Storage Abstractions Are Changing

### Traditional Application

### Emerging Applications

**Storage Interfaces**

**Distributed Storage**

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

---

**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

---

**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.

---

**Storage System Programmability in the Wild**

- 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

---

**Declarative Language**

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

---

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.