A case for IO efficiency as a research metric for storage systems

Shuwen Sun,
Isaac Khor, Peter Desnoyers, Orran Krieger

Northeastern University
Boston University

2nd Northeastern University Systems Day (NUSD ’24), Jan 22, 2024

How can we measure and understand the performance of a large-scale storage system?

- Simple metrics for storage devices
  - capacity, IOPS, throughput, and latency
  - These perf. metrics are intrinsic to a hardware device
- No such metric for storage system (database, key/value or object store)
  - Performance is **not** intrinsic
  - But instead is strongly affected by the speed and scale of the storage devices, network, and CPUs of the infrastructure on which it is deployed

The problem

- Storage systems are not storage devices, but we seem to treat them as if they are
    - These measurements are important—most such systems aim to deliver as much performance as possible
- However, hardware-dependent metrics do not paint a complete picture
    - Little insight into the internals of a system
    - Also difficult to compare unless measured on similar hardware

Key insight

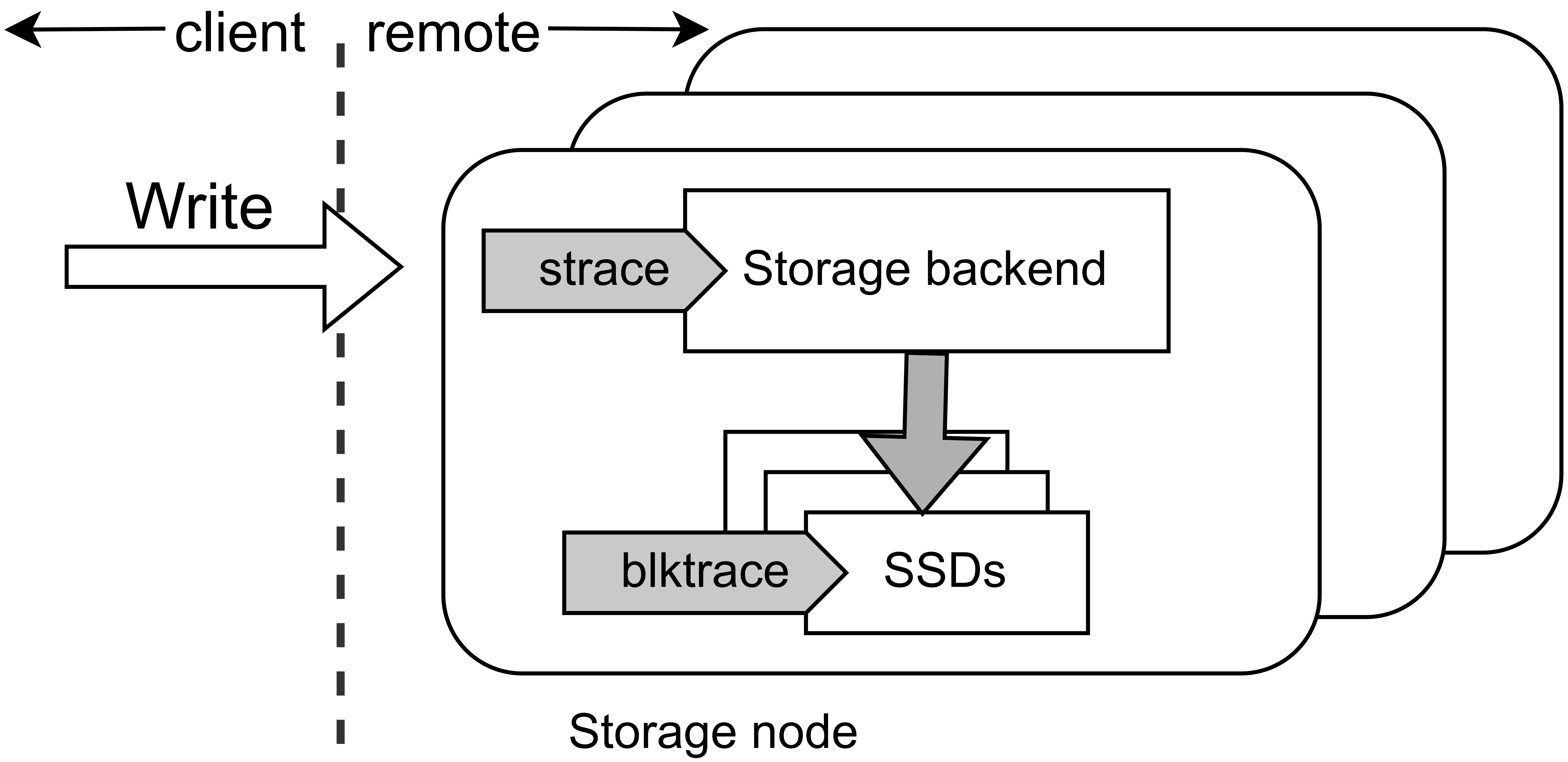
- Augment the standard storage perf. metrics with appropriate **hardware independent** measures of storage system efficiency
    - Such as IO or write amplification: physical/logical written to storage
- IO efficiency is not the only determinant of performance, as an IO-efficient system can have bottlenecks in other areas
- IO efficiency is difficult to predict in today's complex, layered storage systems
    - varied operations for redundancy, consistency, and metadata maintenance
    - layered operations which may expand or merge higher-layer requests

Methodology

- Measure the request efficiency of various storage systems
  - By capturing all backend write requests issued by the system and analyzing them offline
- Isolate the storage backend from other services by configuring virtual disks that are only used by the storage backend
  - record all system calls issued with strace
  - all block-level I/O requests with blktrace

Test bed

- Our testbed has three nodes, all default Ubuntu 22.04 VMs with a separate disk on which the storage systems are configured to use as their backend.



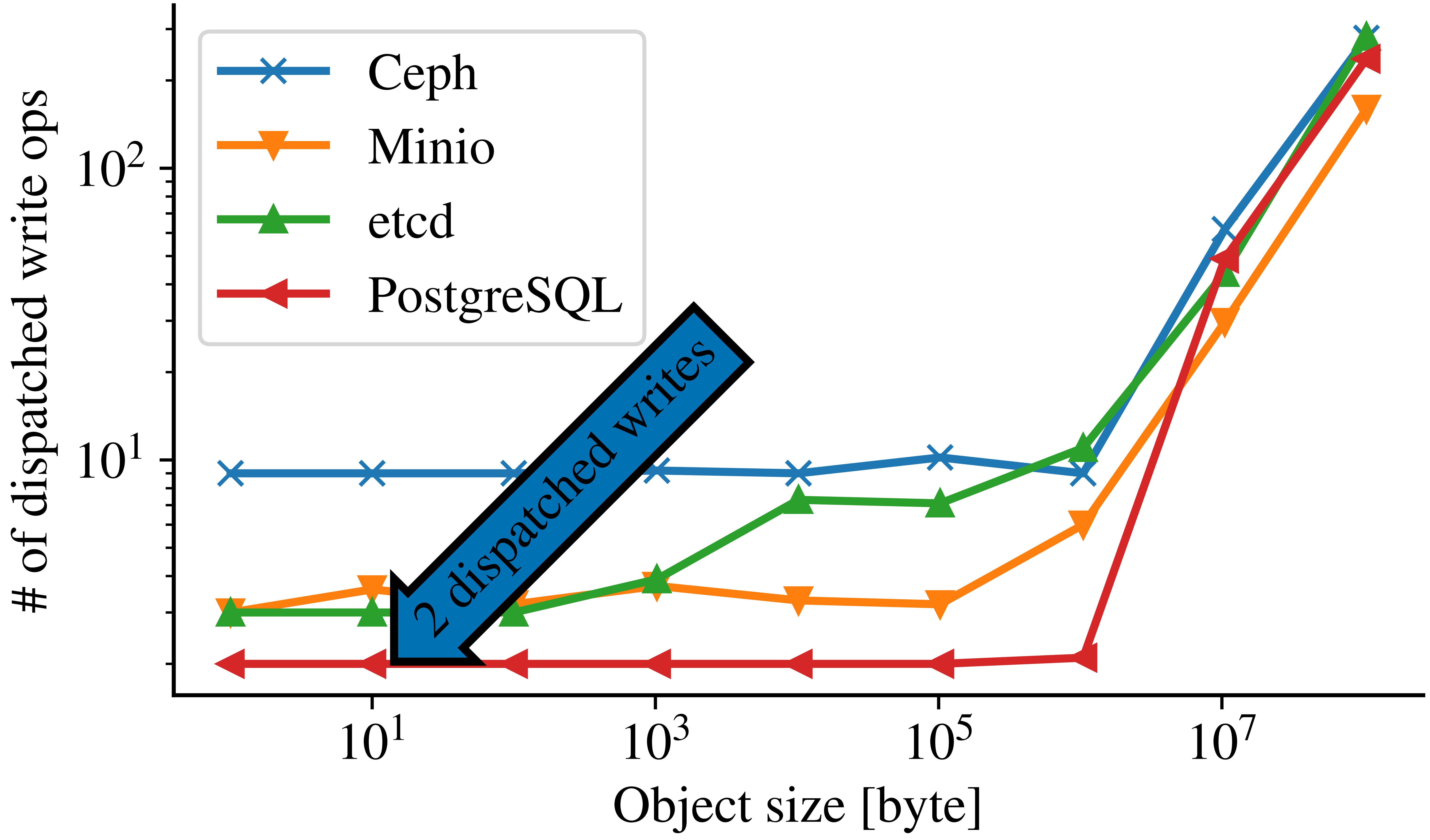
Object store, key-value store, and database studied.

Name	*Type*	*Distributed*
Ceph (BlueStore)	Object store	Triple replication
Minio	Object store	Triple replication
etcd (Raft)	Key-value store	Triple replication
PostgreSQL	Relational Database	Primary with two streaming replicas

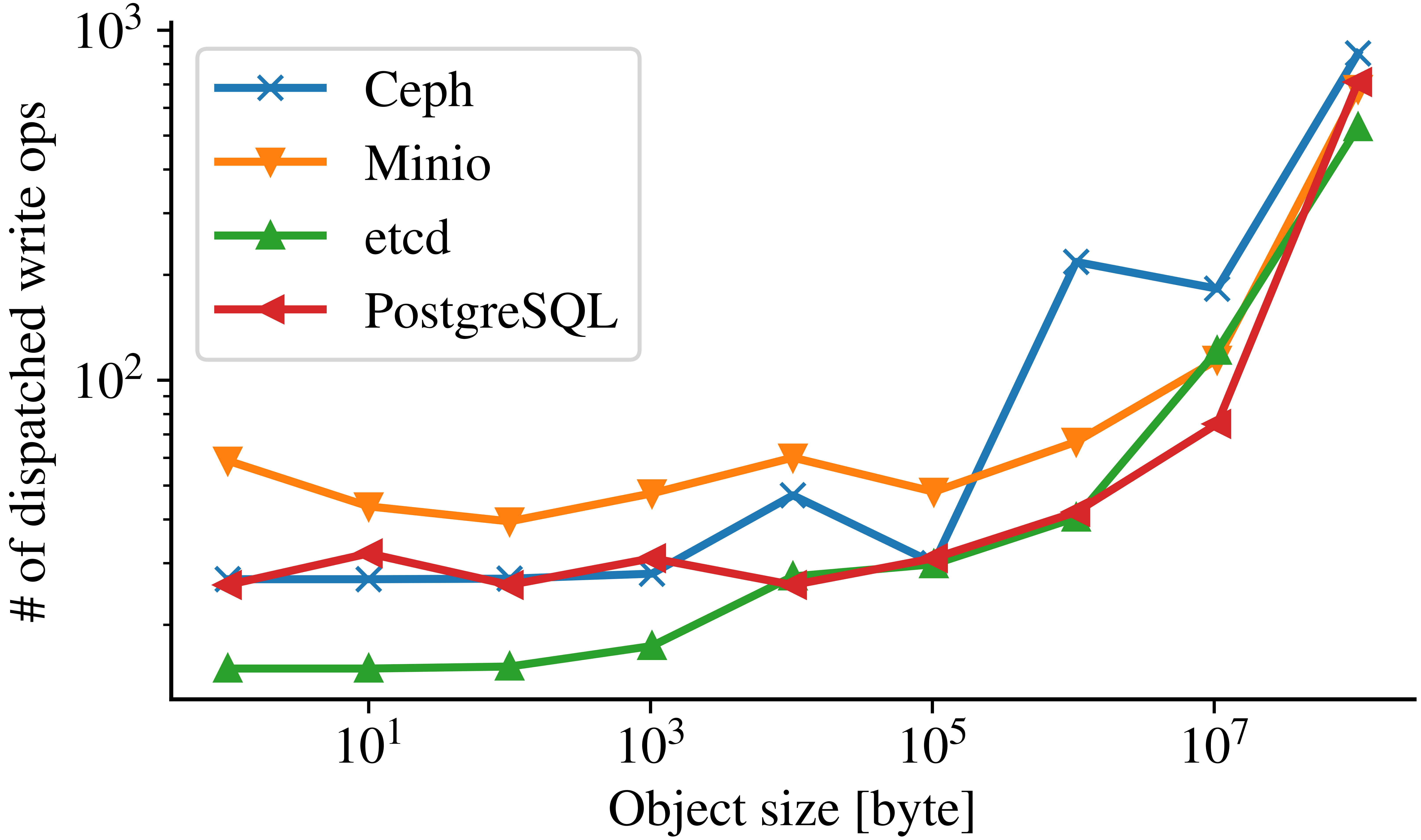
Benchmark operations and workload

Storage	*Operation type*
Object write	
Ceph, Minio	`s3.upload_object(object, bucket, key)`
etcd	`etcd.put(key, object)`
PostgreSQL	`psql: INSERT INTO table (key, object)`
Object delete	
Ceph, Minio	`s3.delete_object(bucket, key)`
etcd	`etcd.delete(key)`
PostgreSQL	`psql: DELETE FROM table WHERE key`
YCSB workload	
Ceph, Minio	`S3 binding in YCSB`
etcd	`Etcd binding in YCSB`
PostgreSQL	`PostgreSQL binding in YCSB`

Local deployment, total number of dispatched write operations observed during uploading a sinlge object. Both axises are in log scale.



Distributed deployment, total number of dispatched write operations observed during uploading a sinlge object.

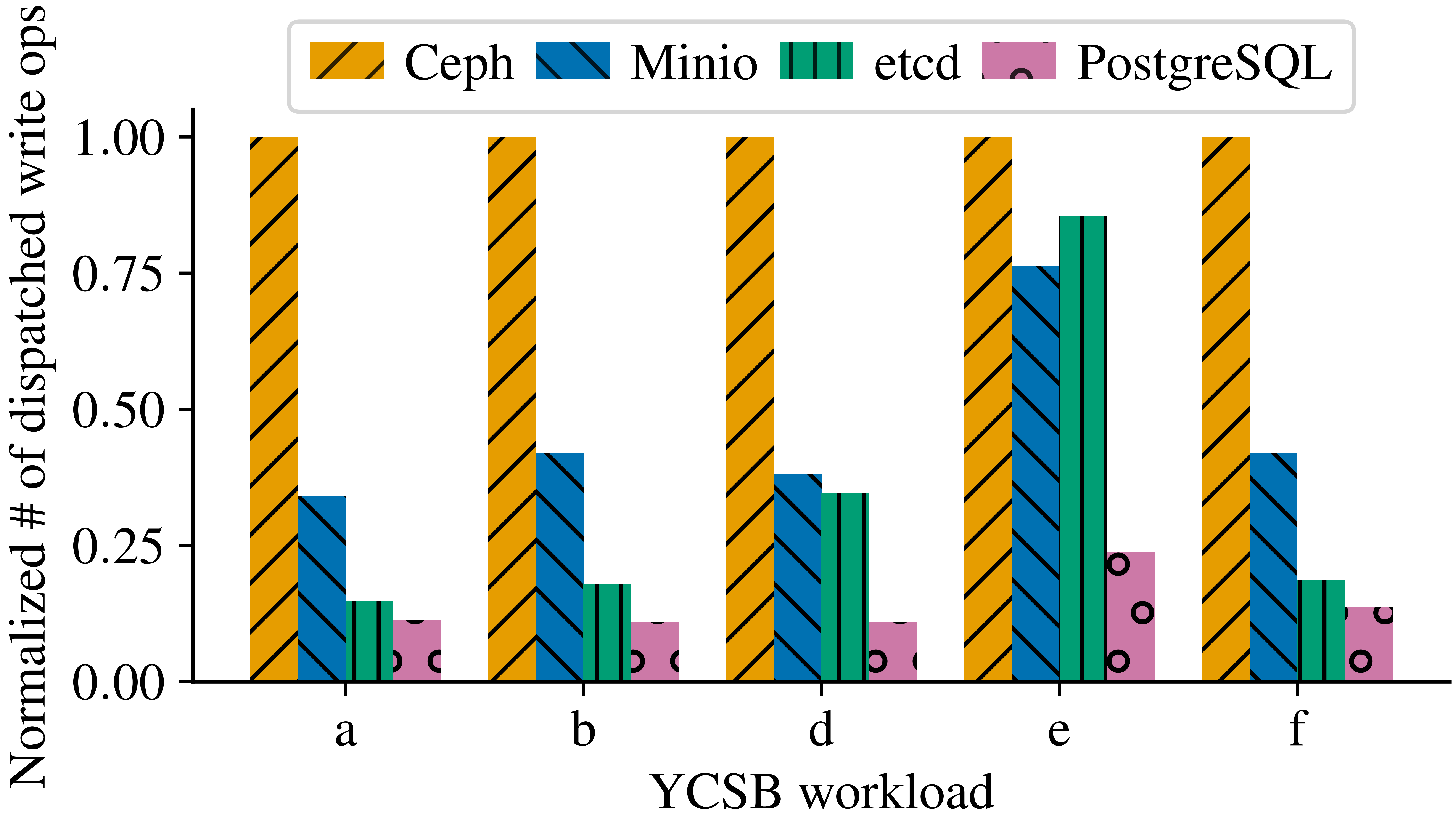


YCSB bindings and workload write operations

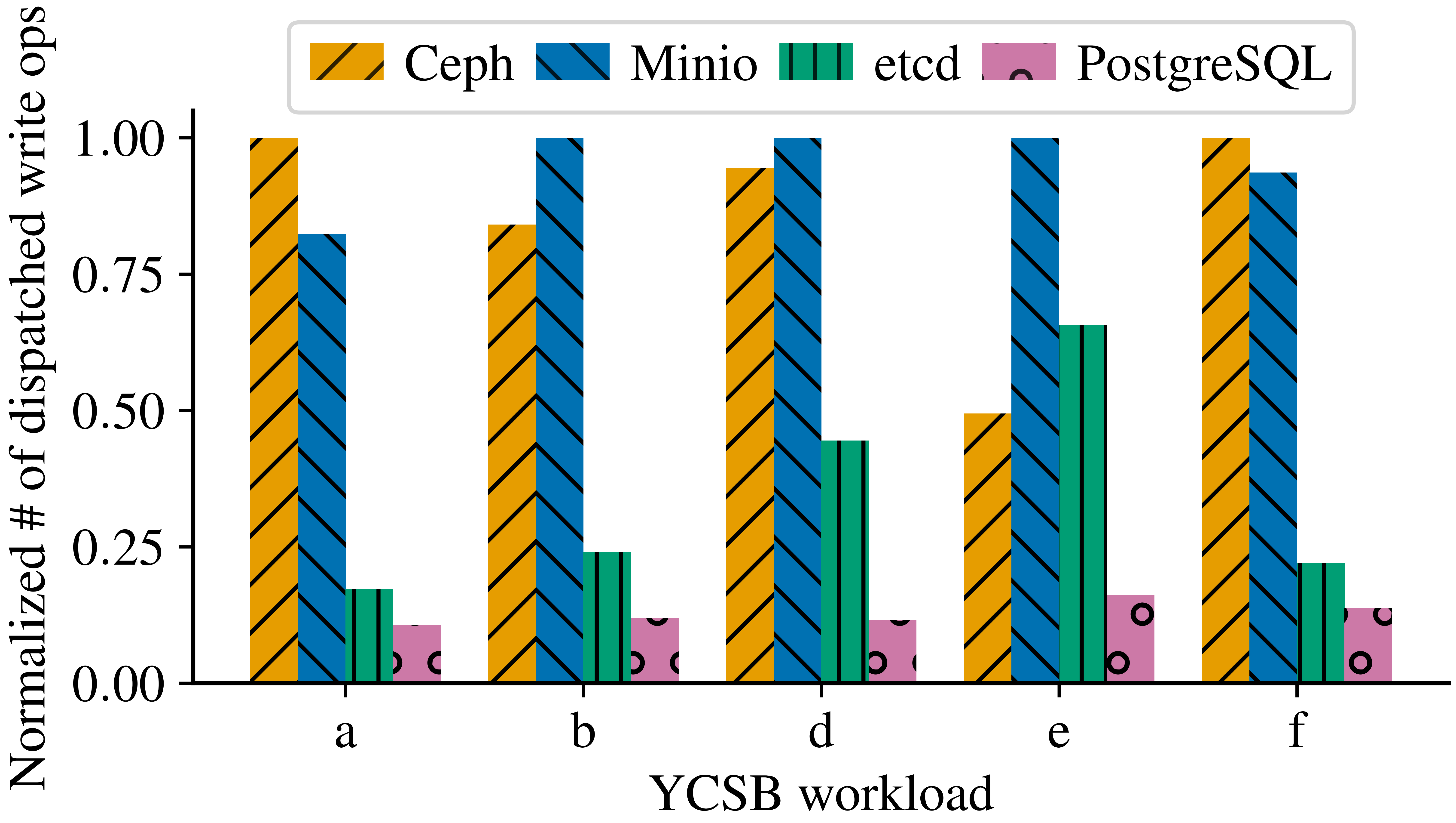
Storage	*YCSB binding*
insert/update	
S3: Ceph, Minio	`bucket, HashMap(key, object)`
etcd	`key, object`
PostgreSQL	`table, HashMap(key, object)`

YCSB workload	*Workload writes*
`a`	50% update
`b`	5% update
`c`	Read only (skipped)
`d`	5% insert
`e`	5% insert
`f`	50% read-modify-write

Local deployment, total number of dispatched write operations observed during YCSB workloads.



Distributed deployment, total number of dispatched write operations observed during YCSB workloads.



Discussion

- Trade-off between WA and capacity amplification
  - Capacity amplification is a measure of how efficiently the file system is using storage
  - Capacity amplification is often bounded, and is bounded by the architecture or design.
  - Because total writes is bounded by the design, it is often easier to predict the amplification factor.
- Limitation and future solution
  - Reliance on usage of blktrace for each operation/workload, and the offline analysis on the blktraces
  - In the future, we can extend with eBPF tracing.

Takeaways

- Today’s storage system performance metric hides details of the underlying system
- However, such a metric does not provide internal information
- This is enough for customers, but not to researchers
- We propose the adopting **write efficiency** as a more research oriented metric for storage system.
- We show that efficiency metric can be used to contract different storage systems.
- We show that with such a metric, researchers can gain more in-depth knowledge of the storage system, which is otherwise hard to obtain with performance metric.