LSST Evaluation of REDDnet and LStore

Evaluating data storage and sharing methods for a coming torrent of astronomy data.

National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

Synopsis

- Broad goals of LSST project and ECSS support.
- Features of REDDnet and Lstore.
- Particulars of the work done during the ECSS support period.
- Future Directions.

External Collaborators

- Alan Tackett, Bobby Brown, Santiago de Ledesma and Mathew Binkley at the Advanced Computing Center for Research and Education (ACCRE) at Vanderbilt University.
- Mike Freemon, Ray Plante, Greg Daus and other members of the LSST project.

LSST Project

- Large Synoptic Survey Telescope
- 3.2 GigaPixel Camera
- Wide field astronomical survey
- Data?

The LSST Data Storage Challenge

- Large Data Volumes (~30TB / night) or raw data + much more processed data
- Long-haul data transfers from South America to primary data facility at NCSA – Illinois and downstream data sharing between collaborators
- Quote from lsst.org:

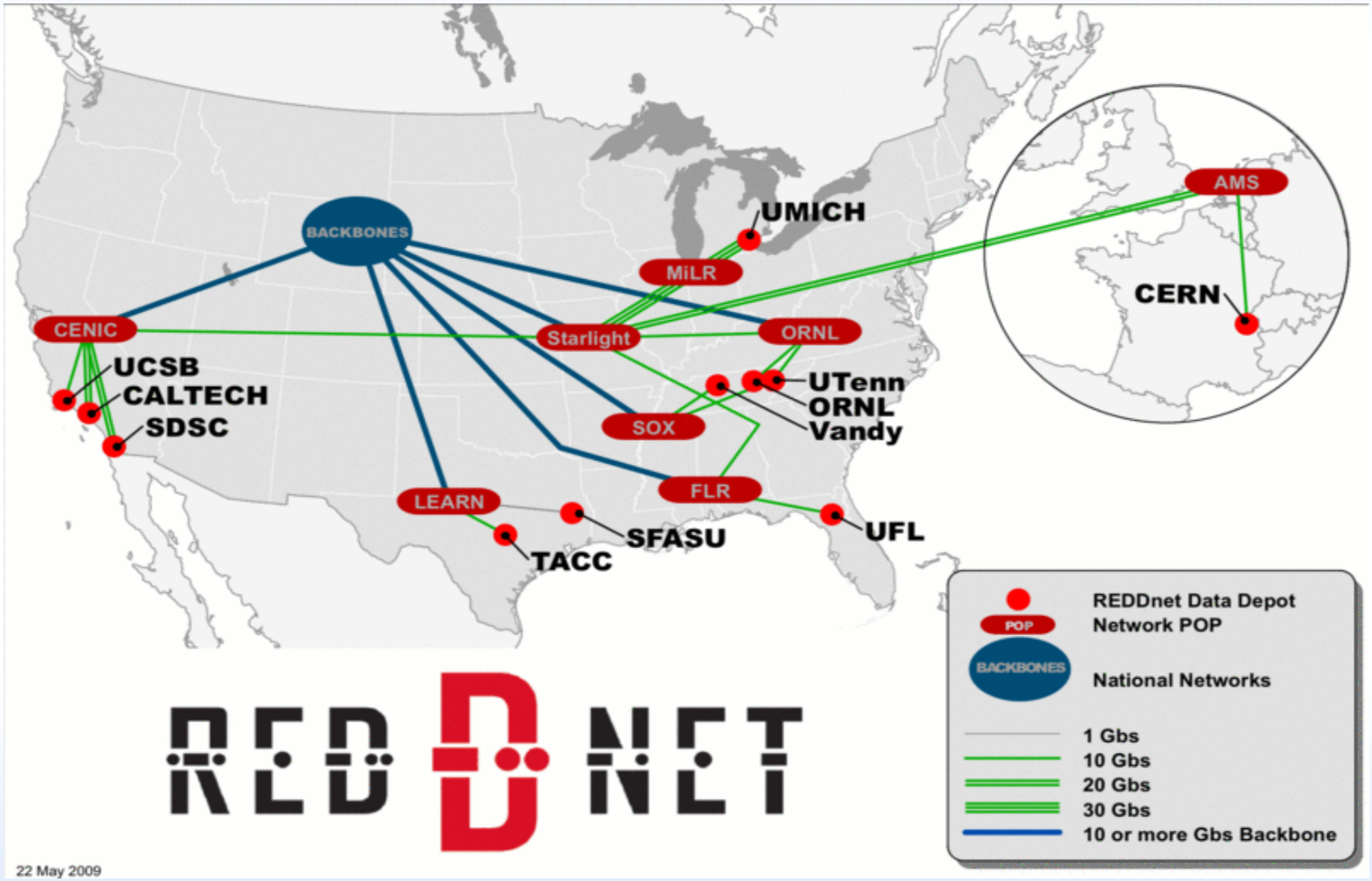
  The image archive produced by the LSST survey and the associated object catalogs that are generated from that data will be made available to the U.S. and Chilean scientific communities with no proprietary period.

- LSST also states the goal of providing open access to their dataset to as many researchers, worldwide as possible.

REDDnet, LSST and ECSS

- The LSST is in the process of evaluating different distributed storage systems including IRODS and REDDnet (Lstore).
- The key requirement here is **distributed storage** with a **global namespace.**
- REDDnet is serving as a testbed for Lstore technology as it comes online.
  - The underlying storage technology of the REDDnet is really Lstore.
  - REDDnet itself is a collaborative cross-institution collection of Lstore IBP Depots.

REDDnet A distributed storage Infrastructure



Under the Hood: LStore Key Features

- Logistical Storage
  - “Bits are bits”
- Built on IBP – Internet Backplane Protocol
- ExNodes – XML Metadata Analogous to the Unix Inode.
- Asynchronous internal architecture.
- Distributed Metadata storage using Apache Cassandra.
- Large data stored in IBP depots
- User defined policies for allocating space, managing replicas etc.

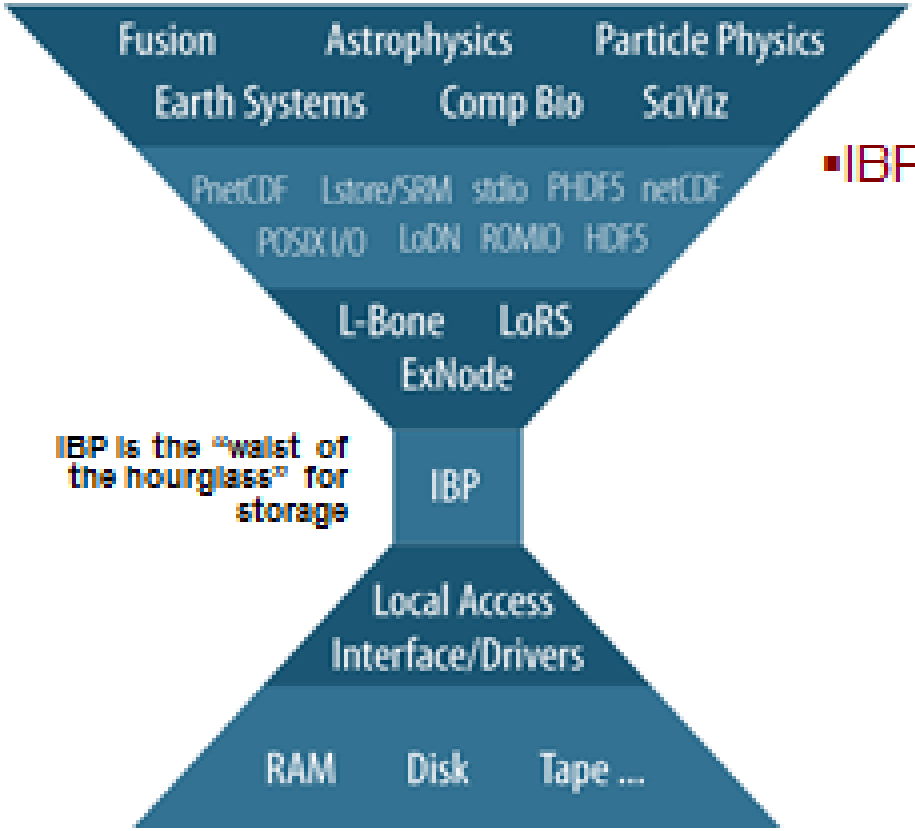
LStore – Logistical Storage

From lstore.org:

- *LOGISTICAL NETWORKING*
  A data transfer protocol is a standard format used to transfer data between computers on a network. L-Store utilizes the Internet Backplane Protocol (**IBP**) developed by the Logistical Computing and Internetworking (**LoCI**) Lab at the University of Tennessee, Knoxville. IBP enables the movement of large data sets via the simultaneous transfer of data fragments rather than requiring the sequential transfer of the entire data set. Mirroring, data striping, fault tolerance, and recovery features are also supported by IBP. The requisite software can be installed on any machine running a Unix/Linux operating system, effectively transforming the machine into a storage depot.

LoCI Tools*

Logistical Computing and Internetworking Lab

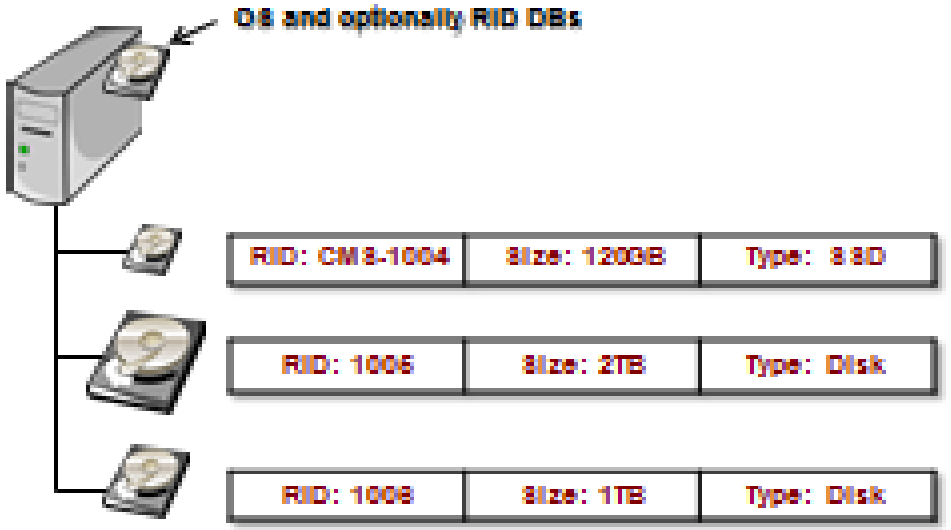


- IBP Internet Backplane Protocol
  - Middleware for managing and using remote storage
  - Allows advanced space and TIME reservation
  - Supports multiple connections/depot
  - User configurable block size
  - Designed to support large scale, distributed systems
  - Provides global "*malloc()*" and "*free()*"
  - End-to-end guarantees
  - Capabilities
    - » Each allocation has separate Read/Write/Manage keys

*http://loci.cs.utk.edu

IBP Server or Depot

- Runs the ibp_server process
- Resource
  - Unique ID
  - Separate data and metadata partitions (or directories)
  - Optionally can import metadata to SSD
- Typically JBOD disk configuration
- Heterogeneous disk sizes and types
- Don't have to use dedicated disks



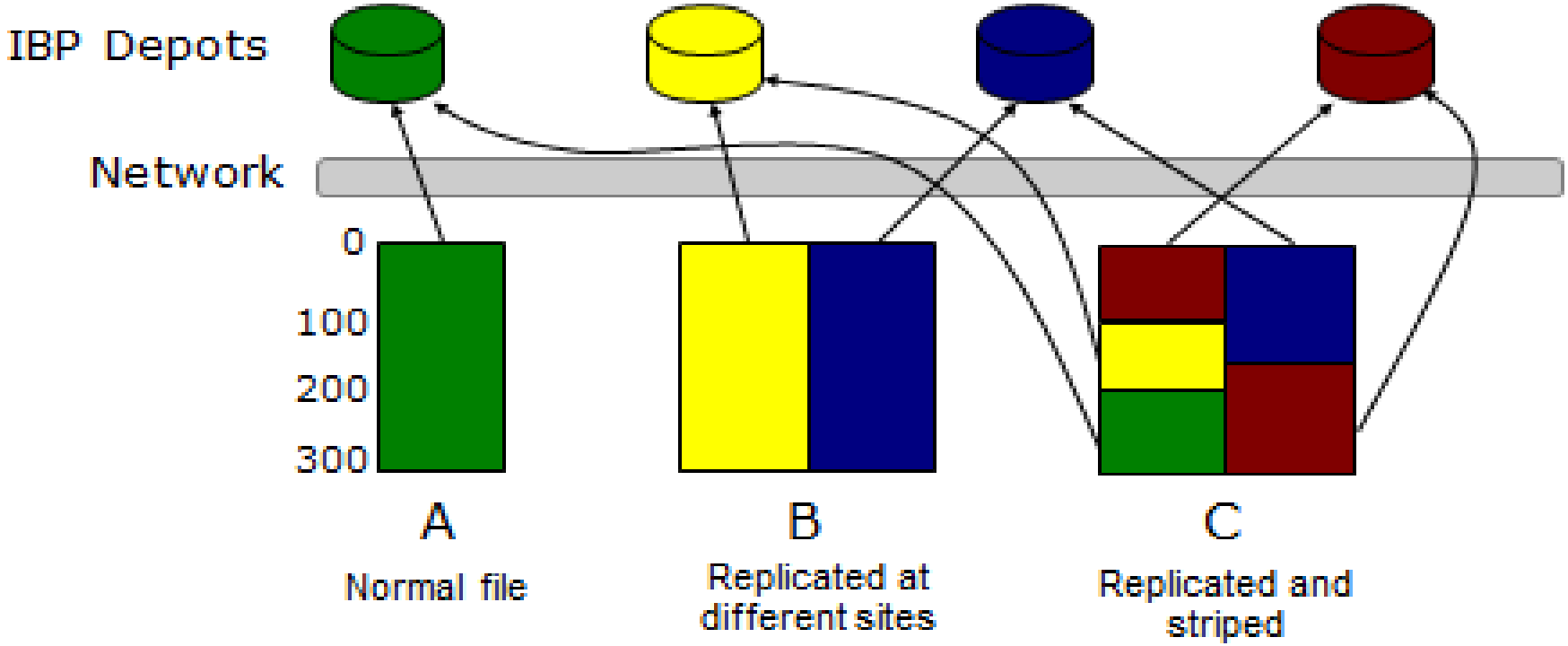
IBP functionality

- Allocate
  - Reserve space for a limited time
  - Can create allocation **Aliases** to control access
  - Enable block level disk checksums to detect silent read errors
- Manage allocations
  - INC/DEC allocation reference count (when 0 allocation is removed)
  - Extend duration
  - Change size
  - Query duration, size, and reference counts
- Read/Write
  - Can use either append mode or use random access
  - Optionally use network checksums for end-to-end guarantees
  - Depot-Depot Transfers
  - Data can be either **pushed** or **pulled** to allow firewall traversal
  - Supports both append and random offsets
  - Supports network checksums for transfers
  - Pheobus support – I2 overlay routing to improve performance
- Depot Status
  - Number of resources and their resource ID's
  - Amount of free space
  - Software version

exNode

XML file containing metadata

- Analogous to a disk I-node and contains
  - Allocations
  - How to assemble file
  - Fault tolerance encoding scheme
  - Encryption keys

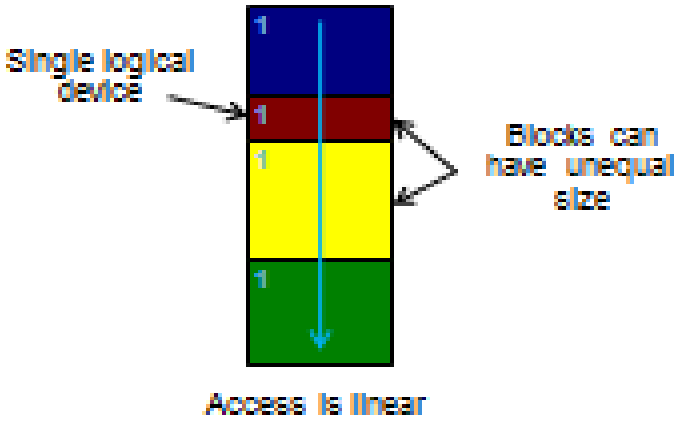


eXnode3

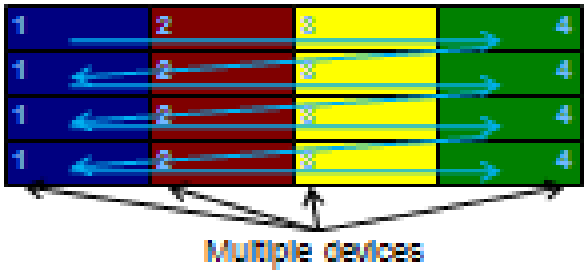
- **Exnode**
  - Collection of layouts
  - Different layouts can be used for versioning, replication, optimized access (row vs. column), etc.
- **Layout**
  - Simple mappings for assembling the file
  - Layout offset, segment offset, length, segment
- **Segment**
  - Collection of blocks with a predefined structure.
  - Type: Linear, striped, shifted stripe, RAID5, Generalized Reed-Solomon
- **Block**
  - Simplest example is a full allocation
  - Can also be an allocation fragment

Segment types

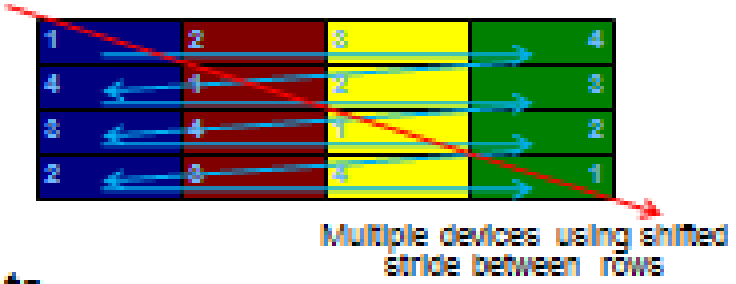
Linear



Striped



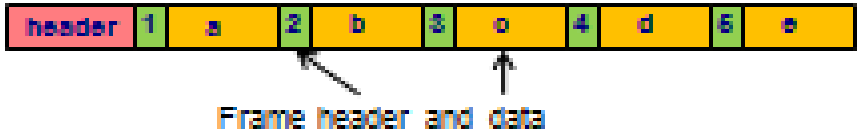
Shifted Stripe



RAID5 and Reed-Solomon segments are variations of the Shifted Stripe

Optimize data layout for WAN performance

Skip to a specific video frame

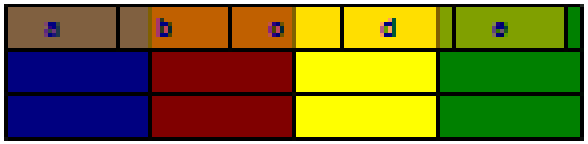


- Have to read the header and each frame header
- WAN latency and disk seeks kill performance

Optimized layout

- Separate header and data into separate segments
- Layout contains mappings to present the traditional file view
- All header data is contiguous minimizing disk seeks.
- Easy to add additional frames

Header and Frame Information

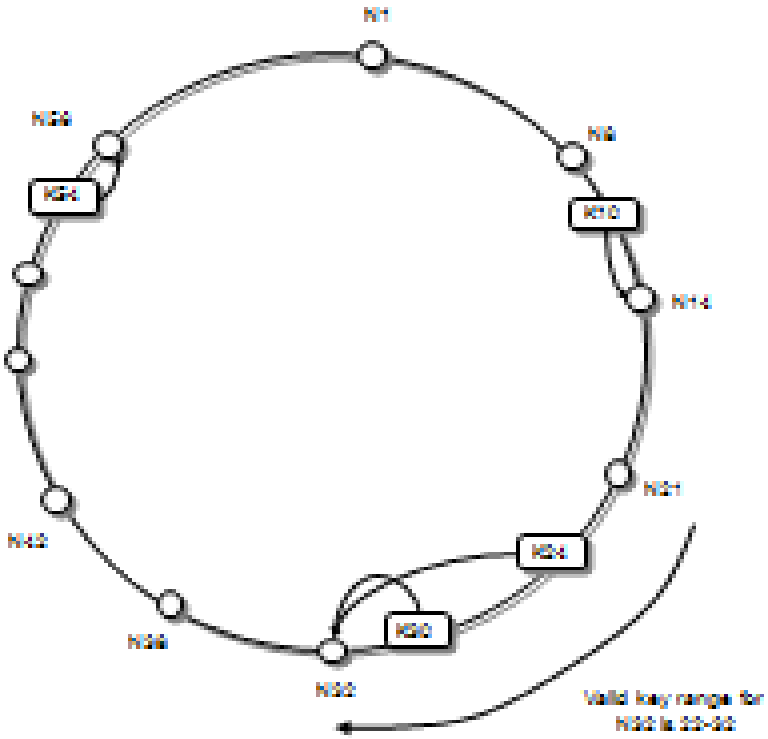


Frame data

Distributed Metadata: Apache Cassandra

- A *NoSQL* database
- Symmetric design
  - No single point of failure (like a Mom node or special root process)
  - Provides linear scaling as nodes are added.
- Distributed Hash Table Lookups
- Tunable Consistency

Chord Ring*
Distributed Hash Table



- Distributed Hash Table
  - Maps a Key (K##) - hash(name)
  - Nodes (N##) are distributed around the ring and are responsible for the keys "behind" them.
  - O(lg N) lookup

*F. Dabek, et al., "Building Peer-to-Peer Systems With Chord, a Distributed Lookup Service." In Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), May, 2001.

LStore Key Features Recap

- Logistical Storage
  - “Bits are bits”
- Built on IBP – Internet Backplane Protocol
- ExNodes – XML Metadata Analogous to the Unix Inode.
- Asynchronous internal architecture.
- Distributed Metadata storage using Apache Cassandra.
- Large data stored in IBP depots
- User defined policies for allocating space, managing replicas etc.

ECSS Role

- Integration of current LSST Archival storage (NCSA MSS tape archive) and LSST REDDnet depots
  - Coordinated with ACCRE team to develop specific features for MSS integration
  - LStore preforms a READ only pull of tarballs from MSS into the global namespace of all (untarred) files
  - A request for a single file living in a MSS tarballl that is not staged will result in a staging request and all of the data file in that tarball will end up on the LSST LStore depot.

Future work

- A FUSE mount for Linux and OSX is currently being tested.
- Explore the use of LStore policies to manage the data depot network
  - Implement efficient caching policies
  - Stage specific data products to geographic locals that request.
  - Bring derived data from various research teams online.

Speculation Slide: A Research “DropBox”

- Commodity buy-in of storage depots for local research teams to meet LSST data sharing goals.
  - DropBox provides a seemless cross-platform online storage interface (oh, and by the way you also get automatic version control). With LStore clients that are similar to those deployed by DropBox, researchers could have local file system mounts for interactive data exploration on their laptops, phones -- whatever.
  - The same infrastructure could be used to provide a local collaborative data sharing space and to even push data back to the larger LSST research community

Summary

- LStore and REDDnet have been deployed on some interesting new technologies (IBP, ExNode, Cassandra)
- Not as mature of a project (eg. Code released documentation etc.) as say, IRODS.
- Lstore leverages the managed REDDnet infrastructure as a benefit (less administrative overhead) to institutions that are interested in contributing resources.
- New features have possibility of satisfying data sharing goals of many Collaborative research projects including LSST.

Links

- http://www.reddnet.org
- http://www.lstore.org
- http://loci.cs.utk.edu/ibp/index.php
- http://cassandra.apache.org/
- http://www.dropbox.com
- http://sparkleshare.org