Storage Challenges for LSST: When Science Is Bigger Than Your Hardware

Jeff Kantor
Project Manager, LSST DM

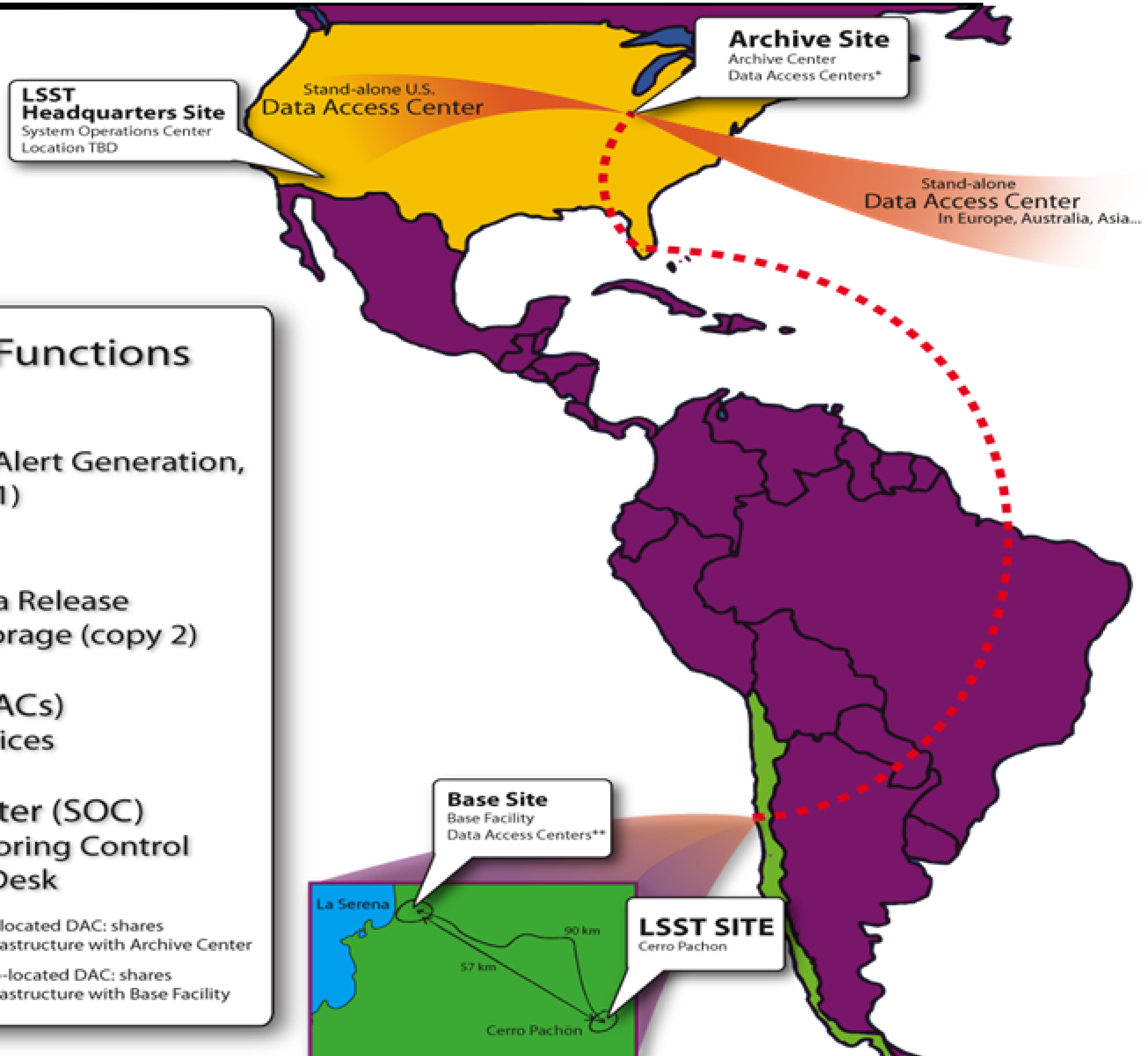
Arun Jagatheesan
San Diego Supercomputer Center &
iRODS.org / DiceResearch.org

26th IEEE (MSST2010) Symposium on Massive Storage Systems & Technologies

May 3-7, 2010

Lake Tahoe, NV

LSST sites



Site Roles and their Functions

- Base Facility
  Real-time Processing and Alert Generation, Long-term storage (copy 1)
- Archive Center
  Nightly Reprocessing, Data Release Processing, Long-term Storage (copy 2)
- Data Access Centers (DACs)
  Data Access and User Services
- System Operations Center (SOC)
  System Supervisory Monitoring Control & End User Support/Help Desk

* Co-located DAC: shares infrastructure with Archive Center

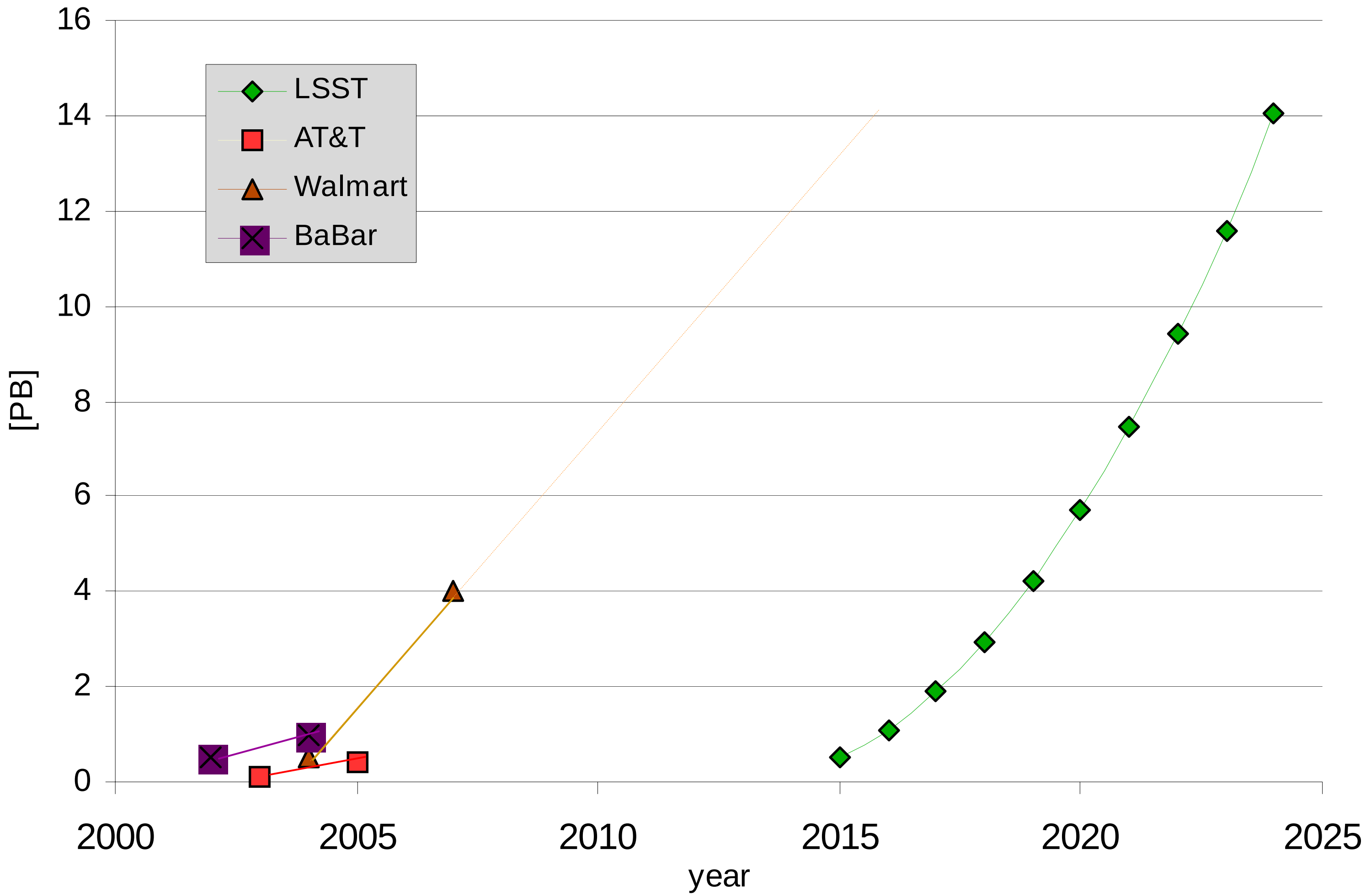
** Co-located DAC: shares infrastructure with Base Facility

Data Sizing: Quick Glance

- **One 6.4 GB image every 17 seconds**
- **15 TB per night for 10 years**
- **45 TB of intermediate results (Calibrated images, etc.)**
  - **Needed for pipeline processing**
  - **Not saved; Recreated from provenance as needed**
- **100 PB final image archive**
- **14 PB final database (data + indexes) (single site)**
  - **Largest table: 3 trillion rows**
- **~100K events per night for 10 years**

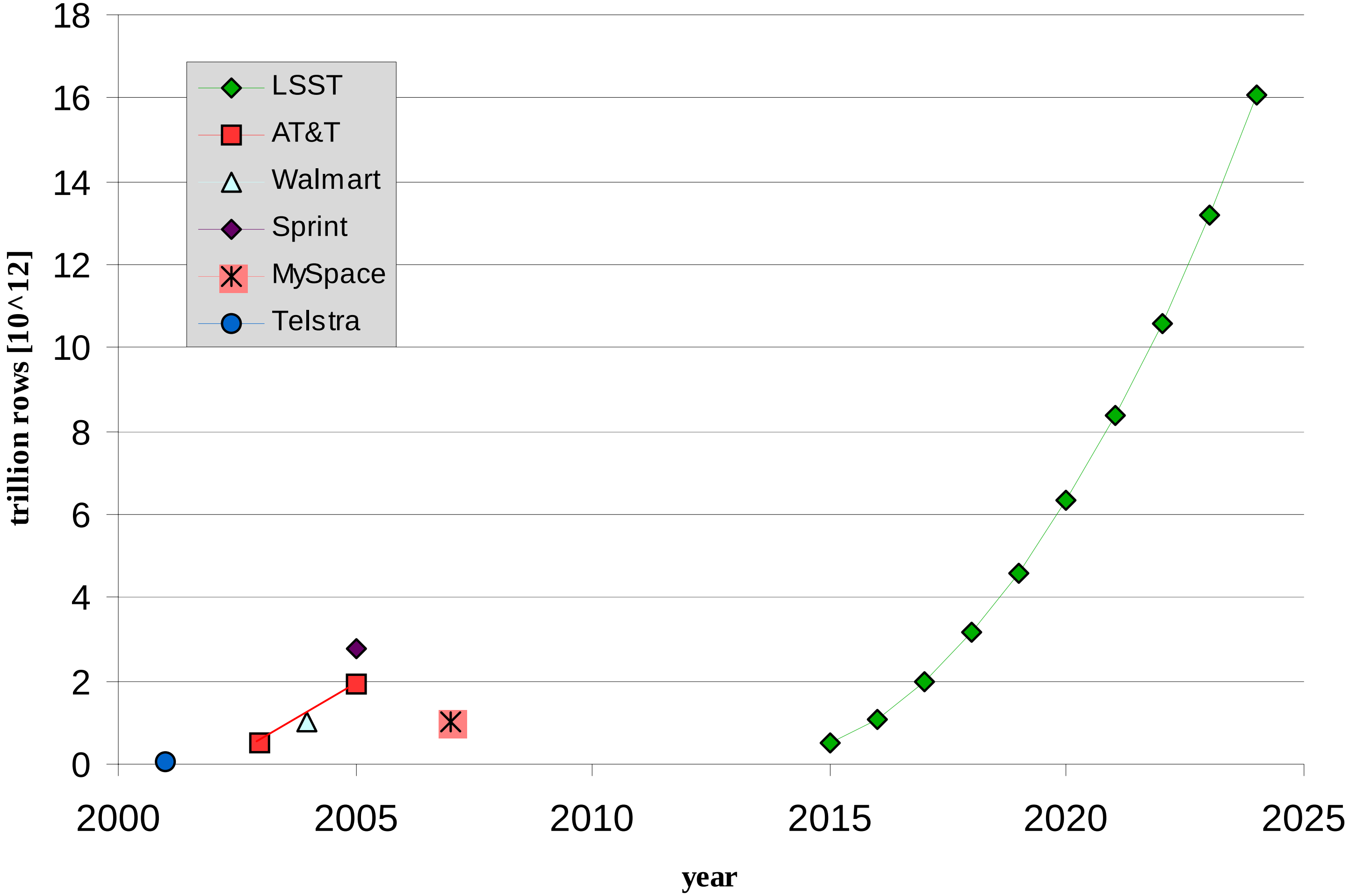
	Archive Site	*Base Site*
Compute (TF)	120 grows to 333	55 grows to 61
Disk for Images (PB)	13 grows to 31	7 grows to 10
Disk for RDBMS (PB)	1 grows to 14	1 grows to 14
Tape (PB)	24 grows to 91	24 grows to 91

Large RDBMS Systems - Data Volumes



* *All numbers based on publicly available data*

Large RDBMS Systems - Number of Rows



* *All numbers based on publicly available data*

We're not Google: the economies of science

- **LSST is a cost-limited project**
  - **Storage is the most expensive hardware component**
  - **Greater investment in storage means less investment in other areas (computing capacity, user resources, etc.)**
- **LSST is I/O intensive**
  - **We will reprocess the entire archive yearly**
  - **Typical science analysis will rely heavily on the catalog products**
    - Correlation studies will often require full table scans
    - Population studies will leverage massive indexes
    - Typical database access will require high-bandwidth to stored tables and indexes
- **LSST has strong through-put requirements**
  - **Nightly observations must be processed in real-time**
  - **Through-put must be sufficient to meet year data releases**
  - **Reliability is important for throughput**

Challenges to the ideal architecture

- **Where is the “sweet spot” that balances cost, throughput, reliability, and ease of access by the community?**
- **Tracking/predicting hardware and data center trends**
  - **How do we optimize cost-performance**
  - **How do these affect long-term preservation?**
- **Managing a hierarchical storage architecture**
- **Managing data across the LSST data sites**
- **Meeting performance requirements for user database searches**

Hardware and Data Center Trends

Preservation Medium: disk versus tape

- **Both disk and tape continue to improve steadily in capacity and cost/TB**
- **Cost/TB trends show tape remaining substantially ahead of disk for the foreseeable future**
- **Will the cost curves ever cross in the next 15 years?**
  - **Are there other costs to factor in (e.g. cooling, licensing)**
  - **"MAID" technologies: dynamic spin up/down for reduced wear and operating costs (still not widely used)**
  - **Solid state for very low latency applications**

LSST Solution

- **Long term storage combining tape and (cheap) disk cache**
  - **Have option of varying proportion of tape and disk over time**
  - **Can migrate to disk if economically expedient**
- **Cheaper tape allows us to invest more in database performance**

Hierarchical Storage Architectures

- **Spinning all of our data is not possible**
- **Hierarchical storage architecture**
  - **Addresses how we use the limited spinning disk with a full archive on tape**
  - **Different levels of storage (in terms of $/TB) for different purposes**
    - The numbers that characterize performance will change over time, but cost class will remain roughly the same for each level.
- **Three levels:**
  - **High performance: high bandwidth disk or disk + SSDs (see Szalay's work)**
    - Attached to pipeline compute platforms for HP I/O with emphasis on performance and capacity
    - Database storage with emphasis on number of spindles for hi bandwidth
  - **Intermediate storage: medium performance for lower cost**
    - Most run as a cache of the most recently produced or used data
  - **Long-term storage: “slow”, cheap disk + tape library**
    - Disk is front end cache to mass storage
    - Performance boosted by increasing spindles

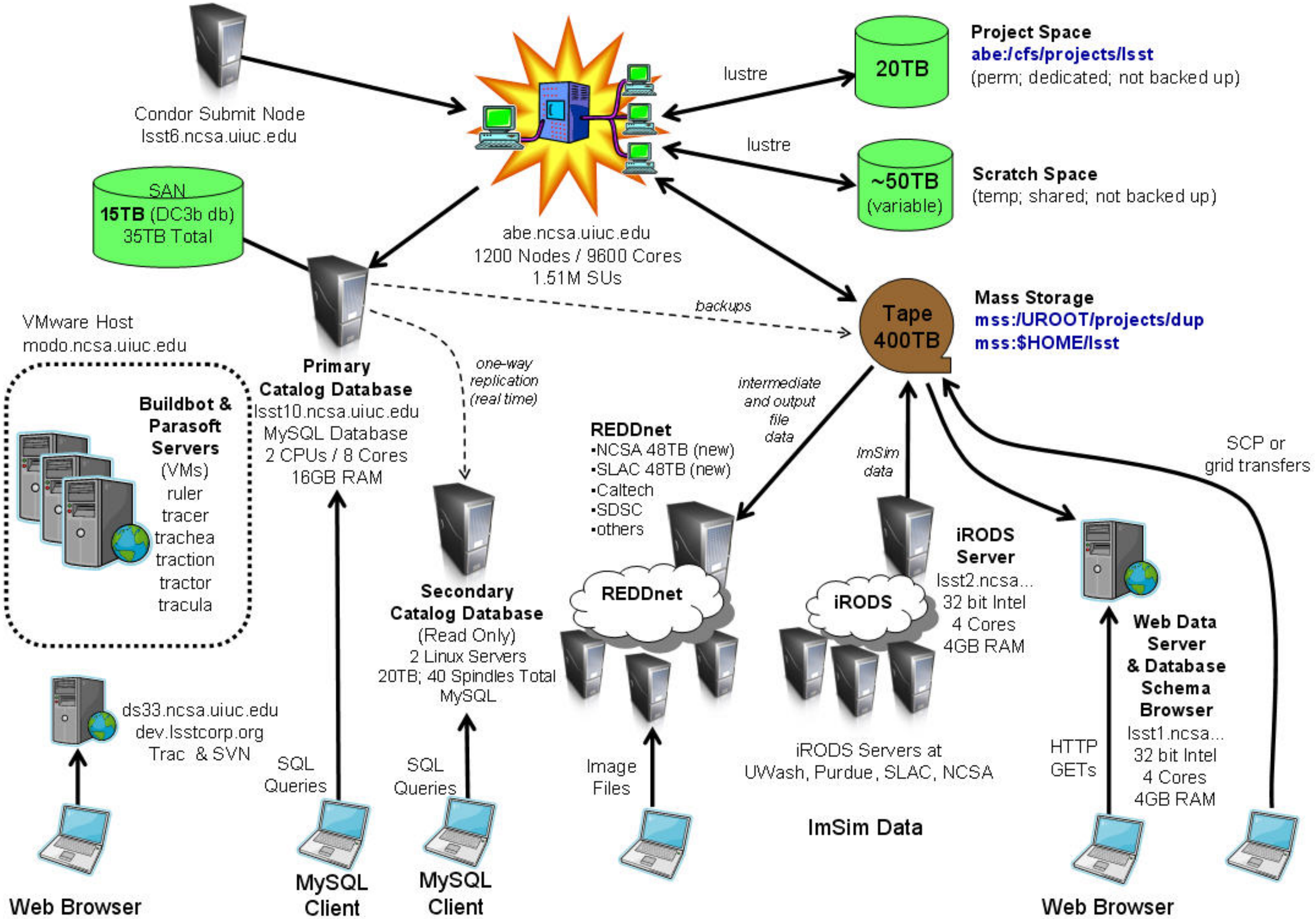
Managing Hierarchical Storage

- **Caching strategies become important**
  - **When reprocessing the archive, we must orchestrate the migration of data between disk and tape**
    - Ideally, like a rolling buffer that can keep up with data processing
    - Can we organize the processing so as to only transfer once?
  - **Pipelines execute assuming all data they need are on disk**
    - Caching ahead is important
    - Constrains the minimum amount of disk needed for caching

High performance access for database and files

- **Optimized for database access**
  - **Performance analysis:**
    - Analysis of user queries -> required memory and bandwidth -> per disk bandwidth, number of spindles
    - Emphasizing a “balanced” system according to Amdahl's law (Graywulf)
    - Capacity exceed data volume by factor of 2-3 (room for second copy).
  - **MySQL scaling tests**
  - **Cost effective performance**
    - SSD systems and USNOB db
- **Optimized for parallel file access**
  - **Server aggregation as a means of improving I/O bandwidth**

Data Challenge 3b Architecture



MSST2010
May 3-7, 2010
Lake Tahoe, NV

LSST sites may grow beyond Americas…

Separate file systems…

\\i\exp\file1.fits
\\i\exp\file3.fits

/usa/exp/file1.fits
/usa/exp/file2.fits

/euro/exp/file2.fits

/chile/exp/file1.fits

Disadvantages of separate file systems

- Increased cost of operation
  - Storage cost for data backups (in petabytes)
- No load sharing
- No inter-site failover
- Need for scripts at each site to coordinate with each other while mirroring the data
- Lots of telecons, emails and frustrated sys-admins

\+ Autonomous operation of data centers within each funding agency (or country) to satisfy their tax payer’s dollars / euros / …

CDLM @ LSST

- **Collaborative Data-lifecycle Management**
  - **Data by itself is a process**
  - **Data has to be social and "collaborate" with producer(s), consumer(s), and storage provider(s)**
- **CDLM @ LSST**
  - **Files and collections are the primary data types**
  - **Multi-continental data centers in (North America, South America and Europe)**
  - **Multiple storage/file systems (NFS, UniTree Mass Storage System, HPSS Mass Storage System, HFS+/HFSX, Lustre)**
  - **Multiple user groups and access permissions**
- **Plug-n-play**
  - **Add or remove: Data centers, Inter-continental collaborations, storage resources and data sets**

LSST CDLM Requirements - 1
(LSST-DLM Req doc)

Large Synoptic Survey Telescope

- **Data Access Services (DAS)**
  - ***Access Transparency*** **irrespective of geological location of data, client, storage protocol, hardware etc.,**
  - ***Automated Replica and Storage Selection*** **to optimally use the right data, storage location based on heuristics**
  - ***Query and discovery of files***
  - ***IVOA standards and Public interfaces***
  - ***Virtual data on demand*** **- convert an image access request into a request to create images on demand and deliver them**

LSST CDLM Requirements - 2 (LSST-DLM Req doc)

- **Data Distribution Services (DDS)**
  - Replicate data X in Y hours (or Move/copy/transfer data)
  - Support multiple protocols (TCP, non-TCP)
  - Application-driven multi-point data transfer scheduling
- **Some Others**
  - RBAC (Role Based Access Control)
  - Support a major site failure and recovery without disrupting operations
  - Evolve along with data storage evolution
  - Allow external storage to be plugged into LSST DLM (plug-n-play of data centers)

iRODS – possible solution to CDLM in LSST

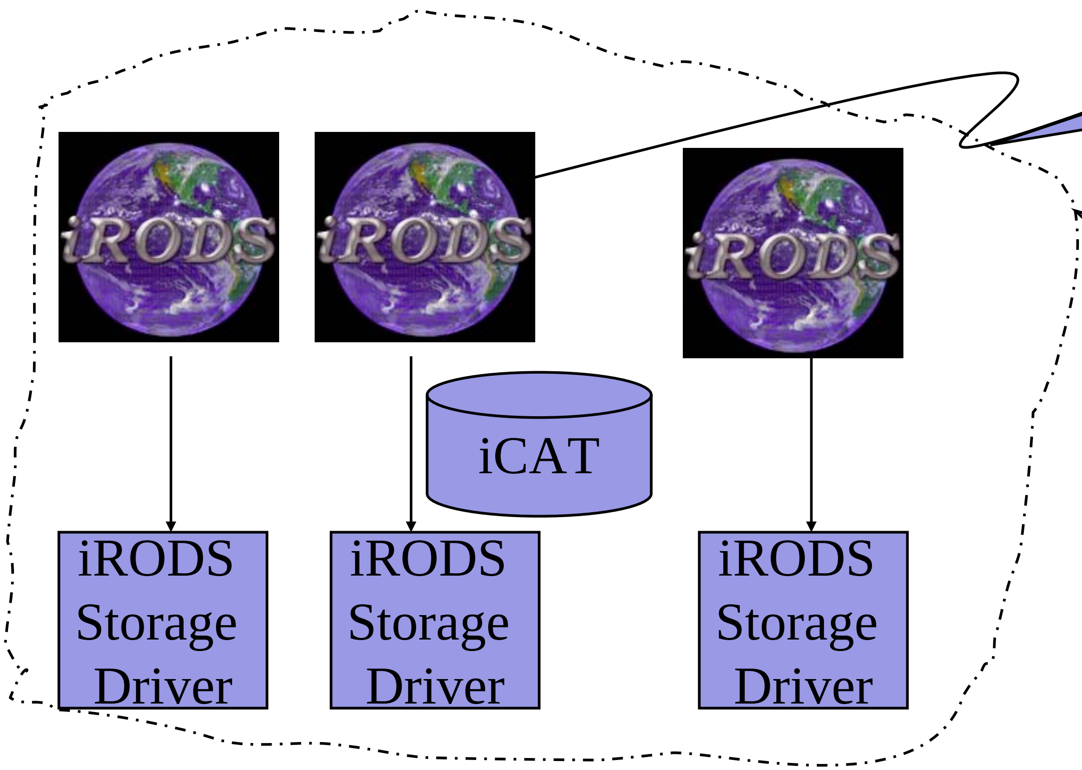
- **iRODS**
  - Integrated Rule-Oriented Data System
  - (Data Grid Management System)
- **Logical data storage namespace**
  - Logical directory structure with files, replicas and collections from multiple locations
- **Rules and Microservices**
  - Management of data using policies or simple ECA rules.

  - [More www.irods.org]

Peer-2-peer (like) iRODS servers

Client can connect to any distributed iRODS server

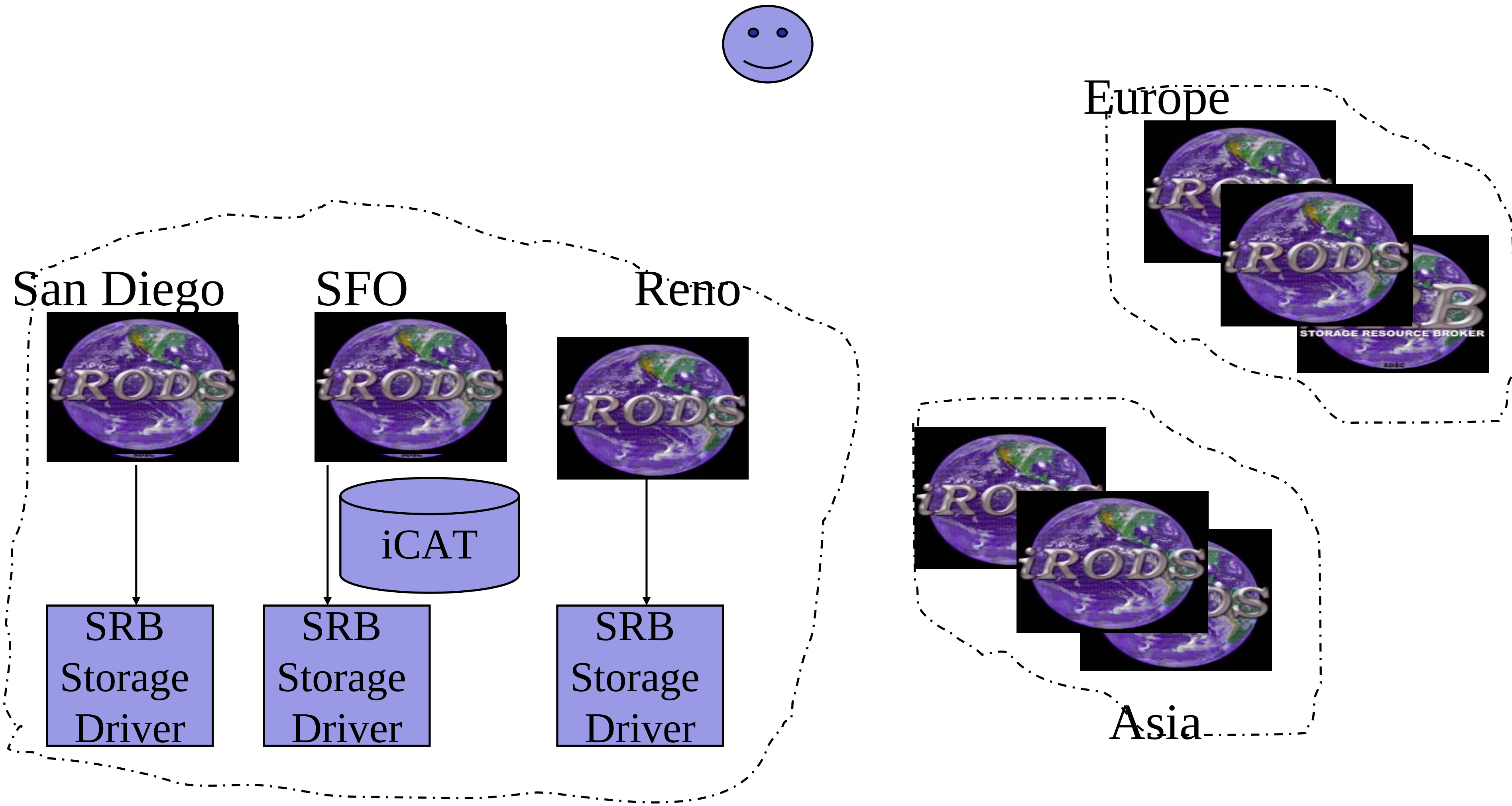
iCAT-Enabled Server



An iRODS zone

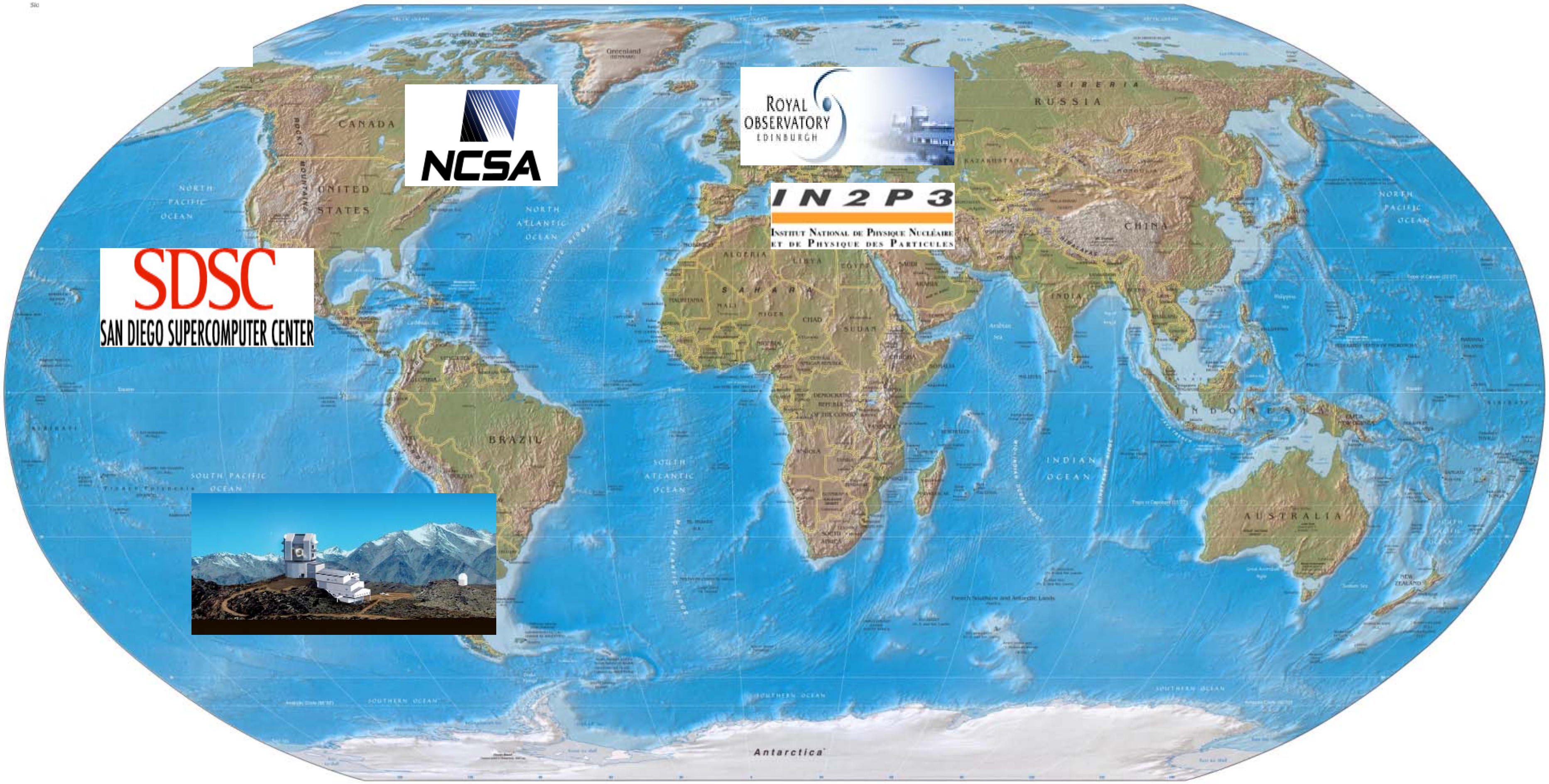
The role of iCAT and lack of leader election protocol does not make the servers fully P2P

Peer-2-peer iRODS Zones



Finalist HPC Storage Challenge - SC 08

Large Synoptic Survey Telescope



Simulation of the pipeline (SC08)

- Pipeline processing of images
  - Data from telescope (IN2P3 France) ingested into iRODS resource
  - Images automatically replicated into Base at UK (iRules)
  - ImageSubtract Pipeline process started by iRODS software itself at Base (UK) after each Image exposure is replicated from France
  - Data again replicated to NCSA - Archival center
  - More detailed ImageSubtract pipeline at NCSA for the same images
- Data-lifecycle in Action
  - Rules or policies managing data pipelines, replication
  - LSST files have the same file name everywhere on this single confluence of systems spanning HPC, data delivery, archives

  - // This slide can be skipped

Performance & Scalability (GREEN TESTS)

- MAX number of files
  - 9.2 quintillion (billion times billion)
  - LSST will have to have an ingest rate of little more than 30 billion files/second to reach MAX count in our infrastructure software
- MAX File Size for one file (NOT TESTED)
  - 1 Exabyte (if you have a file system that can store it and bandwidth to transfer it)
- MAX File System size for WHOLE system (NOT TESTED)
  - 9.2 undecillion bytes (10^{36})
  - Considering replicas also it will be just over one hundredth of quindecillion bytes (10^{47}) bytes (way smaller than a googol)
- MAX number of files in a directory (collection)
  - 9.2 quintillion

The QUINTILLION MARK (GREEN WAY)

srbbrick15:/data1/LSST-SC08/V4-stressTest/iRODS % iput -R quintillion+ Makefile countown6

srbbrick15:/data1/LSST-SC08/V4-stressTest/iRODS % iput -R quintillion+ Makefile countown5

srbbrick15:/data1/LSST-SC08/V4-stressTest/iRODS % iput -R quintillion+ Makefile countown4

srbbrick15:/data1/LSST-SC08/V4-stressTest/iRODS % iput -R quintillion+ Makefile countown3

srbbrick15:/data1/LSST-SC08/V4-stressTest/iRODS % iput -R quintillion+ Makefile countown2

srbbrick15:/data1/LSST-SC08/V4-stressTest/iRODS % iput -R quintillion+ Makefile countown1

srbbrick15:/data1/LSST-SC08/V4-stressTest/iRODS % iput -R quintillion+ Makefile countown0

ERROR: putUtil: put error for /LSSTzone/home/rods/quintillion/countown0, status = -806000 status = -806000 CAT_SQL_ERR

Research and Near Future Issues

- Disconnected Data Centers within a Data Grid
  - What if Chile is not connected to US or Europe for a week due to some link problem how does the CDLM (or) iRODS handle it?
- Dynamic cost based storage placement
  - Currently we use fixed heuristics that require manual one-time update (which is usually ok in most scenarios)
- Multipoint data distribution plans
  - How to distribute data from Site-A to sites L,M,N,O,P ?
- Can Europe grab data from US data centers?
  - How to incorporate acceptable inter-zone transfers and priority users
- NVM (Non Volatile Memory) and iRODS
  - Optimal way to use SSDs or PCM for LSST and iRODS

Acknowledgements

- Ray Plante and Mike Freemon, NCSA
- Jacek Becla, SLAC
- Members of LSST DM from multiple institutions

Storage Challenges for LSST: When Science Is Bigger Than Your Hardware

Jeff Kantor
Project Manager, LSST DM

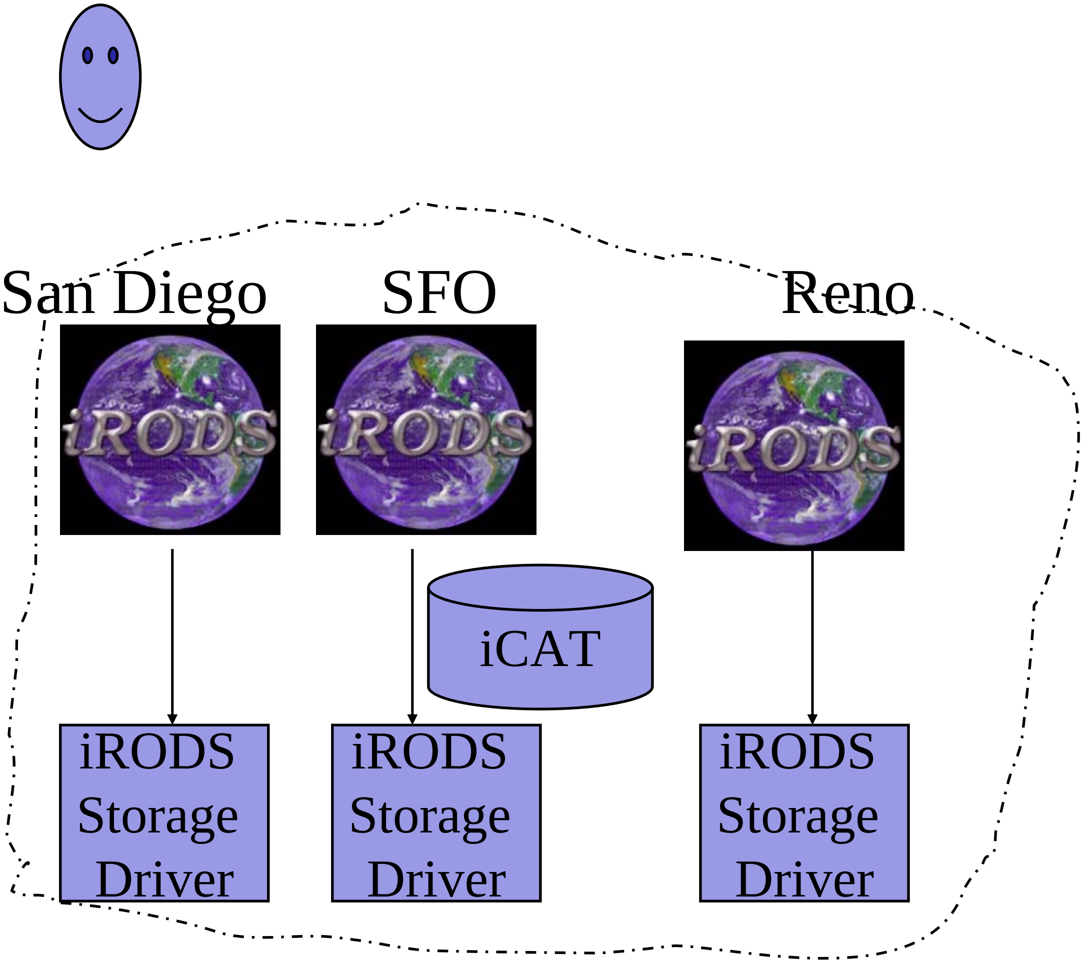
Arun Jagatheesan
San Diego Supercomputer Center &
iRODS.org / DiceResearch.org

26th IEEE (MSST2010) Symposium on Massive Storage Systems & Technologies

May 3-7, 2010

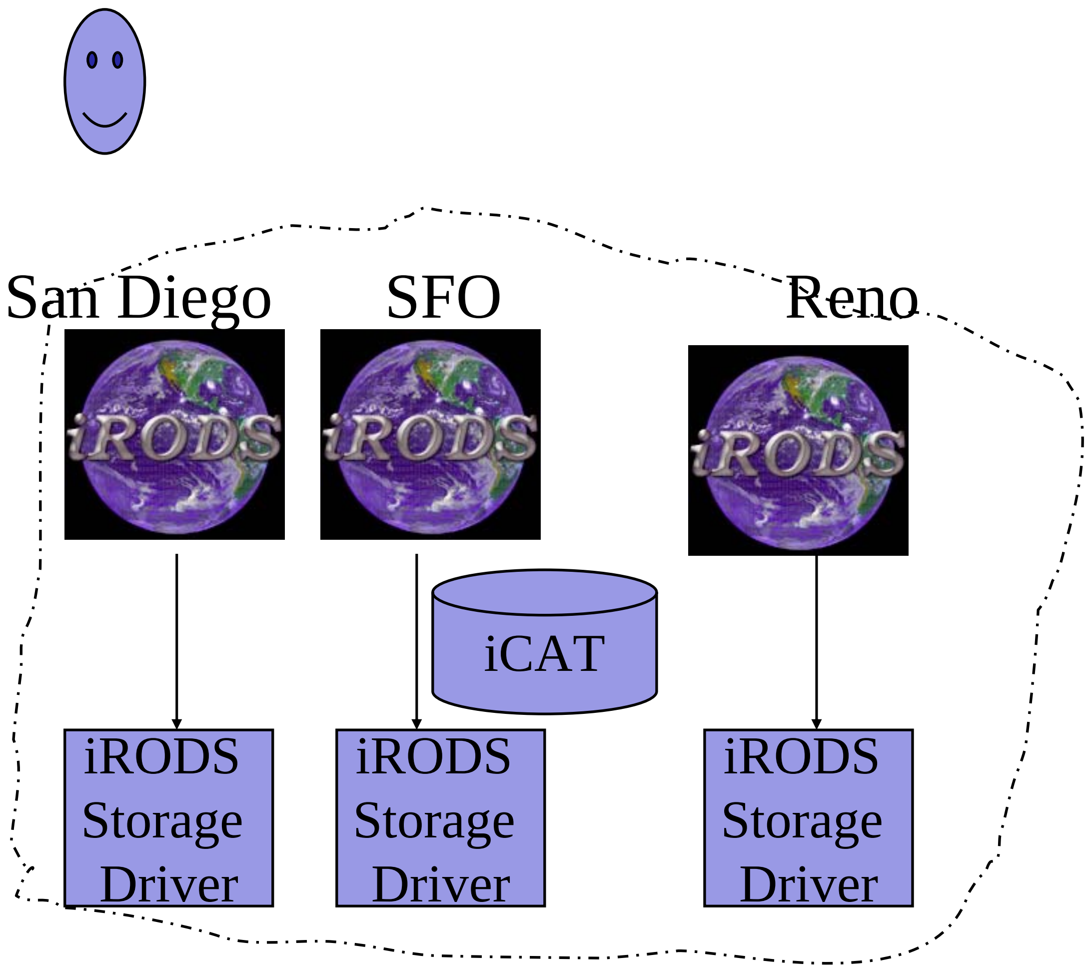
Lake Tahoe, NV

Basic put explained
(with iRules - Trigger_like)



1. Check Auth
2. <pre_process>
3. Decide on a data path option, number of threads, bandwidth etc
4. [sink the data (failover to replica resource automagically]
5. <post_process>
6. <error_recovery>

Basic get explained



1. Check Auth (Logon-server connects to iCAT server)
2. Find optimal copy of the file for that particular client request (uses simple heuristics)
3. Decide on a data path option, number of threads, bandwidth etc
4. Send the data (failover to replica automagically