tarballs or disk images, pack/unpack locally
- Transfer in parallel
  - GridFTP, PFTP, BitTorrent (Murder @ Twitter)
  - Aggressively pre-deploy replicas
    - ...or let a CDN do it for you
- Don't forget compression/deduplication

OK, Now What?



Replication Semantics

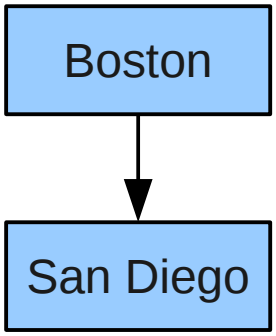


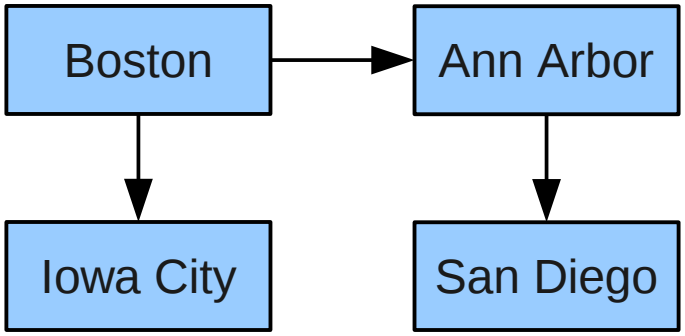
latency sensitive

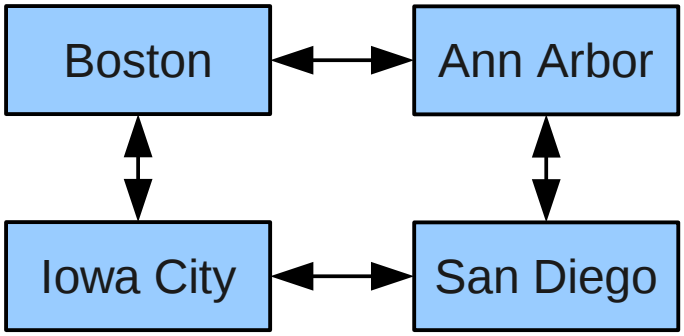
continuous logging

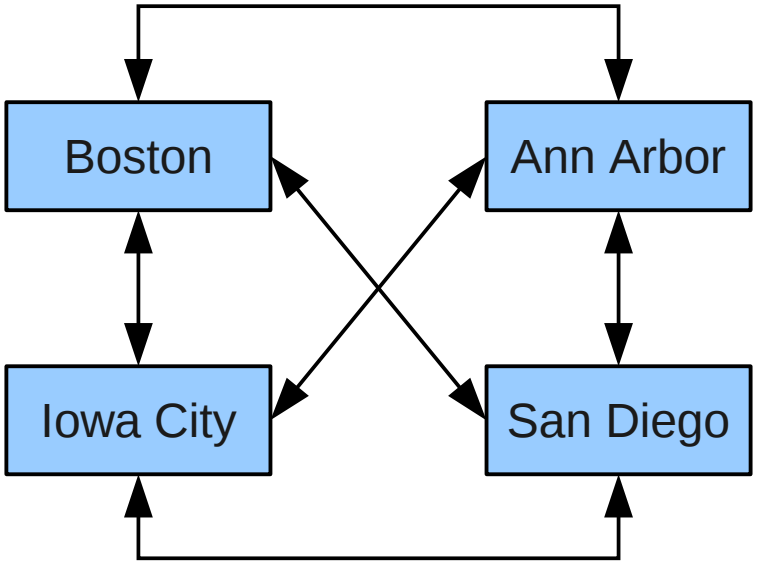
periodic scanning

Replication Topologies









Other Distinctions

- Directionality
  - static master, floating master, peer to peer
- Migration and caching are replication too
  - expressed vs. assumed interest
  - partial
  - expendable (not dependable)

Replication Lite

- Consider using an overlay/union FS
  - Unionfs, AUFS, overlay mounts
- Each client has their own overlay on top of same read-only base
- Ship overlay back home to apply (and resolve conflicts?) at leisure
- Free version history

Sync vs. Async

- Synchronous replication
  - divergence very small
    - still possible with errors
  - performance limited by latency
- Asynchronous replication
  - divergence can be quite large
    - conflict handling becomes most of the code
  - performance limited by bandwidth

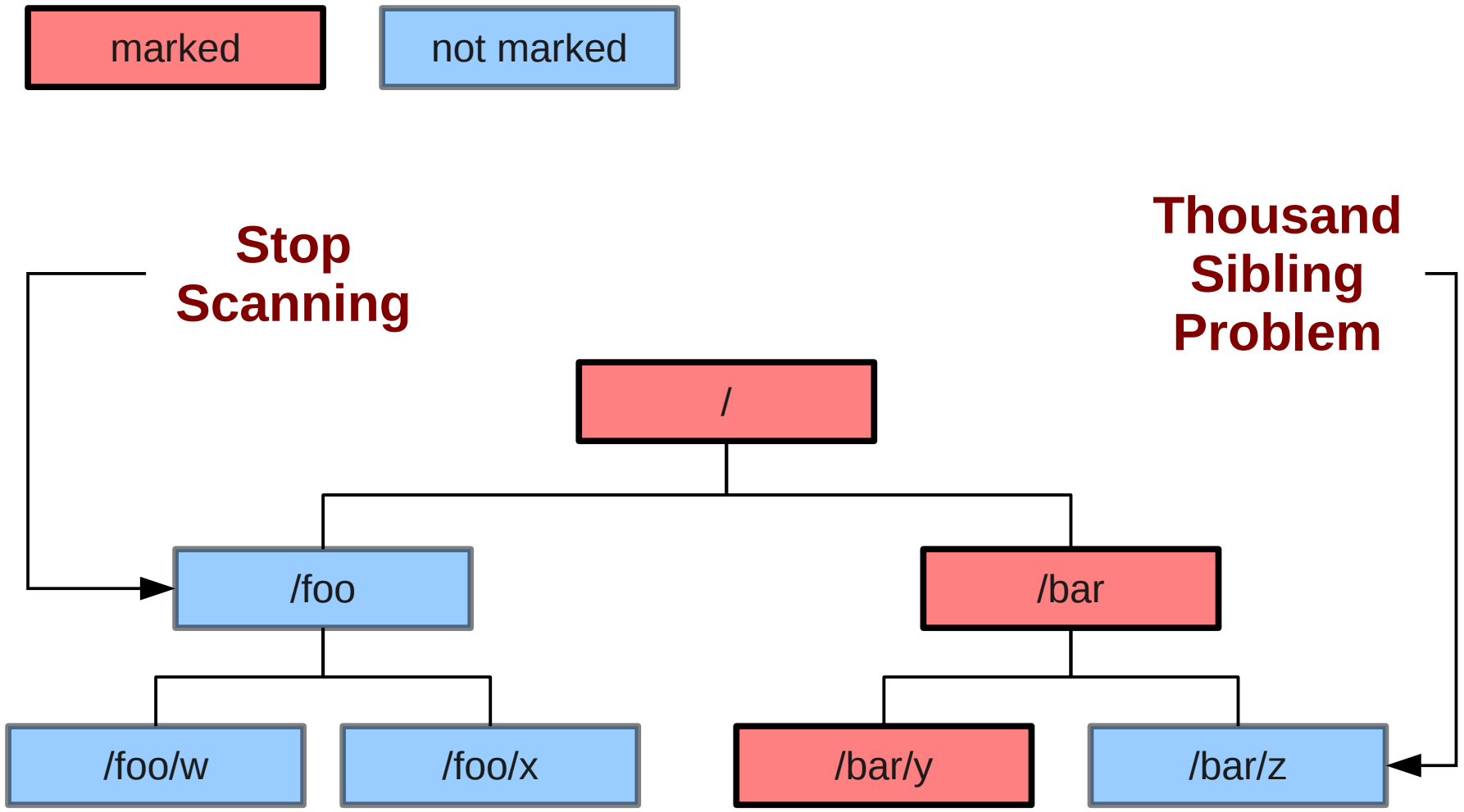
Scanning vs. Logging

- Scanning negatives
  - naive versions are slow and resource intensive
  - even smart versions have high divergence
    - “many siblings” problem
    - often missing info for proper conflict resolution
- Logging negatives
  - requires local buffer space
    - one more thing to provision/manage (or have fail)
  - network interruptions still create divergence

Improving on rsync (1 of 3)

- Wrap a script around it
  - connection setup and credential management
  - parallel streams
  - continuous iteration
- Optimize scanning
  - mark changes up toward root
  - don't scan unchanged subtrees
  - next slide

Improving on rsync (2 of 3)



Improving rsync (3 of 3)

- So it's better than before
  - higher scanning rate
  - more automated
- ...but...
  - scanning is still inherently inefficient
    - still have to find changes within files and/or transfer more than necessary
  - divergence is still high
    - changes appear in scanning order, might conflict

By The Way...

- That's pretty much GlusterFS geo-sync, but I'm not here to talk about that.
- Current project: ordered async replication
  - “pony” replication, as in “all that and...”
  - full duplex mesh, partition tolerant
  - vector-clock conflict resolution
  - maybe I'll be able to talk more about it next year
  - meanwhile, see Resources (last link)

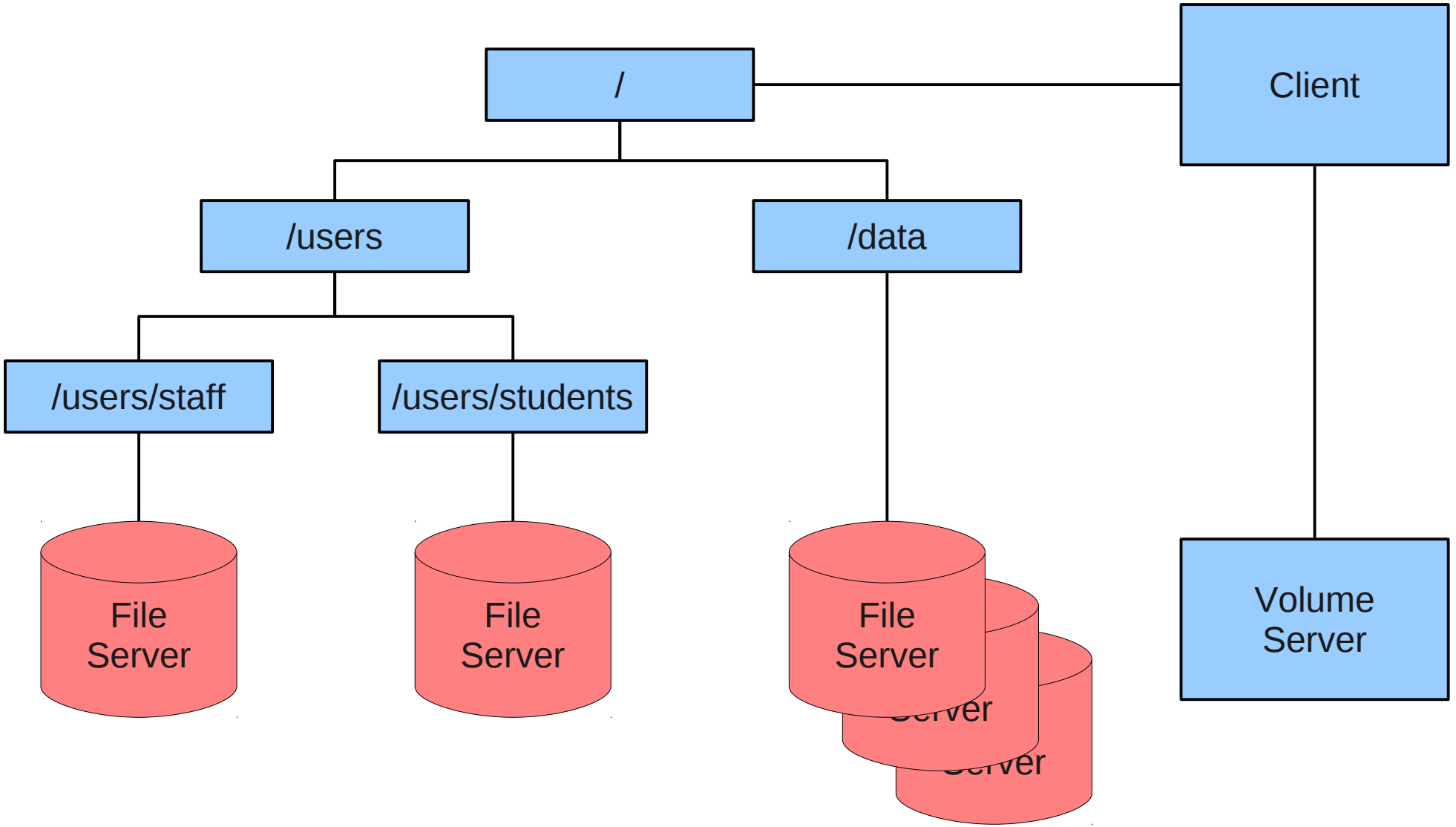
What's Wrong?

- Isn't this all rather . . . manual?
  - you manage scheduling
  - you manage parallelism
  - you manage credentials
  - you manage conflicts
- Yes, it is!
- Let's look at more transparent solutions.

AFS

- The grand-daddy of wide-area distributed filesystems
- Deployed successfully at hundreds of sites, tens of thousands of users
- Only one writable replica, others read-only
- Static file->server assignment
- Notoriously hard to administer and debug
  - seven types of servers, “unique” communication/locking protocols

AFS Diagram


/
Client
/users
/data
/users/staff
/users/students
File Server
File Server
File Server
Volume Server

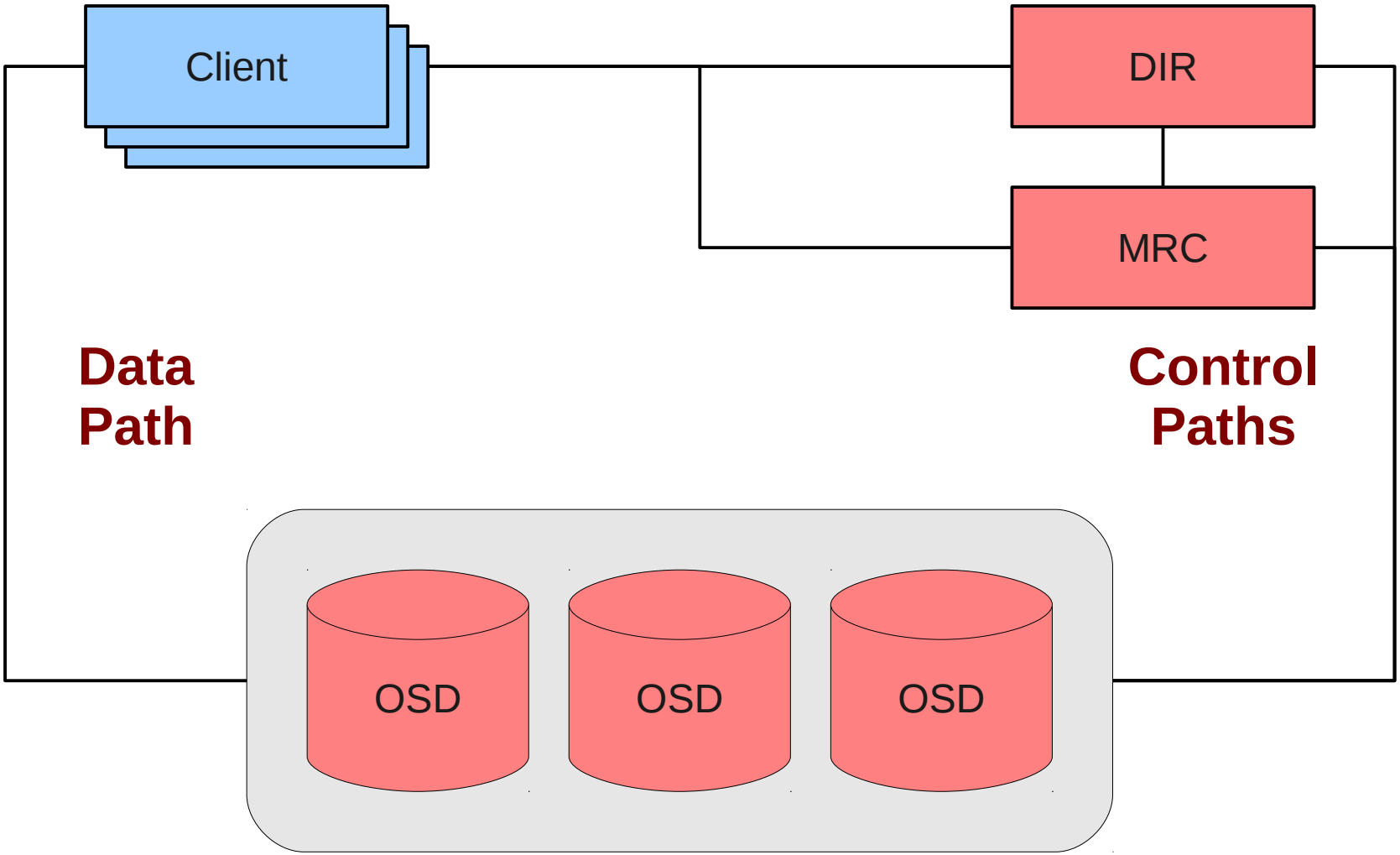
Coda

- AFS descendant
- Adds disconnected-client operation
- Adds multi-way write replication between servers
- Conflict resolution is automatic but type-specific
- Shares other drawbacks with AFS
- Not widely deployed

XtreemFS

- European XtreemOS/Contrail projects
- Servers: one DIR, one MRC, multiple OSD
  - dynamic placement on OSD (better than AFS)
  - DIR/MRC replication/failover still immature?
- Historically: read-only replication (pull model)
- More recently: read/write replication
  - floating master, leases
- Snapshots

XtreemFS Diagram



Other Solutions

- dCache, iRODS: archival orientation, online information almost unreadable
- Sector: paired with Sphere (Hadoop alternative), claims WAN distribution
- DRBD: two-way async block replication
- FS-Cache: client caching add-on to NFS, AFS
- PeerDist/BranchCache: content-addressable caching from SMB/CIFS

Conclusions

- Initial sync is easy, staying in sync is hard
- Conflict resolution is a major issue
  - potential for failure plus performance concern
  - segregate data by consistency requirements
    - including read only
  - try to choose “just enough” consistency
- Some assembly required

Resources

- Saito and Shapiro (essential!) http://www.ysaito.com/survey.pdf
- Academic Background
  - Bayou http://www-users.cs.umn.edu/~he/iss/iss_08122002.ppt
  - Ficus http://www.lasr.cs.ucla.edu/ficus/ficus_summary.html
  - OceanStore http://oceanstore.cs.berkeley.edu/
- Production Code
  - http://rsync.samba.org/
  - http://www.openafs.org/
  - http://www.coda.cs.cmu.edu/
  - http://www.xtreemfs.org/
  - http://www.gluster.org/
- http://hekafs.org/index.php/2011/10/all-that-and-a-pony/