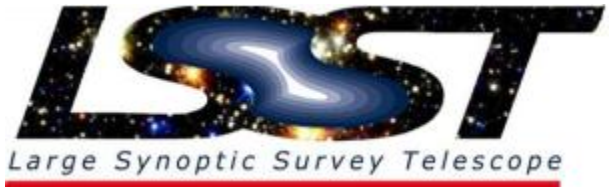
Analysis with Extremely Large Datasets

CHEP'2012
New York, USA

Jacek Becla
SLAC National Accelerator Laboratory

SLAC NATIONAL ACCELERATOR LABORATORY



SLAC
NATIONAL ACCELERATOR LABORATORY

Outline

Background

Techniques

SciDB, Qserv and XLDB-2012

SLAC NATIONAL ACCELERATOR LABORATORY

The Main Challenge

Scale & data complexity & usage

Storing petabytes is easy.
It is what you do with them that matters

SLAC
NATIONAL ACCELERATOR LABORATORY

Scale

Science

HEP
- many petabytes now

Bio
- sequencing – data growth well beyond Moore's law

Geo, photon science
- low petabytes, incompatible data sets

Astro
- low petabytes now
- 100s of PB soon, exabytes planned

Industry

Web companies
- 100s of petabytes now

Many approach peta-scale
- drug discovery
- medical imaging
- oil & gas
- banks

SLAC NATIONAL ACCELERATOR LABORATORY

Data Complexity

adjacency

order

proximity

uncertainty

data lineage

ad-hoc structure

multiple data sources

SLAC NATIONAL ACCELERATOR LABORATORY

Queries / Analysis

Operational load

- *Still challenging @petascale*

Discovery-oriented

- *Complex workflows*
- *Increasingly complex*
- *Ad-hoc, unpredictable, hand-coded, sub-optimal*
- *Not just I/O, but often CPU-intensive*
- *Repeatedly try/refine/verify*
- *Annotate, share*
- *More data = new ways of analyzing it!*

Varying response time needs

- *Long-running – need stable environment*
- *Real-time – need speed, indexes*

Example use cases

- *Pattern discovery*
- *Outlier detection*
- *Multi-point correlations in time and space*
  - *Time series, near neighbor*
- *Curve fitting*
- *Classification*
- *Multi-d aggregation*
- *Cross matching and anti-cross matching*
- *Dashboards / QA*

Lots of commonalities between science and industry

Data access drives the architecture more than anything else!

SLAC NATIONAL ACCELERATOR LABORATORY

Analytics: Industry vs Science

Industry

Time is money
- Real time
- High availability
- Hot fail-over

Rapid change
- Agile software

Science

$0 billion market for analytics

Multi-lab experiments
- No firm control over configuration

Long-term projects
- Extra isolation
- Unknown requirements
- Unknown hardware

SLAC NATIONAL ACCELERATOR LABORATORY

Outline

Background

Techniques

SciDB, Qserv and XLDB-2012

SLAC NATIONAL ACCELERATOR LABORATORY

Architectures

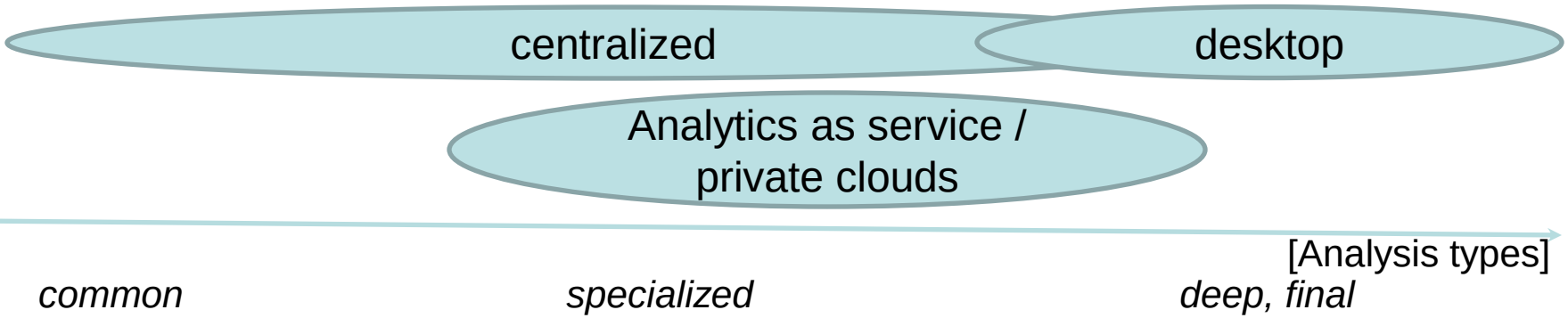
Produce data, take home and analyze
- Photon science

Independent sites, often very different
- Geo, bio

Hierarchical data centers
- HEP
- Astro will follow

Requires paradigm shift

Ultimate architecture?



SLAC NATIONAL ACCELERATOR LABORATORY

Hardware

- Commodity
- Shared-nothing
- Parallelization and sharing resources essential
- Recovery in software
- Watch out for *unexpected* bottlenecks
  - like memory bus
- GPUs?
- 👎 Appliances, vendor locking

SLAC NATIONAL ACCELERATOR LABORATORY

Data Models and Formats

Relational
- ✓ Metadata
- ✓ Calibration

Graphs
- ✓ Non-linear meshes
- ✓ Business data
- ✓ Connection mapping

Arrays
- ✓ Pixel data
- ✓ Time series
- ✓ Spatial
- ✓ Sequences

Strings
- ✓ Sequences

Key/value
- ✓ Anything goes

SLAC NATIONAL ACCELERATOR LABORATORY

Software

- DBMS?
- Hybrid?
- M/R?
- Key-value stores?

What are your must-haves?

- Rapid response?
- Complex joins?
- Strict consistency?
- High availability?
- Full control over data placement?
- Dynamic schema?
- Reproducibility?
- Scalability?
- Full table scans?
- Aggregations?

One size does not fit all

In database + near database processing

SLAC NATIONAL ACCELERATOR LABORATORY

Reducing Cost

- Discarding
- Streaming
- Summarizing
- Compressing
- Virtualizing

SLAC NATIONAL ACCELERATOR LABORATORY

Pushing Computation to Data

- Moving data is expensive

- Happens at every level
  - Send query to closest center
  - Process query on the server that holds data

SLAC NATIONAL ACCELERATOR LABORATORY

Improving I/O Efficiency

- Limit accessed data
  - Generate commonly accessed data sets.
    Cons: delays and restricts
- De-randomize I/O
  - Copy and re-cluster pieces accessed together
- Trade I/O for CPU
  - Parallel CPU much cheaper than parallel I/O
- Combine I/O
  - Shared scans

Chunking and Parallelizing

- Only certain sharding algorithms appropriate for correlated data
- Data-aware partitioning
  - Multi-level
  - With overlaps
  - Custom indices

Easy crude cost estimate

Easy intermediate results

SLAC NATIONAL ACCELERATOR LABORATORY

Data Access – Indexes

- Essential for real-time

- But indexes can be very harmful
  - High maintenance cost
  - Too much random I/O

Chunking = indexing

SLAC NATIONAL ACCELERATOR LABORATORY

Isn’t SSD the Solution?

- SSD vs spinning disk
  - Sequential I/O ~3x faster
  - Random I/O ~100-200 x faster
- 1% of 1 billion records = 10 million accesses

SLAC NATIONAL ACCELERATOR LABORATORY

Data Access – Shared Scanning

- Idea
  - Scan partition per disk at a time
  - Share fetched data among queries
  - Synchronize scans for joins
- Benefits
  - Sequential access
  - Predictable I/O cost and response time
  - Conceptually simple, no need for indexes
  - Best bet for unpredictable, ad hoc analysis touching large fraction of data

SLAC NATIONAL ACCELERATOR LABORATORY

Operators, APIs, Tools

- Procedural language for complex analyses
  - C++, Python, IDL, even Fortran
  - Pig, sawzall
  - SQL far from ideal
- Frequently rely on low-level access
- Tools (R, MATLAB, Excel, …) don't scale
  - Sample or reduce to fit in RAM
- Specialized, domain-specific tools

Final Hints

- Keep things simple
- Don’t forget to automate … everything
- Reuse, don’t reinvent
- Share lessons learned

SLAC NATIONAL ACCELERATOR LABORATORY

Outline

Background

Techniques

SciDB, Qserv and XLDB-2012

SLAC NATIONAL ACCELERATOR LABORATORY

SciDB

- Open source, analytical DBMS
  - All-in-one database and analytics
  - True multi-dimensional array storage
    - With chunking, overlaps, non-integer dimensions
  - Runs on commodity hardware grid or in a cloud
  - Complex math, window operations, regrid, …

- Production version this summer

"download now, try it out, and give us feedback and feature requests"
– SciDB CEO

SLAC NATIONAL ACCELERATOR LABORATORY

Qserv

- Blend of RDBMS and Map/Reduce
  - Based on MySQL and xrootd
- Key features
  - Data-aware 2-level partitioning w/overlaps, 2nd level materialized on the fly
  - Shared scans
  - Complexity hidden, all transparent to users
- 150-node, 30-billion row demonstration
- Beta version late summer 2012

SLAC NATIONAL ACCELERATOR LABORATORY

LHC talk?

SLAC NATIONAL ACCELERATOR LABORATORY

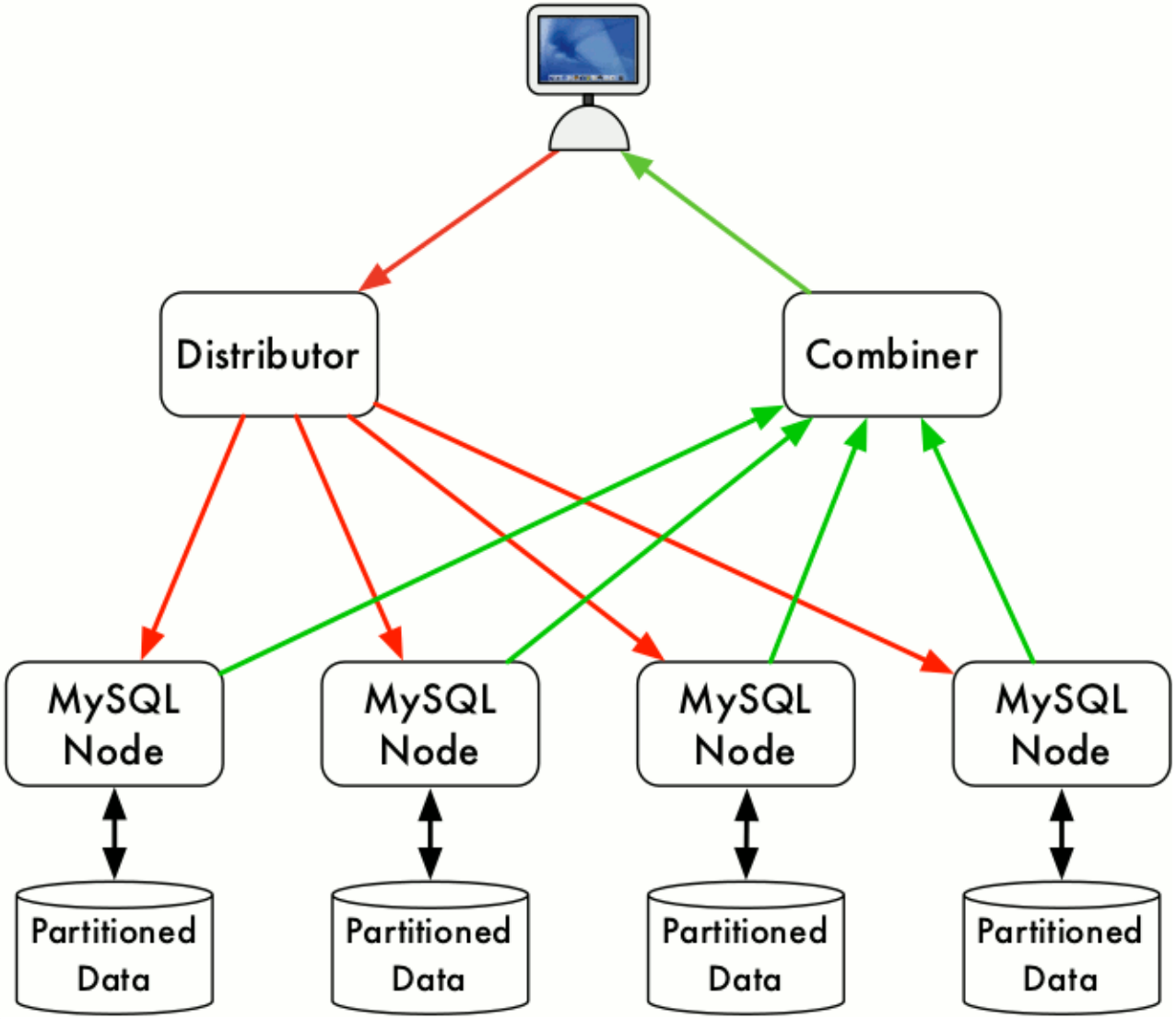
Summary

- Storing a petabyte is easy. Analyzing it is *not.*
- Scale, data complexity, access patterns drive architectures
- Many commonalities between science and industry
  - Share lessons
  - Reuse, don’t reinvent

SLAC NATIONAL ACCELERATOR LABORATORY

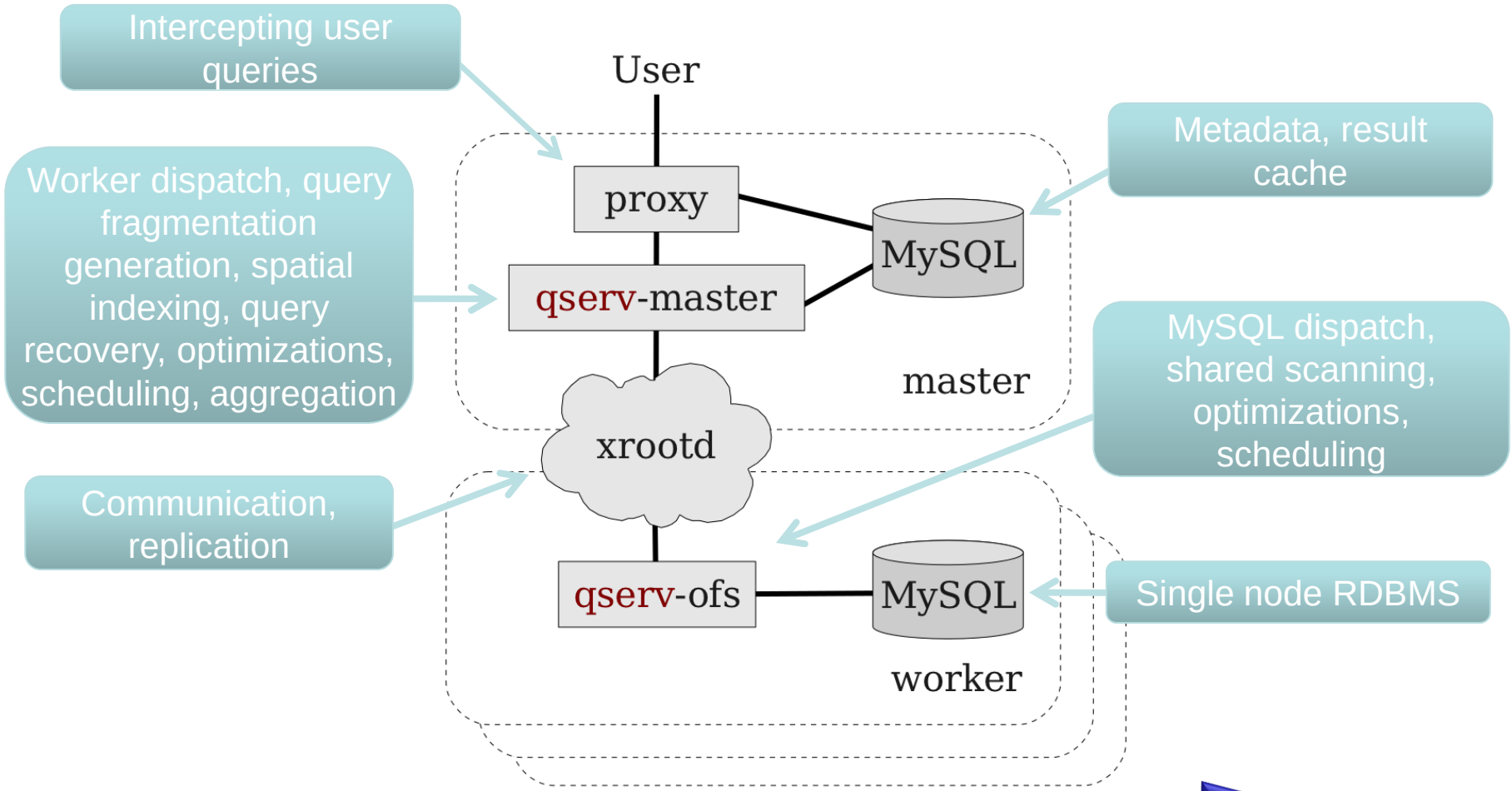
Back up slides

Baseline Architecture



SLAC NATIONAL ACCELERATOR LABORATORY

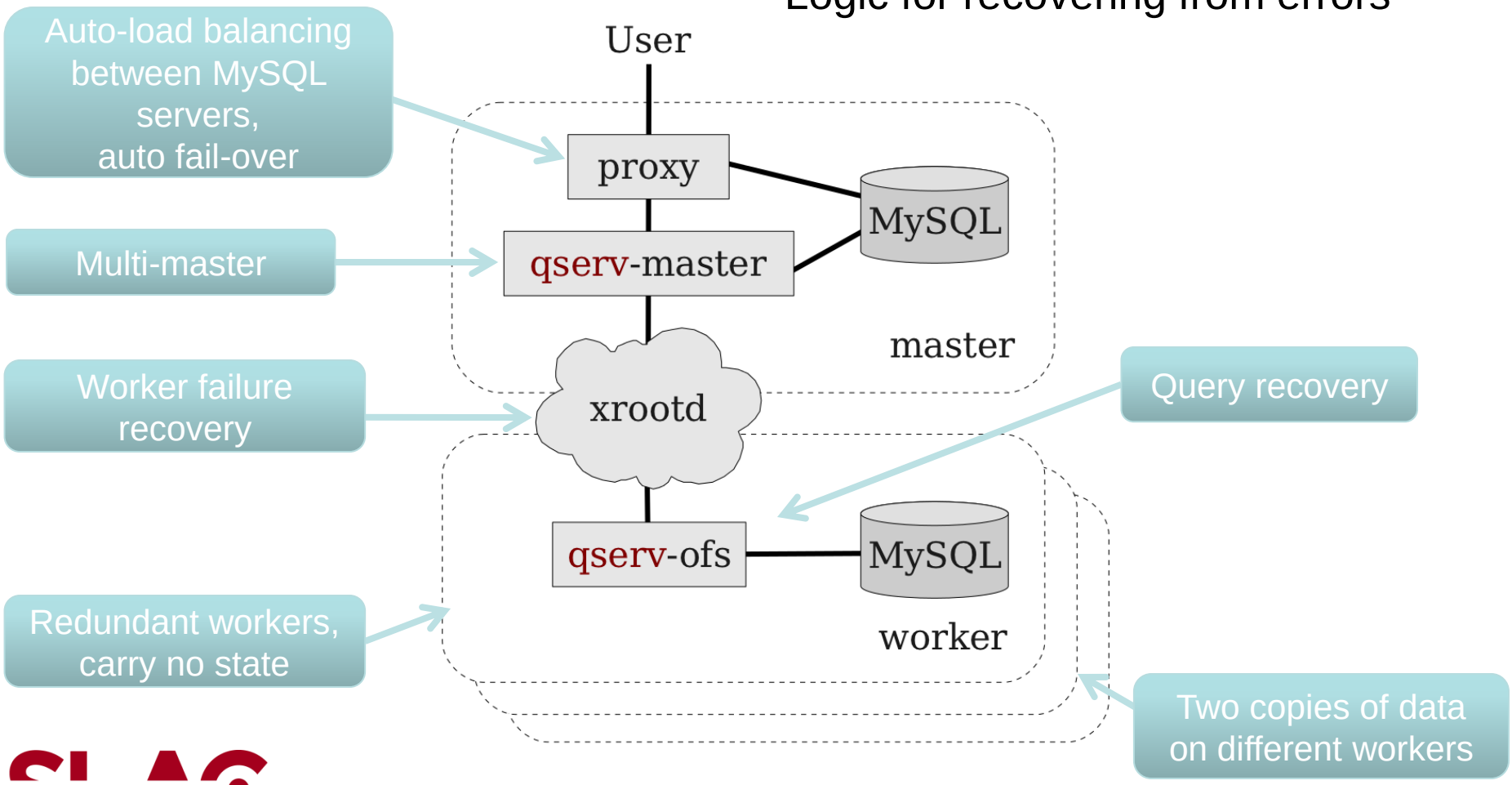
Prototype Implementation - *Qserv*



SLAC NATIONAL ACCELERATOR LABORATORY

Qserv Fault Tolerance

- Components replicated
- Failures isolated
- Narrow interfaces
- Logic for handling errors
- Logic for recovering from errors



SLAC NATIONAL ACCELERATOR LABORATORY