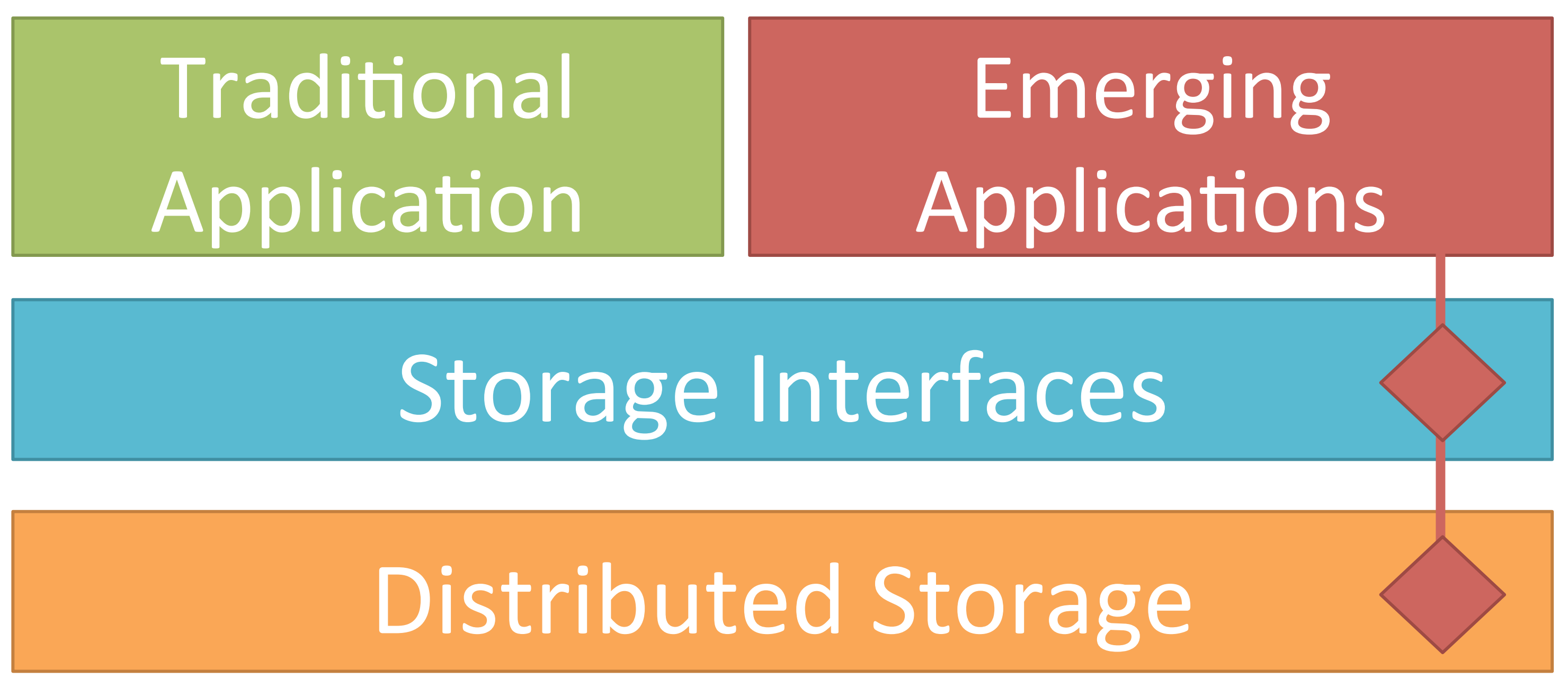
Brados: Declarative,Programmable Object Storage

Noah Watkins, Michael Sevilla, Ivo Jimenez,
Neha Ohja, Peter Alvaro, Carlos Maltzahn

UC SANTA CRUZ CROSS CENTER FOR RESEARCH IN OPEN SOURCE SOFTWARE

Storage Abstractions Are Changing

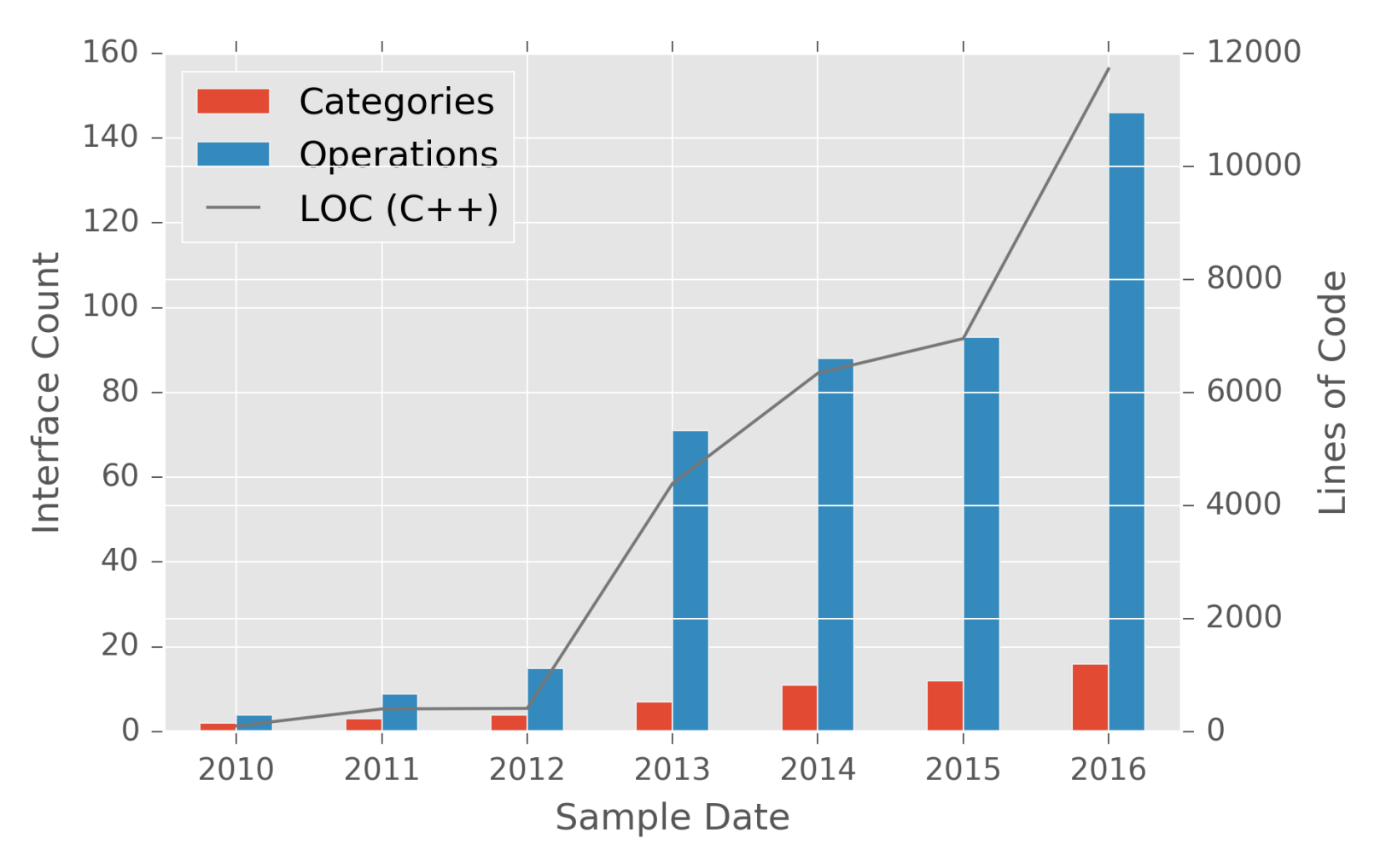


Emerging applications are integrating into the entire storage stack, constructing domain-specific interfaces, and reusing services.

- Clear, direct application semantics
- Control over low-level data layouts

Storage System Programmability in the Wild

Category	Specialization	Methods
Locking	Shared Exclusive	6
Logging	Replica State Timestamped	3 4 4
Garbage Collection	Reference Counting	4
Metadata Management	RBD RGW User Version	37 27 5 5

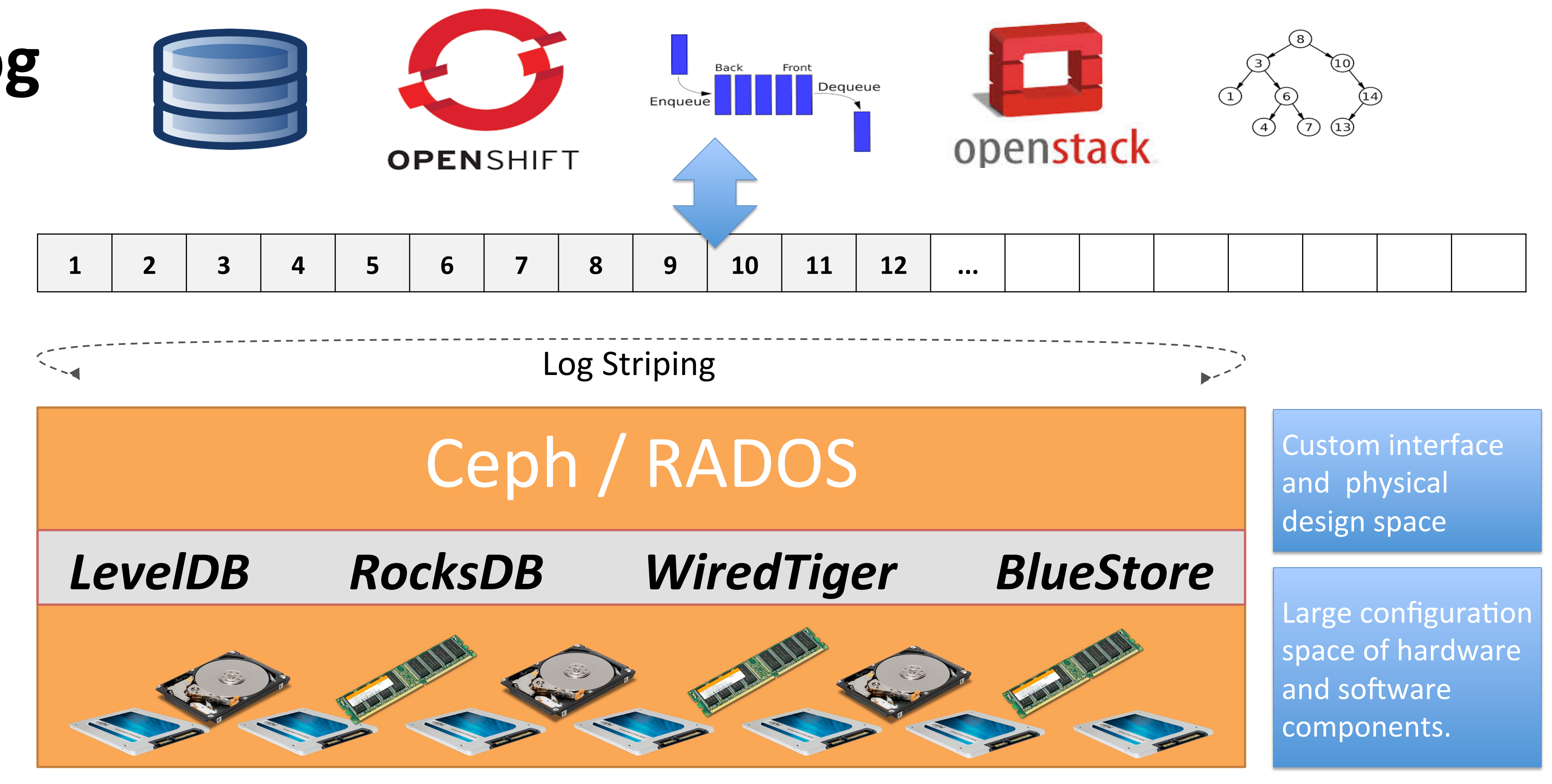


- Open-source storage systems are exposing internal services to applications
- Ceph and RADOS provide numerous domain-specific interfaces
- In-production interfaces support high-profile applications (e.g. OpenStack)
- Beginning to see third-party interface contributions

Example Service : Distributed Shared-Log

Driving example is ZLog, an implementation of the CORFU [1] high-performance shared-log protocol on top of software-defined storage.

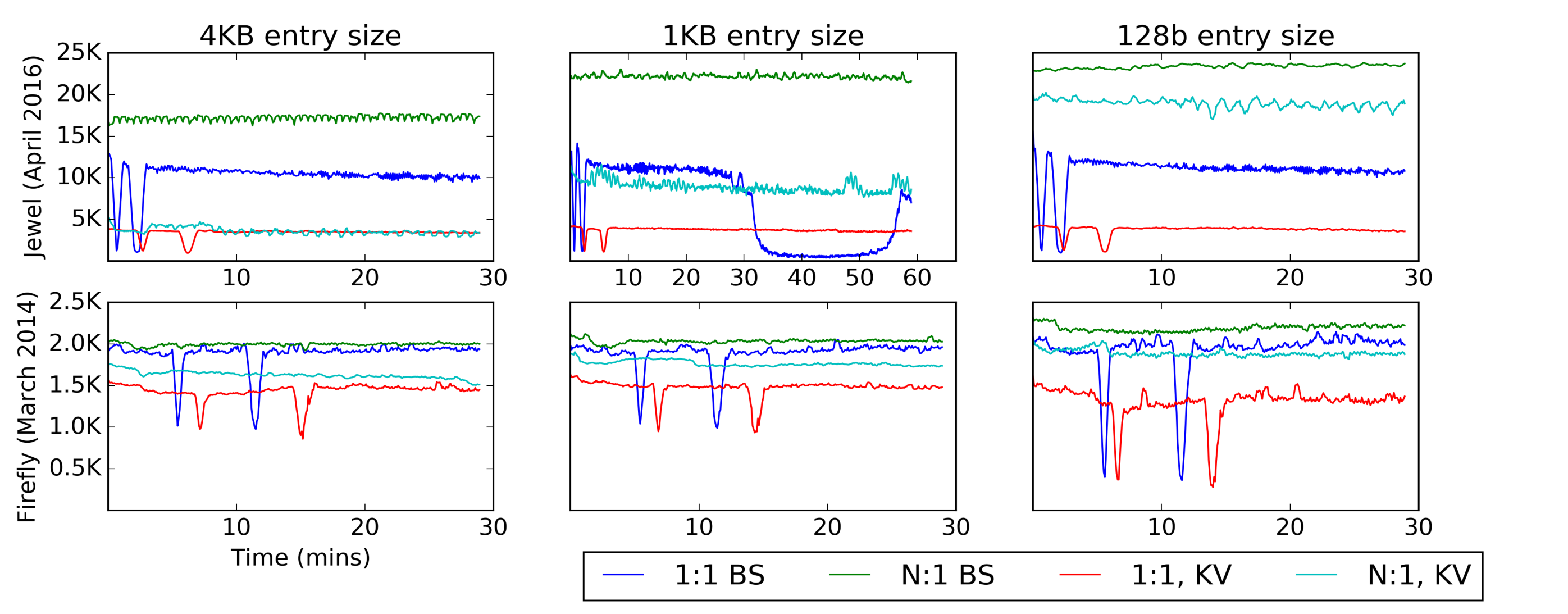
- Service reuse: replication and erasure coding
- Transparent upgrades and tiering
- Explore new interface implementations



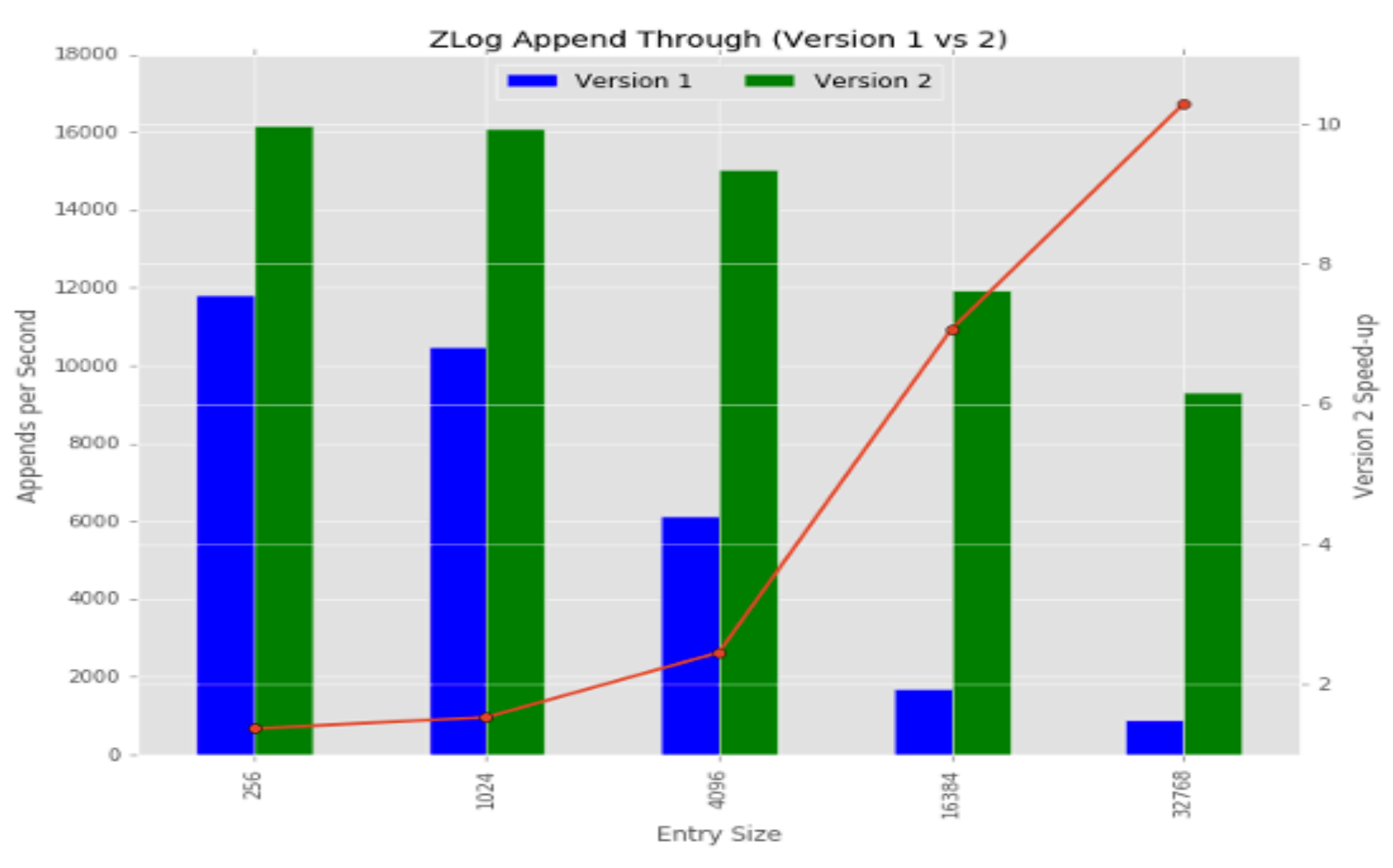
[1] Balakrishnan, et. al, "CORFU: A Shared Log Design for Flash Clusters", NSDI 2012

Large Design State Space

Existing approaches to extensibility rely on hard-coded interfaces and data layouts. A large design space complicates development and upgrade decisions.



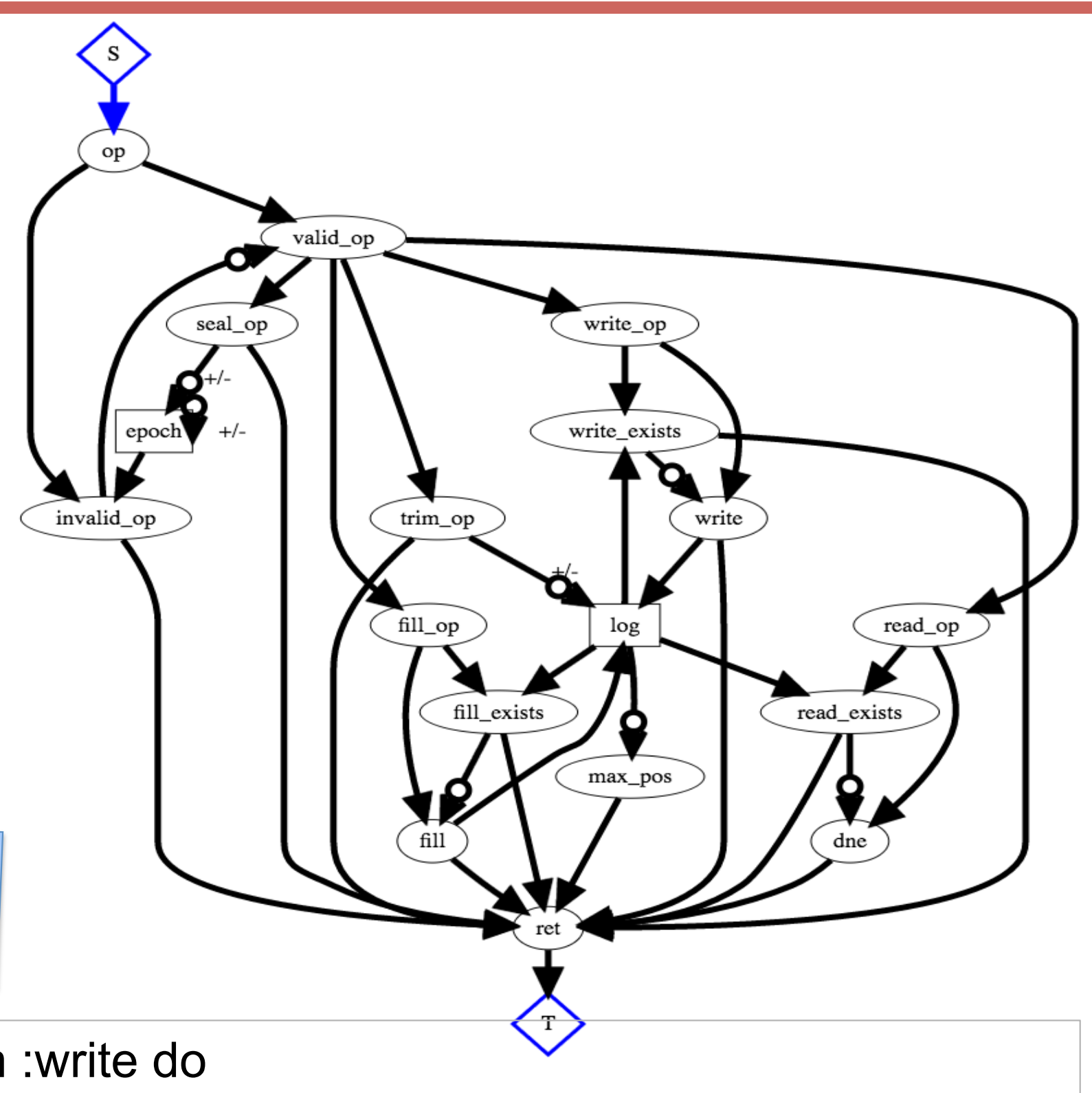
Relative performance difference between two versions of Ceph using different storage strategies. Developer may have selected non-optimal solution in older version.



Two implementations of the same interface may have up to an order of magnitude difference in append performance across log entry sizes. When the size of the design space is large automated techniques to generate physical designs are needed.

Declarative Language

- *Dataflow analysis*
- *Performance statistics from storage system*
- *Optimization*
- *Plan generation*



```
bloom do
  # epoch guard
  invalid_op <= (op * epoch).pairs{|o,e|
    o.epoch <= e.epoch}
  valid_op <= op.notin(invalid_op)
  ret <= invalid_op{|o|
    [o.type, o.pos, o.epoch, 'stale']}

  # op's position found in log
  found_op <= (valid_op * log).lefts(pos => pos)
  notfound_op <= valid_op.notin(found_op)

  # demux on operation type
  write_op <= valid_op {|o| o if o.type == 'write'}
  seal_op  <= valid_op {|o| o if o.type == 'seal'}
end
```

```
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

bloom :seal do
  epoch <- (seal_op * epoch).rights
  epoch <+ seal_op { |o| [o.epoch] }
  temp :maxpos <= log.group([], max(pos))
  ret <= (seal_op * maxpos).pairs do |o, m|
    [o.type, nil, o.epoch, m.content]
  end
end
```

Brados is a declarative language based on Bloom (Alvaro, CIDR '11) that is used to express storage interfaces. Shown above is a snippet of the specification of the CORFU protocol. Optimization techniques are applied to generate an implementation.

This work is partially supported by a CROSS research appointment. For more information about CROSS please visit http://cross.ucsc.edu. The Zlog project an an open-source project published at https://github.com/noahdesu/zlog. You can contact the author Noah Watkins at jayhawk@soe.ucsc.edu.