DeclStore: Layering is for the Faint of Heart

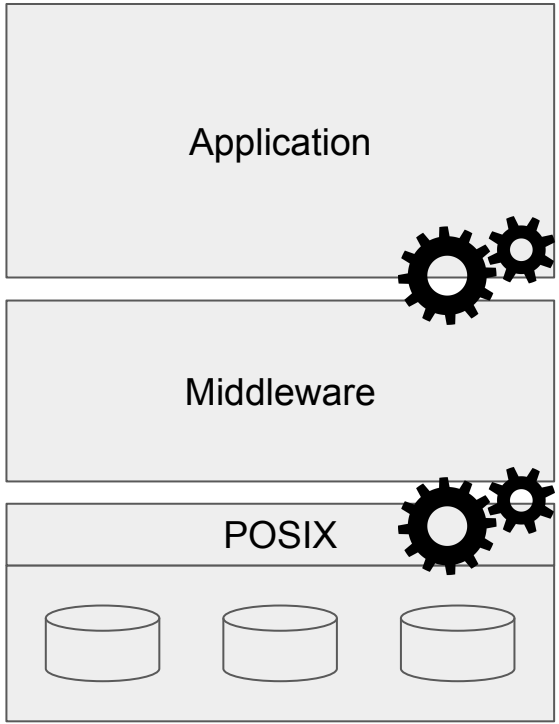
Noah Watkins, Michael Sevilla, Ivo Jimenez, Kathryn Dahlgren, Peter Alvaro, Shel Finkelstein, Carlos Maltzahn

UC SANTA CRUZ

CROSS | CENTER FOR RESEARCH IN OPEN SOURCE SOFTWARE

SRL
UCSC Systems Research Lab

Layers on layers on layers



Application-specific storage systems

Unified storage systems

Application

Middleware

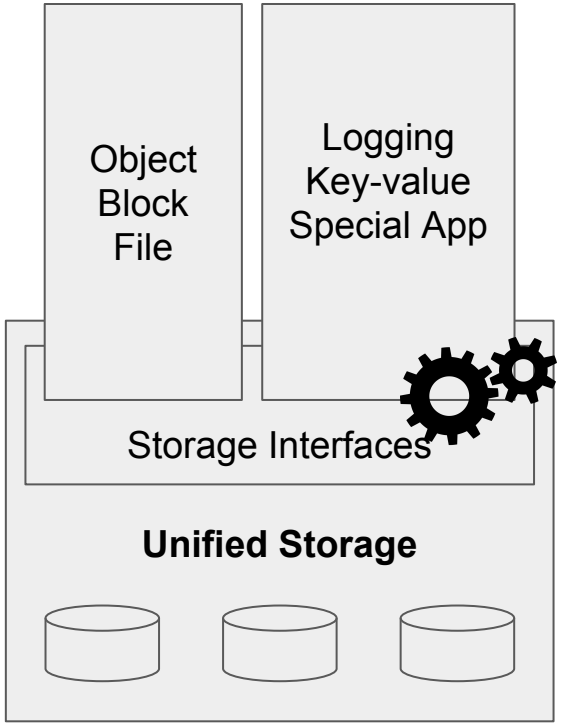
POSIX

Special App

Fancy Sto

Examples

- OnTap
- Swift
- Ceph
- GPFS
- EMC



SRL
UCSC Systems Research Lab

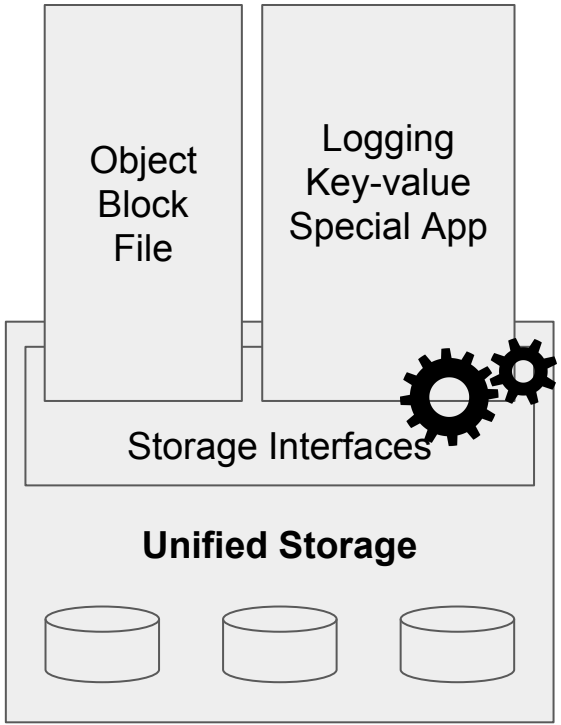
Eliminating layers: big challenges, big rewards

Big Question

If unified storage becomes the new normal, what new challenges will need to be addressed in system design?

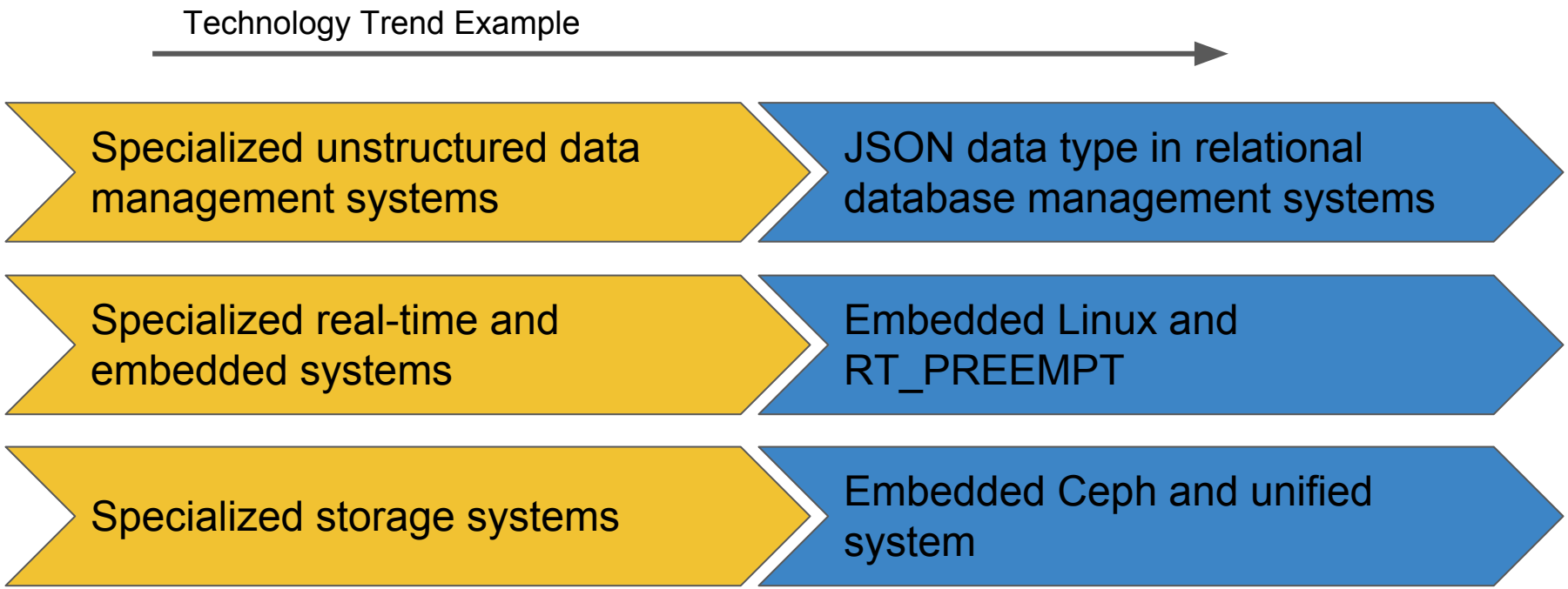
Talk Outline

- Motivating unified storage
- Storage interface design space
- Declarative storage interfaces

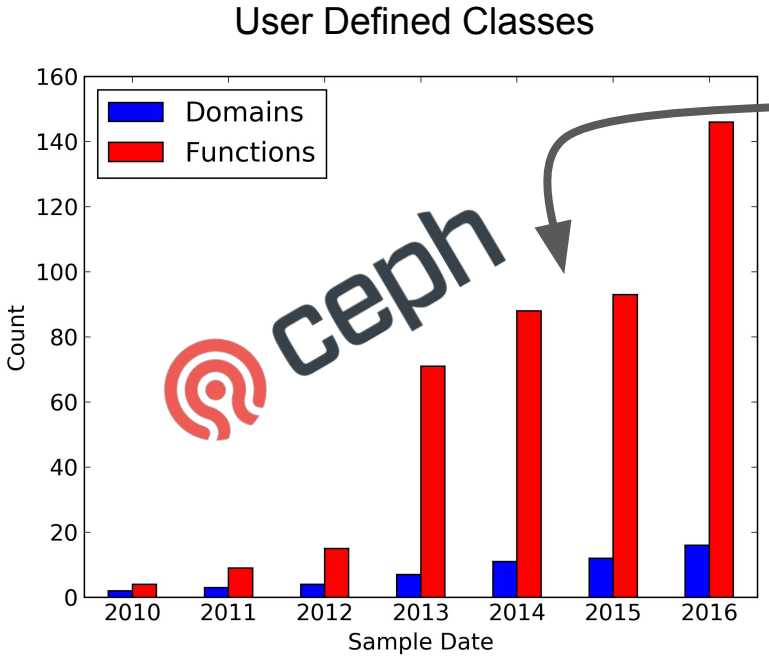


SRL
UCSC Systems Research Lab

Motivation: systems trend towards generality



Motivation: breaking the narrow waist model

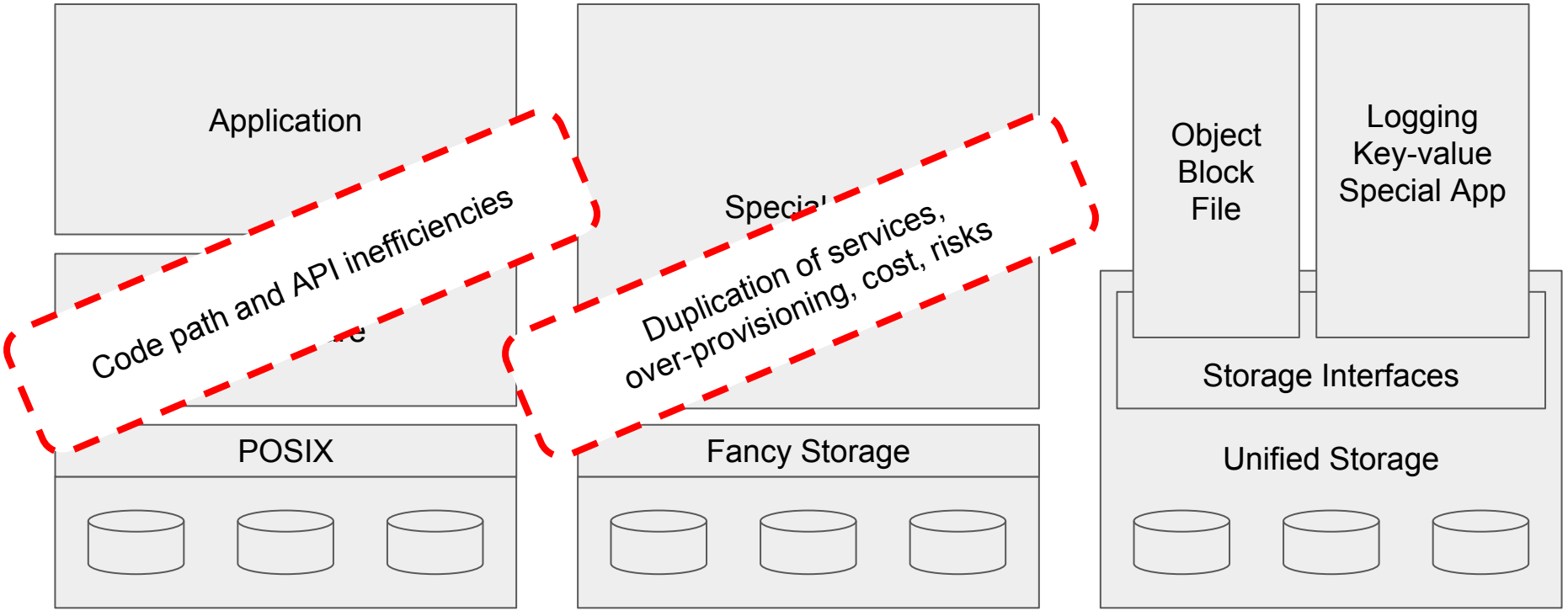


Examples

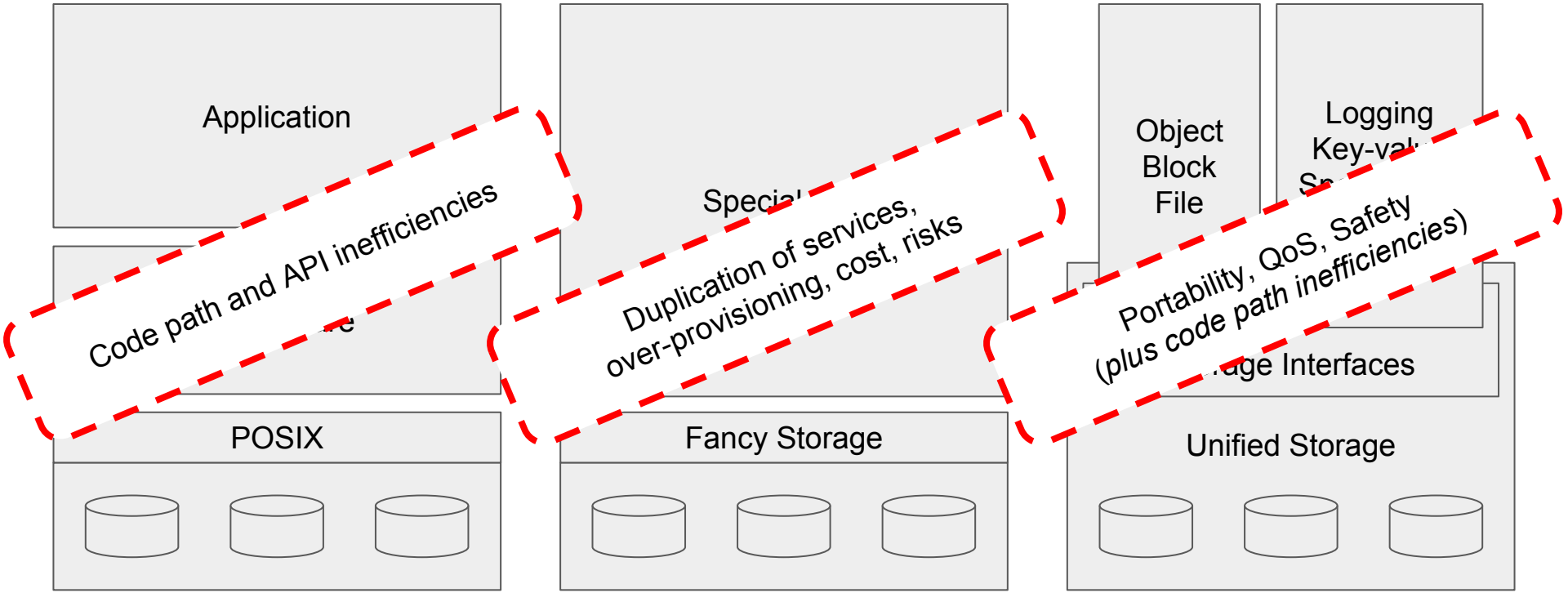
- Locking, Log
  Concurrency
- Metadata
  managemer
- Remote com

Developers willing to break layers and use non-standard APIs

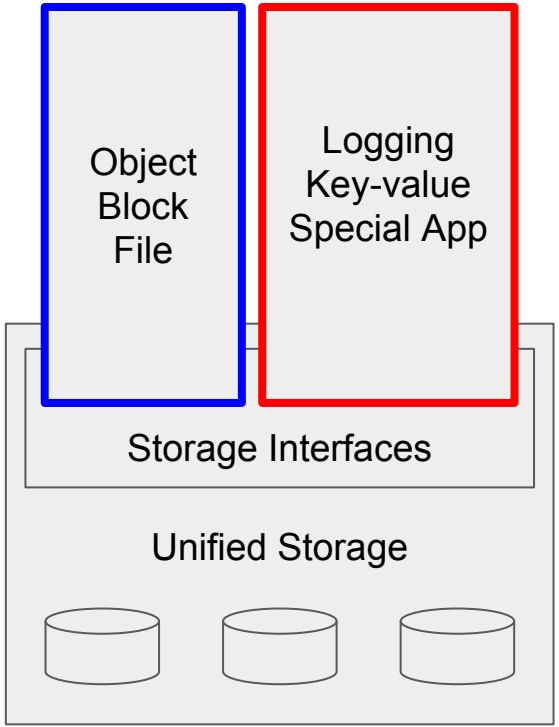
Why in practice do people break down layers?



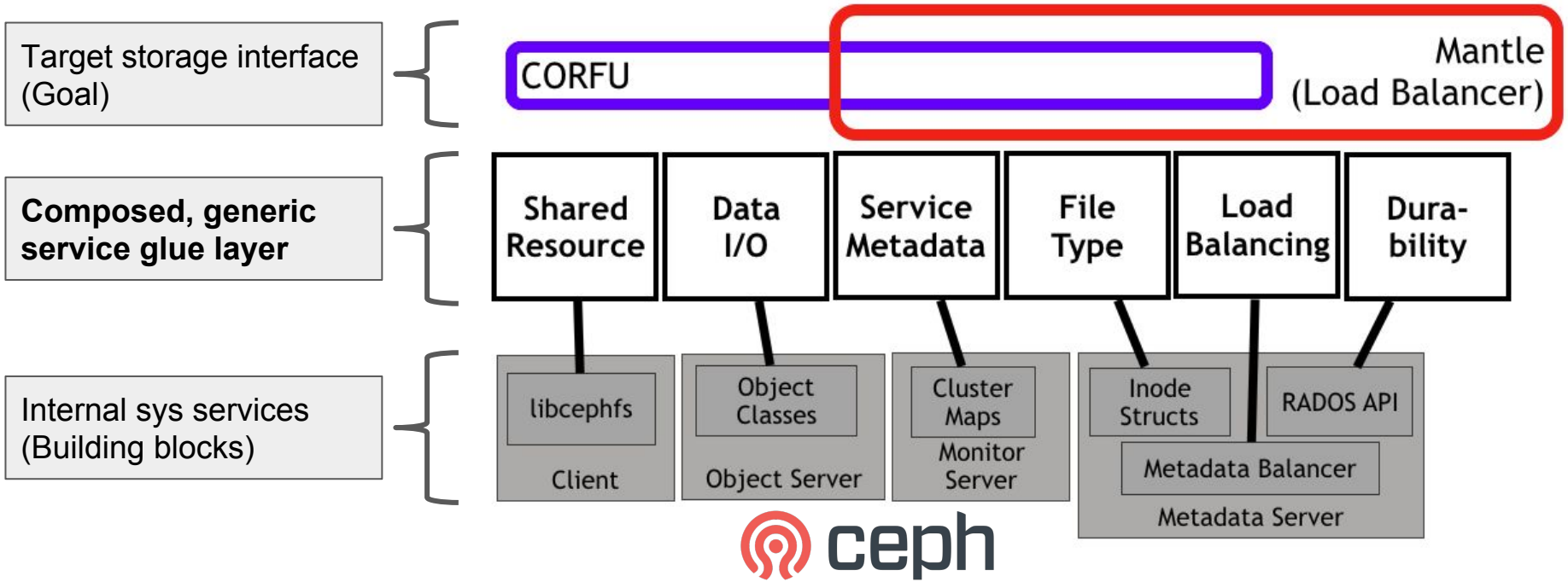
Unified storage has its own set of challenges



Exploring design challenges in unified storage



Malacology: programmable interface research platform

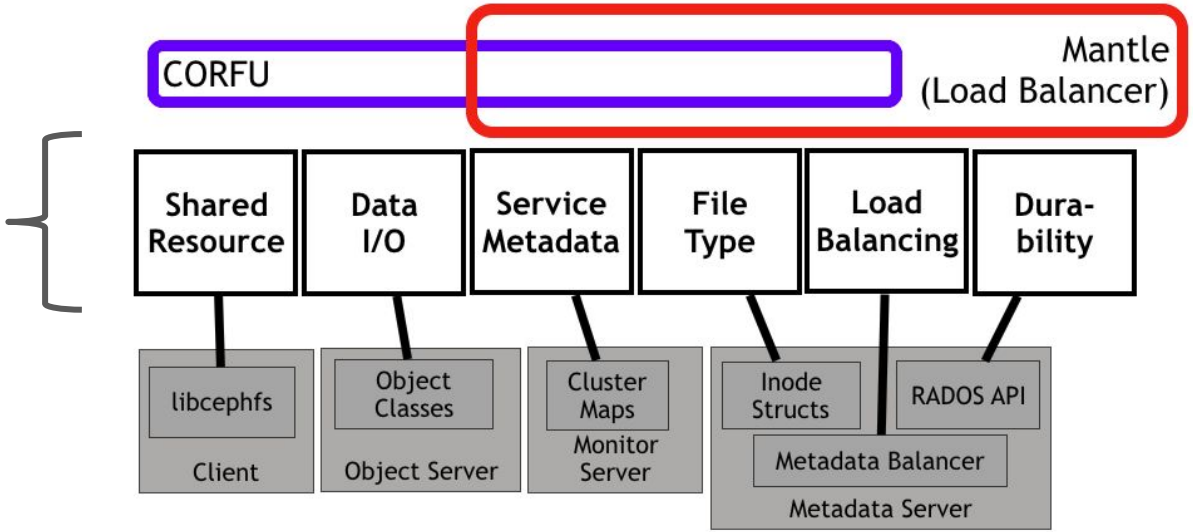


Malacology: A Programmable Storage System, M. Sevilla, N. Watkins, I. Jimenez, P. Alvaro, S. Finkelstein, J. LeFevre, C. Maltzahn, EuroSys '17
Mantle: A Programmable Metadata Load Balancer for the Ceph File System, M. Sevilla, N. Watkins, C. Maltzahn, I. Nassi, S. Brandt, et. al, SC '15
CORFU: A Shared Log Design for Flash Clusters, Mahesh Balakrishnan, Dahlia Malkhi, Vijayan Prabhakaran, and Ted Wobber, Michael Wei, et. al, NSDI '12

SRL UCSC Systems Research Lab

Keeping pace with evolving storage systems

Hard-wired prototype implementations



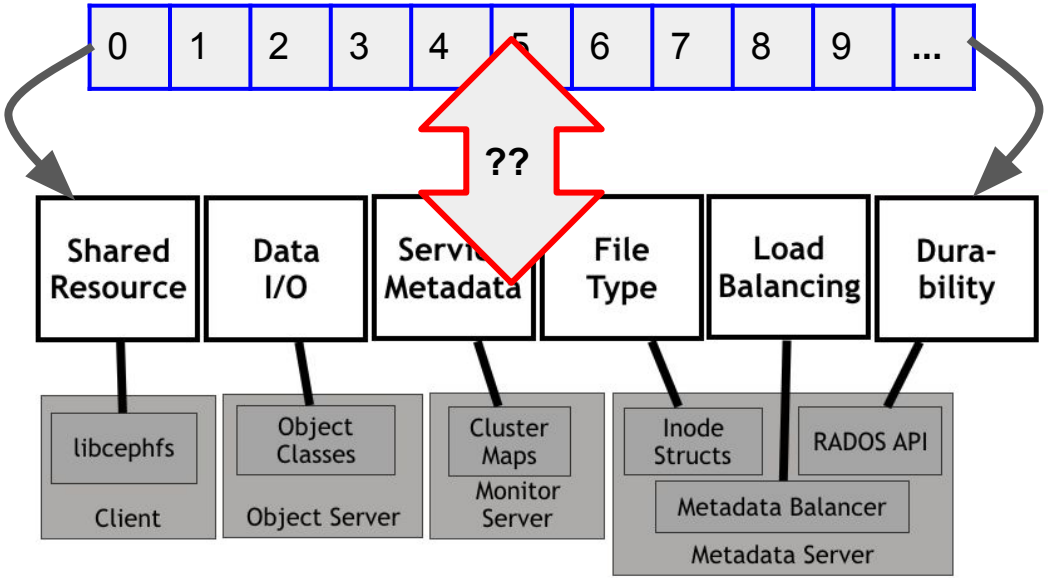
How to take advantage of...

- Software upgrades
- New hardware
- Additional services
- Performance features

SRL
UCSC Systems Research Lab

Example: building the CORFU shared-log interface

- CORFU[1] is a dist. shared-log
- High-performance design
  - Append and random reads
  - I/O parallelism (striping)
  - Soft-state network counter
- **Challenges**
  - Find a *good* mapping
  - Performance optimizations
  - Minimal maintenance

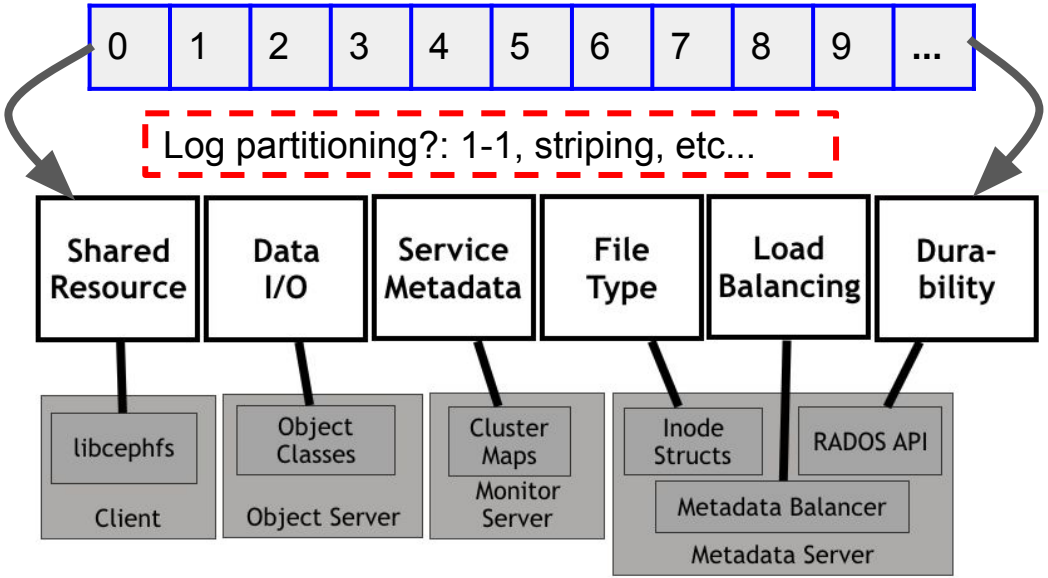


CORFU: A Shared Log Design for Flash Clusters, Mahesh Balakrishnan, Dahlia Malkhi, Vijayan Prabhakaran, and Ted Wobber, Michael Wei, et. al, NSDI '12

SRL UCSC Systems Research Lab

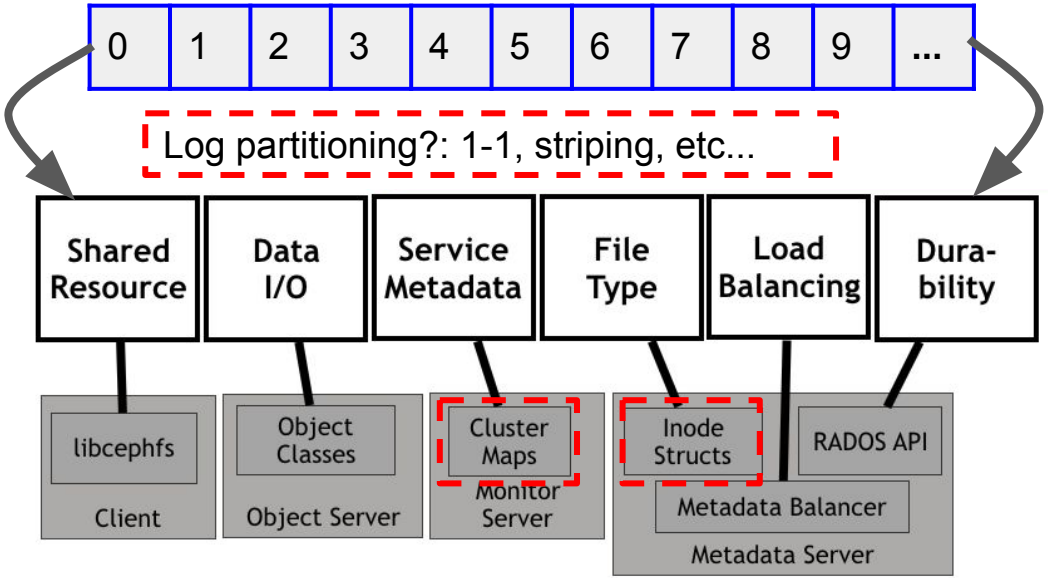
Example: building CORFU distributed shared-log

- Implementation options
  - Partitioning



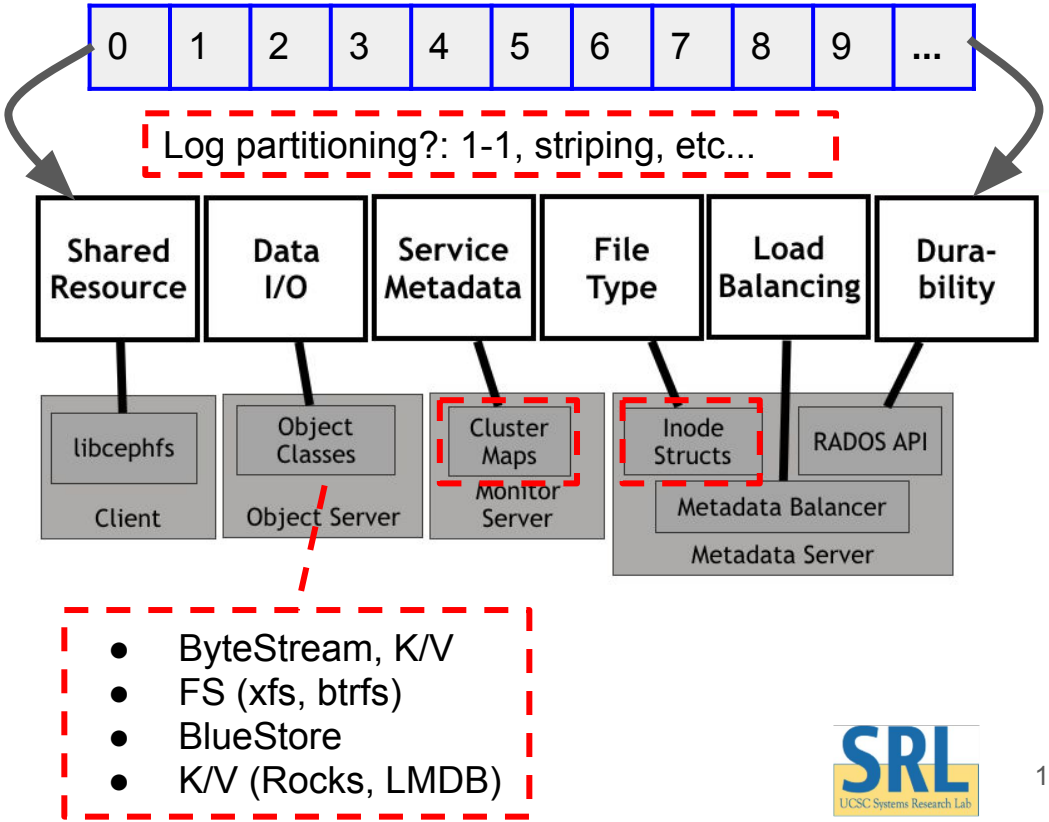
Example: building CORFU distributed shared-log

- Implementation options
  - Partitioning
  - Metadata



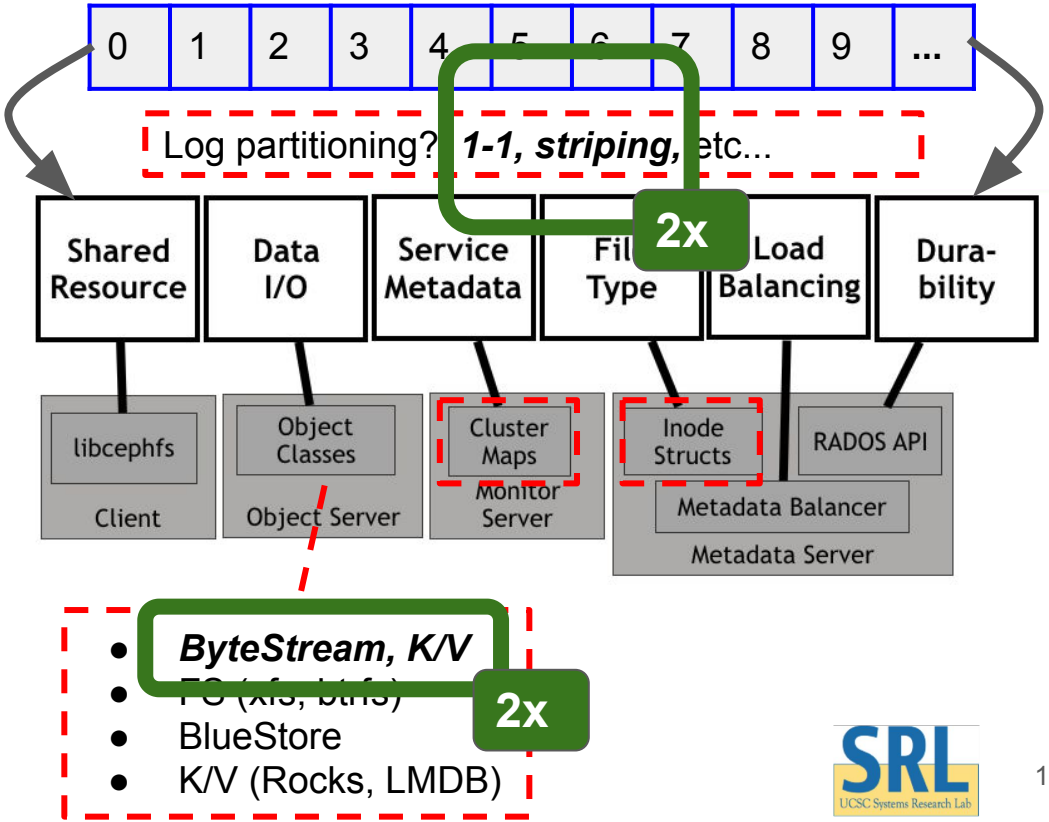
Example: building CORFU distributed shared-log

- Implementation options
  - Partitioning
  - Metadata
  - I/O interfaces



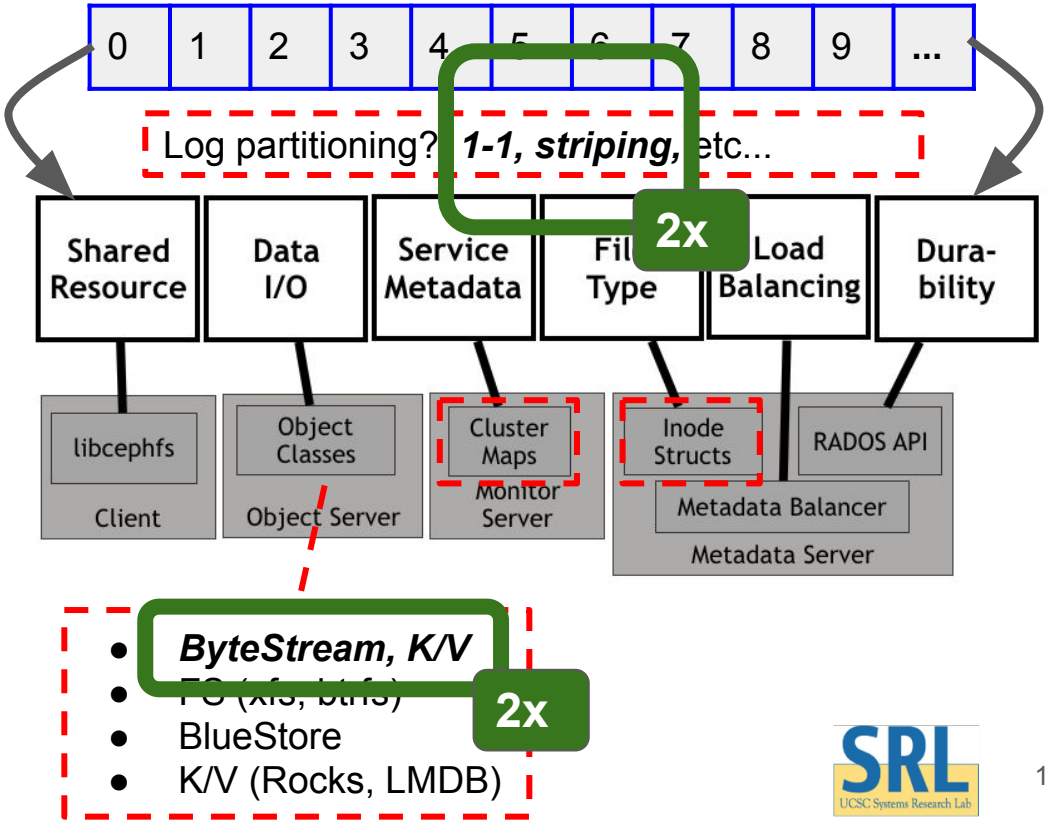
Example: building CORFU distributed shared-log

- Implementation options
  - Partitioning
  - Metadata
  - I/O interfaces
- Constructed 4 versions
  - **2** partitioning * **2** I/O interfaces
  - Partitioning (**1-1, striping**)
  - I/O interface (**bytestream, K/V**)



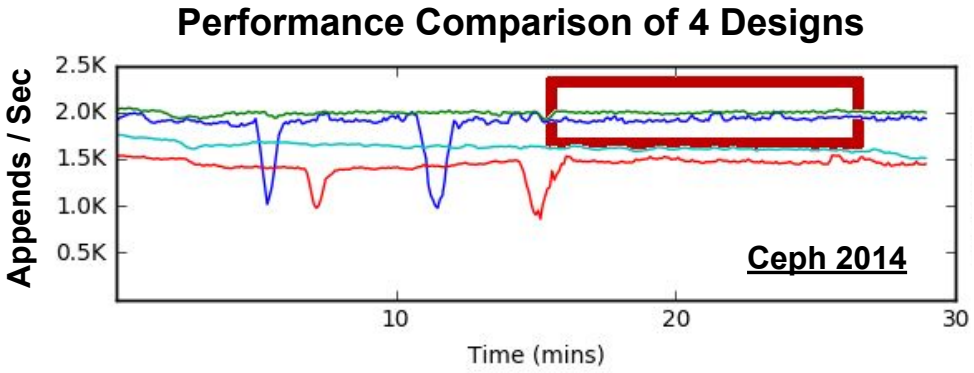
Example: building CORFU distributed shared-log

- Implementation options
  - Partitioning
  - Metadata
  - I/O interfaces
- Constructed 4 versions
  - **2** partitioning * **2** I/O interfaces
  - Partitioning (**1-1, striping**)
  - I/O interface (**bytestream, K/V**)
- Hard-wired implementations
  - Few 1000's LOC
  - Manually optimize and switch between approaches



Example: shared-log performance toss-up in 2014

- Four implementations
- Ceph version circa 2014
- Graph takeaways
  - Clear performance losers
  - **Similar** top performers
- Our claim…
  - Select simpler implementation
  - Added complexity for no benefit
- What is complexity?
  - Lines of code
  - Conceptual

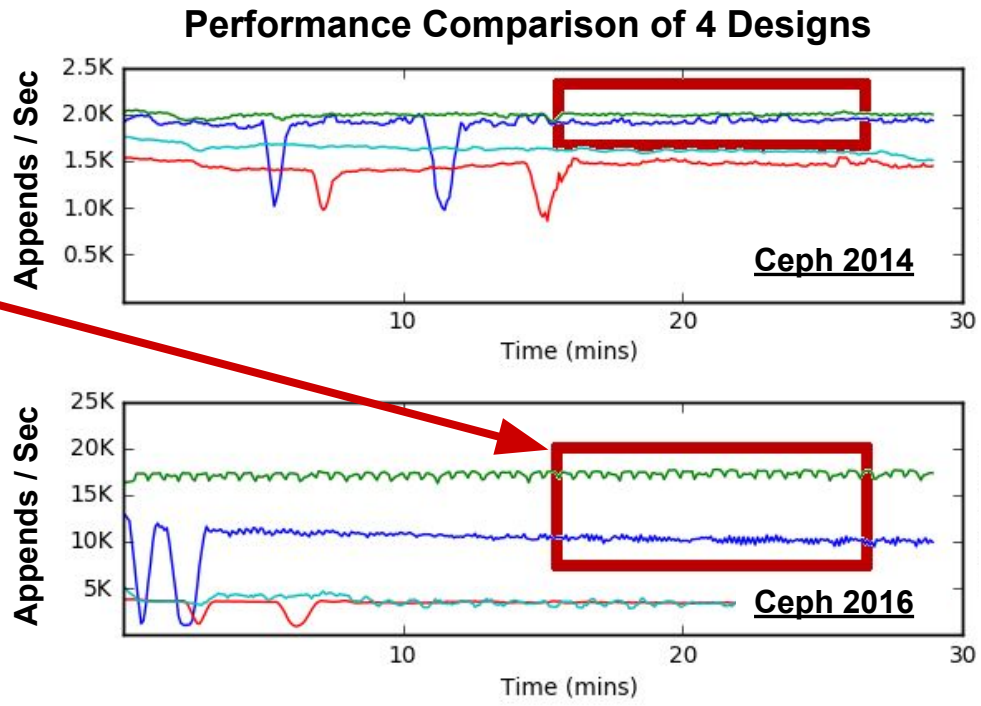


Example: clear design choice in 2016

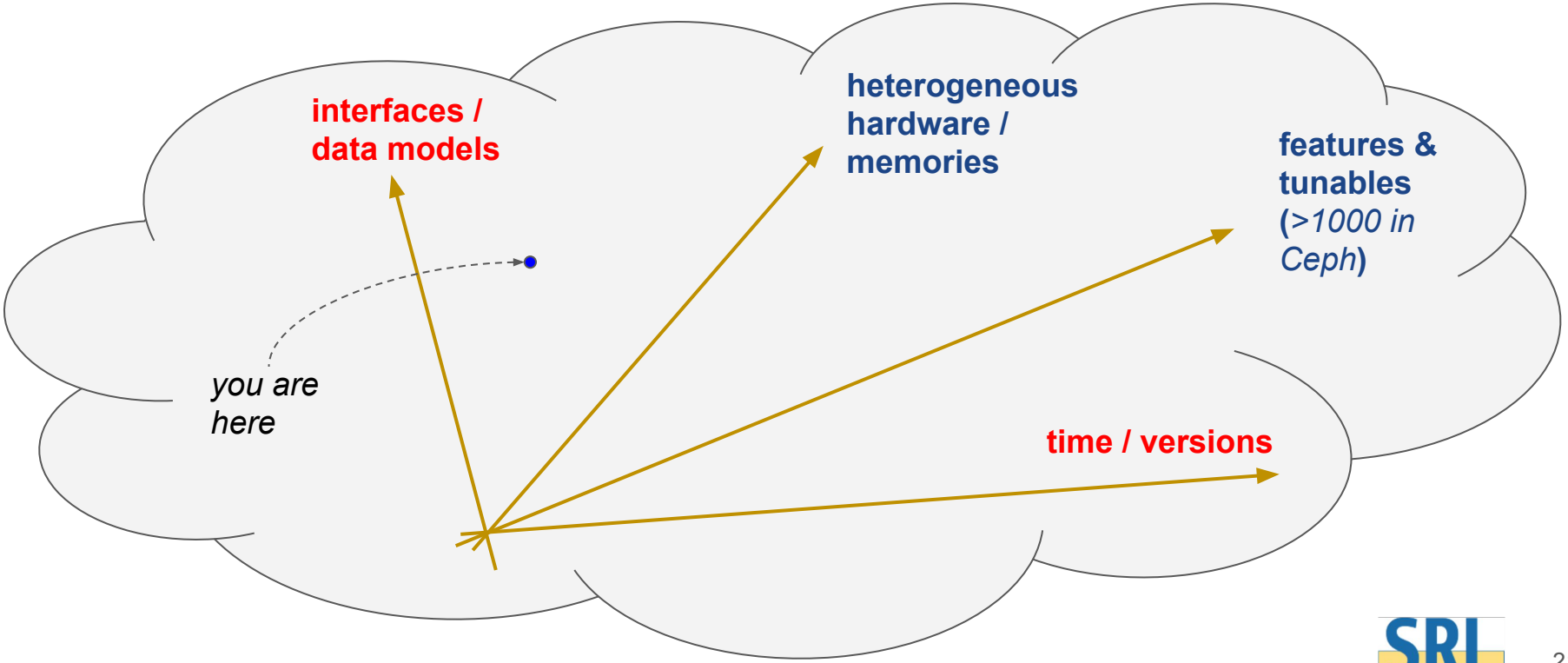
- Same implementations
- Same hardware / benchmark
- Newer version of Ceph
- ***Clear performance winner in 2016***

Takeaway:

- **A reasonable choice in 2014 would be a poor choice in 2016 after a simple upgrade**



Problem: navigation of large design space



DeclStore: *express* storage interfaces declaratively

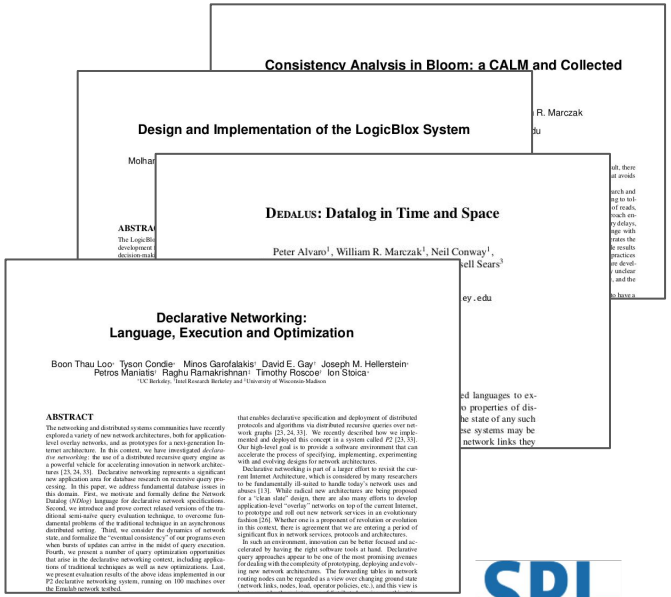
- Freedom from storage system and domain expertise
- Abstractions over storage services and interfaces
- The point is… we need a high-level language
  - Many possible mappings
  - Many degrees of freedom
- Generate storage interface implementations
- Exploit relational database optimization research

SRL
UCSC Systems Research Lab

DeclStore: building on a strong foundation

- Based on *Bloom* programming language
- High-level declarative language (Alvaro, CIDR `11)
  - Abstract relational data model
  - Programs are queries
- Many degrees of freedom on reordering
  - Subject to optimization
- Evidence this is possible...
  - LogicBlox (Aref, SIGMOD 2015)
  - Dedalus (Alvaro, et. al, Datalog 2010)
  - Declarative Networking (Loo, SIGMOD 2006)

<~ bloom



Example: a declarative specification for CORFU

- System state modeled as a set of relations
  - Persistent state
  - Input/Output
- Interfaces are queries over a request stream
  - Operate on persistent state
  - Collection operations
- Full specification
  - Just another slide's worth
  - Compare 1K's LOC C++

```
# persistent relations
table :epoch, [:epoch]
table :log,   [:pos] => [:state, :data]

# interfaces are also like tables
interface input,  :op,   [:type, :pos, :epoch] => [:data]
interface output, :ret,  [:type, :pos, :epoch] => [:retval]
```

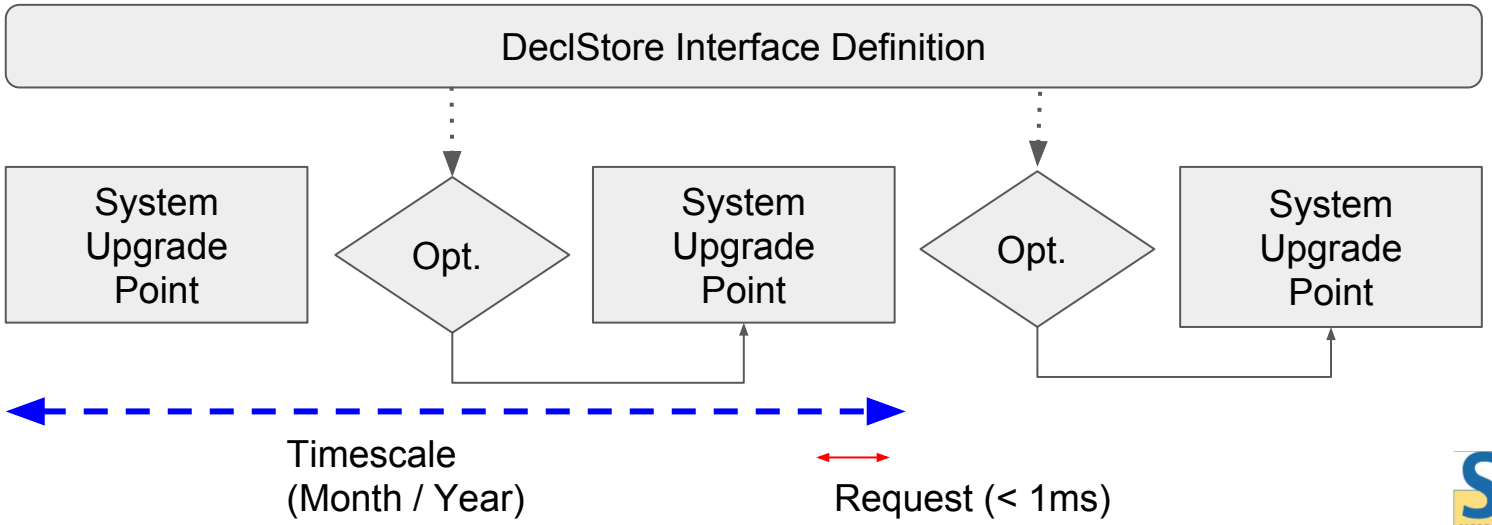
```
bloom :write do
  temp :valid_write <= write_op.notin(found_op)
  log <+ valid_write { |o| [o.pos, 'valid', o.data] }
  ret <= valid_write { |o| [o.type, o.pos, o.epoch, 'ok'] }
  ret <= write_op.notin(valid_write) { |o|
    [o.type, o.pos, o.epoch, 'read-only']
  }
end
```

What’s the catch?

- This is a storage system, not a database...
- Generality introduces overhead
- Additional complexity and layers
  - We don’t want layers!
- The cost of online optimization

Taking advantage of time scales: offline optimization

- We are generating *storage service implementations*
- Automate integration on new features and hardware
- Planned upgrade points v.s. per-request optimization



Example: a declarative specification for CORFU

- No hard-wired choice of storage interface!
  - Bytestream vs K/V
  - Partitioning strategy
- Take advantage of existing analysis tools and techniques!
- Convincingly correct!

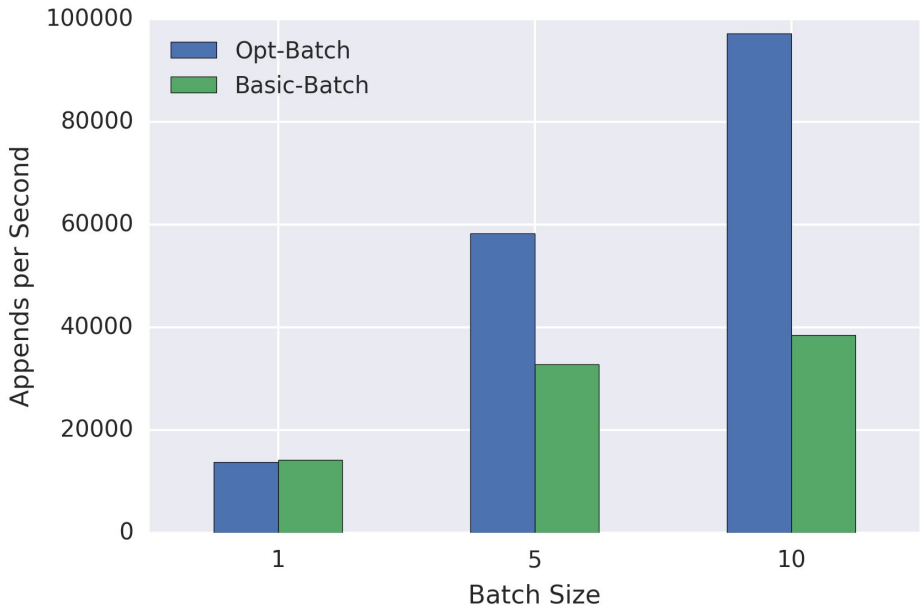
```
# persistent relations
table :epoch, [:epoch]
table :log,   [:pos] => [:state, :data]

# interfaces are also like tables
interface input,  :op,   [:type, :pos, :epoch] => [:data]
interface output, :ret,  [:type, :pos, :epoch] => [:retval]
```

```
bloom :write do
  temp :valid_write <= write_op.notin(found_op)
  log <+ valid_write { |o| [o.pos, 'valid', o.data] }
  ret <= valid_write { |o| [o.type, o.pos, o.epoch, 'ok'] }
  ret <= write_op.notin(valid_write) { |o|
    [o.type, o.pos, o.epoch, 'read-only']
  }
end
```

SRL UCSC Systems Research Lab

Example: performance benefit of group commit

- Ceph I/O handling is conservative
  - Queuing, locking, ordering
- Log operations are independent
  - Determined through analysis
  - Amortize across I/O and code path
- Basic Operation Batching
  - Lots of internal code traversal
- Using efficient interfaces
  - Vector or range-based interfaces
- **See paper for more details**
  - **Cost models and outliers**



SRL
UCSC Systems Research Lab

Conclusion: expanding the set of storage interfaces...

- **is becoming increasingly common**
  - new systems being built
  - old systems being adapted
- **will create an entirely new set of challenges; we showed**
  - hard-wired solutions are difficult to maintain
  - even a basic software upgrade can have a big impact
- **through declarative specifications can address many concerns**
  - lower maintenance costs
  - automate system evolution

SRL
UCSC Systems Research Lab

Thank you… Questions?

DeclStore: Layering is for the Faint of Heart

Noah Watkins, Michael Sevilla, Ivo Jimenez, Kathryn Dahlgren, Peter Alvaro, Shel Finkelstein, Carlos Maltzahn