Towards 0-Latency Durability

Sang-Won Lee
(swlee@skku.edu)

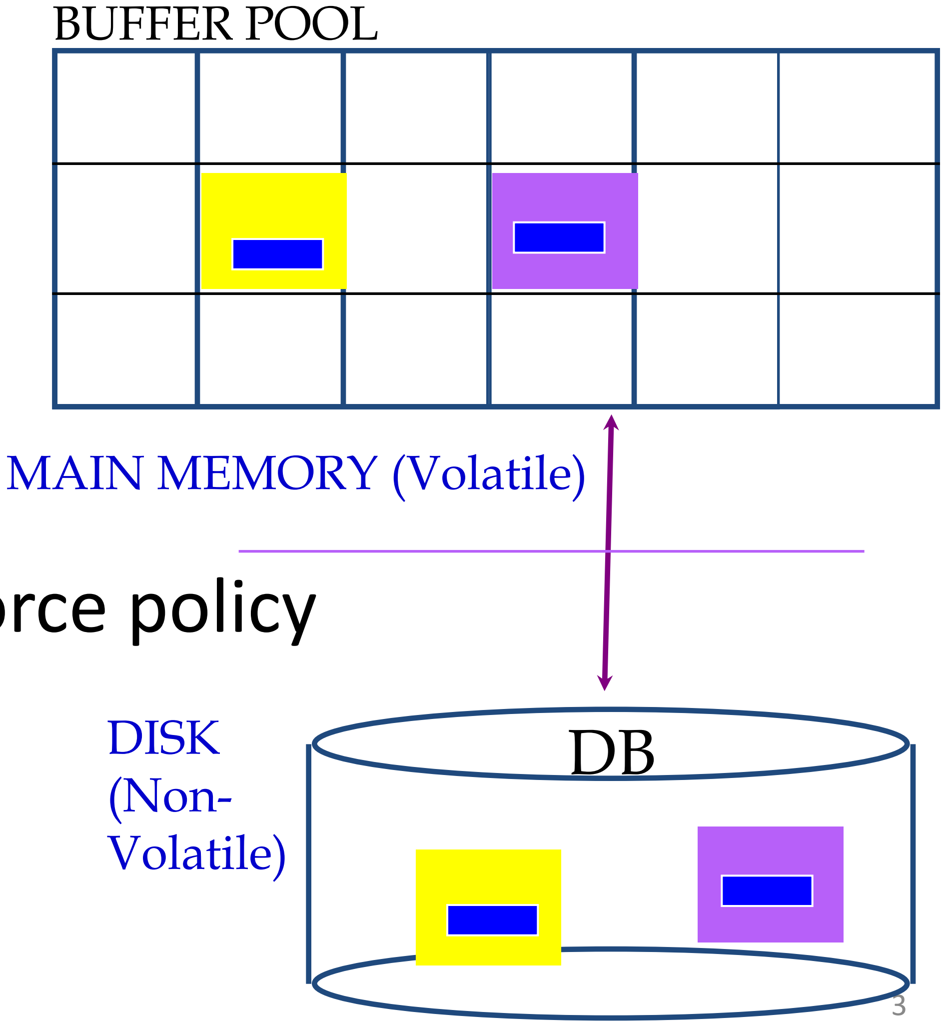
Ack.: Moon, Yang, Oh and SKKU VLDB Lab. Members

NVRAMOS 2014

NVRAM is for 0-latency Durability

(DB) Transaction and ACID

- E.g. 100$ transfer from A to B account
- ACID
  - Atomicity
  - Consistency
  - Isolation
  - Durability
- Durability latency in force policy
  - 20ms @ HDD
  - < 1ms @ SSD
  - 0-latency @ NVDRAM

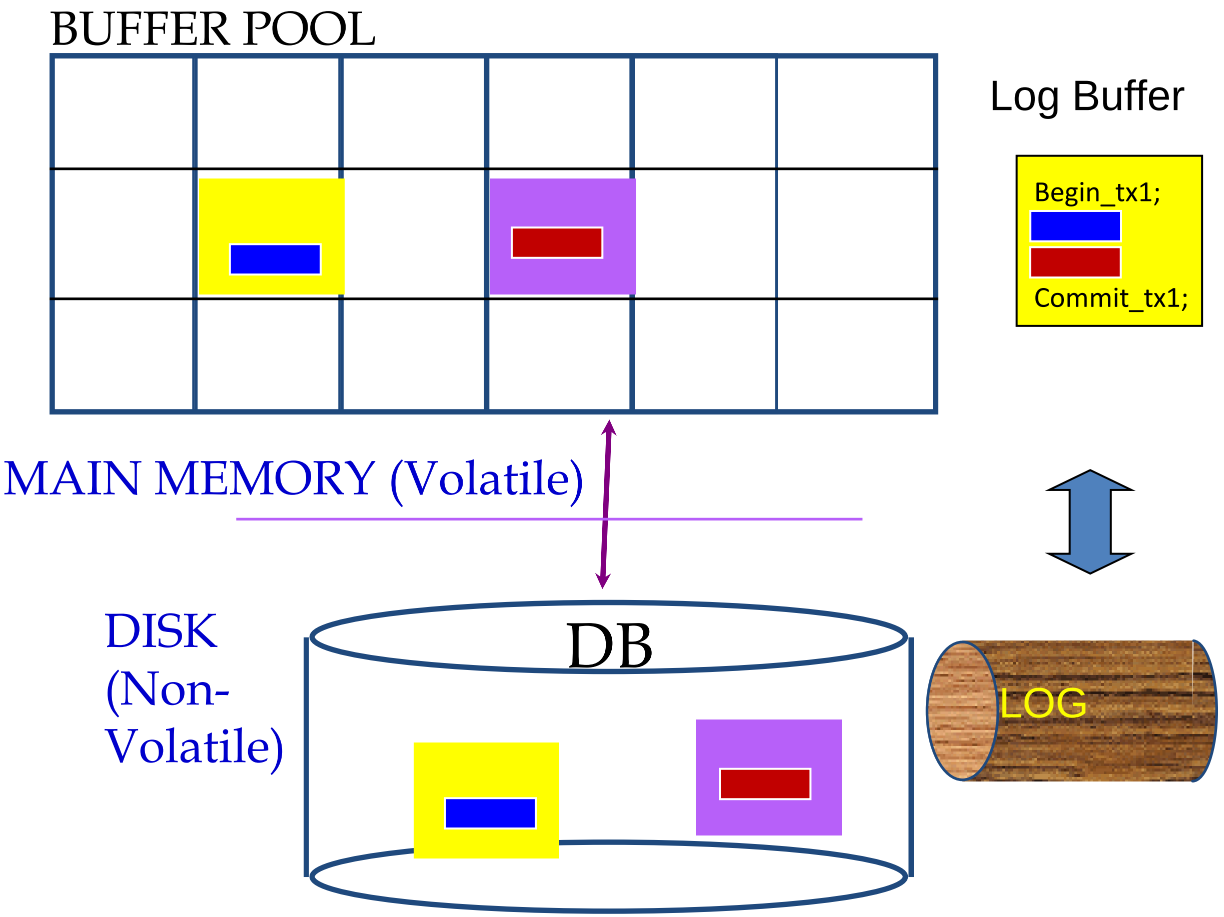


Transaction and ACID

- Durability latency in force policy
  - Atomicity devil
    - Redundant write is inevitable: {RBJ, WAL}@SQLite, Metadata Journaling@FS , DWB@MySQL, FPW@Postgres, …
    - Thus, worse latency
  - 0-latency @ NVDRAM??
    - What about UNDO for atomicity?

WAL for Durability and Atomicity

- Durability latency in WAL Log
  - 2ms @ HDD
  - 0.2ms @ SSD
  - 0-latency @ NVDRAM??



Durable and Ordered Write in Transactional Database

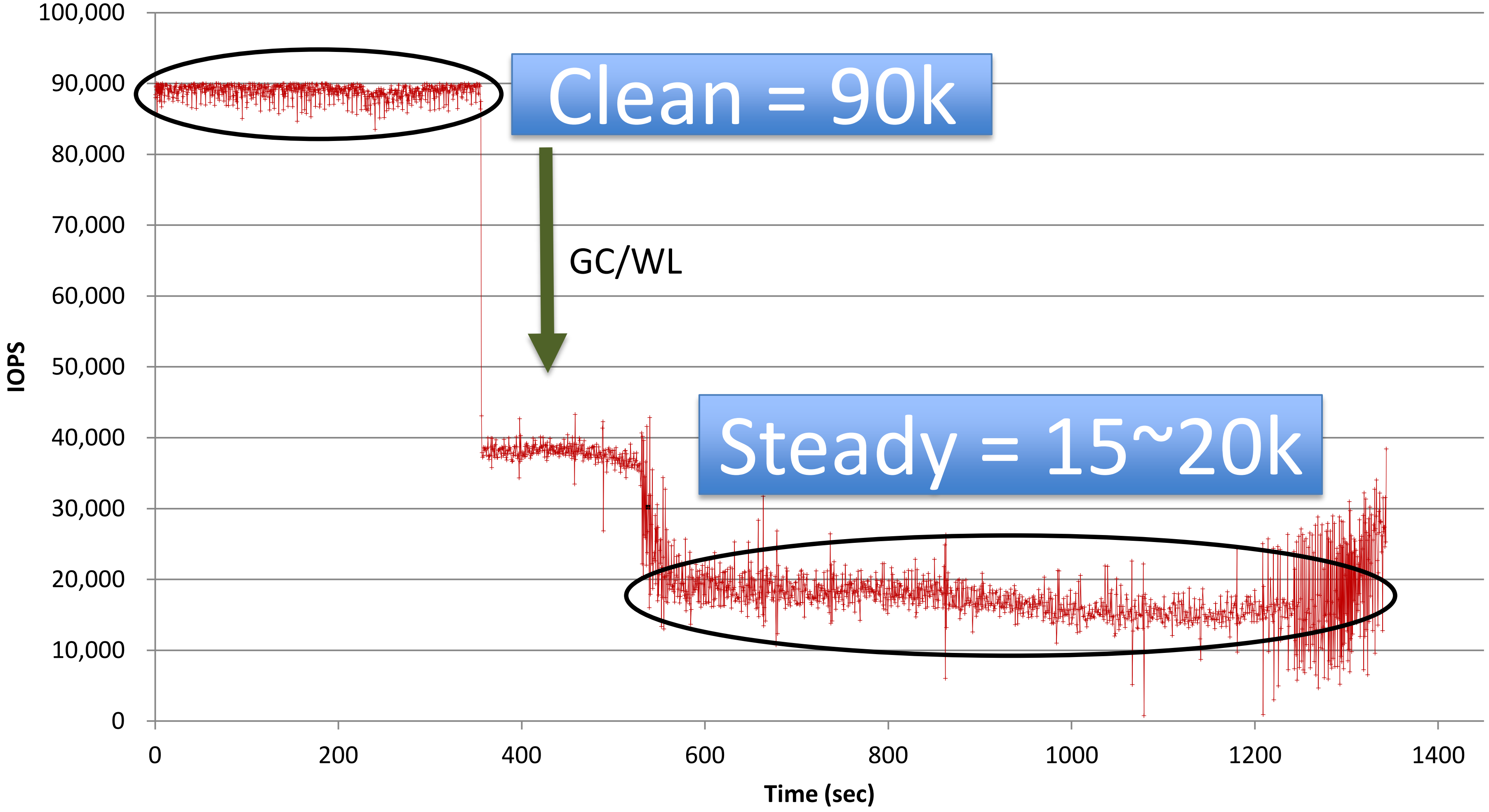
- In addition to ACID property of logical transaction level, a few properties of IO are critical for transactional database.
  - Page write should be durable and atomic
  - In some case, ordering between two writes should be preserved

Contents

- DuraSSD [SIGMOD2014]
- Latency in WAL log
  - WAL paradigm is ubiquitous!!!
  - DuraSSD vs. Ideal Case in TPC-B
  - DuraSSD vs. Ideal Case in NoSQL YCSB
- Future directions

Native SSD Performance

- Random write performance
  - `$> fio 4KB_random_write`



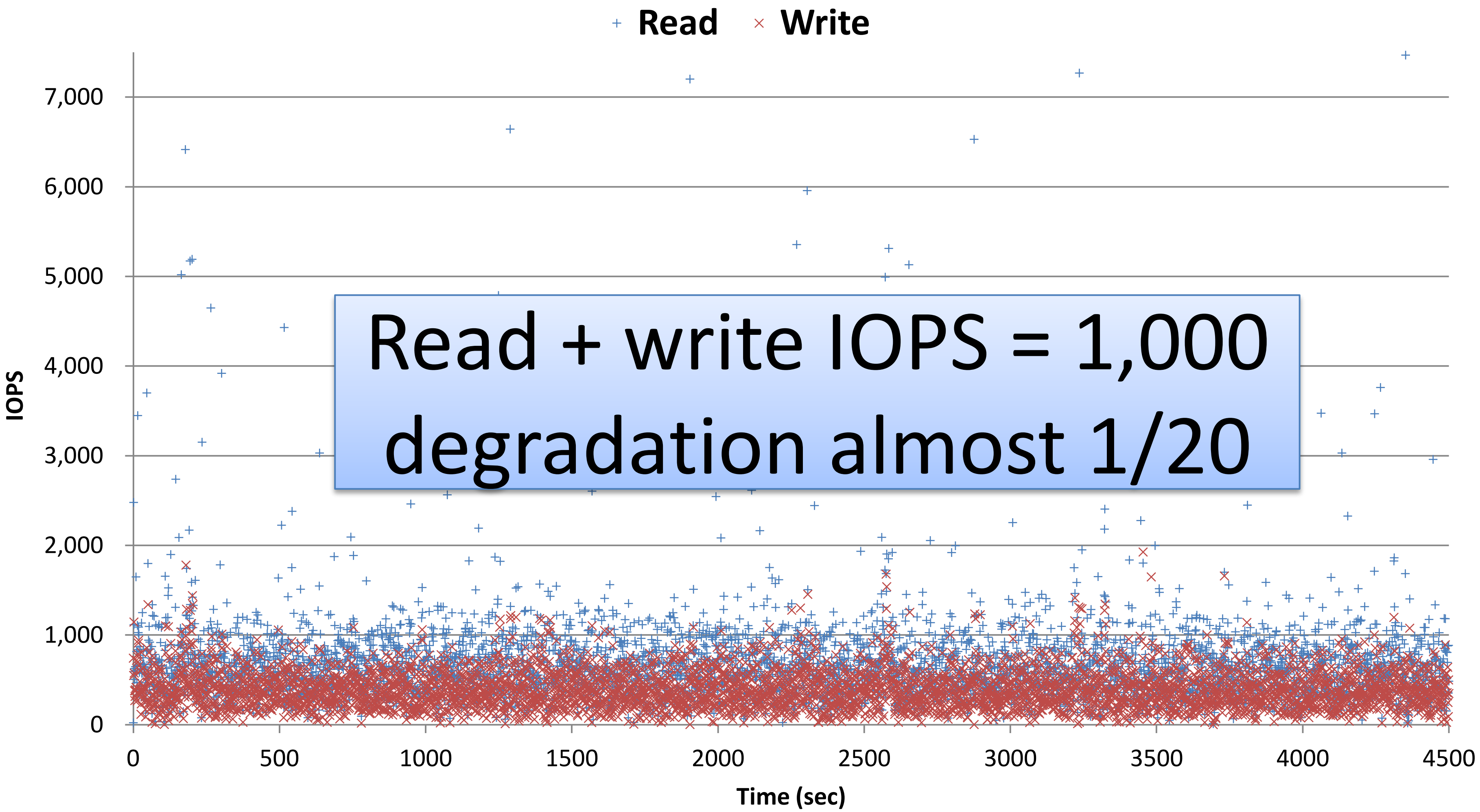
Clean = 90k

GC/WL

Steady = 15~20k

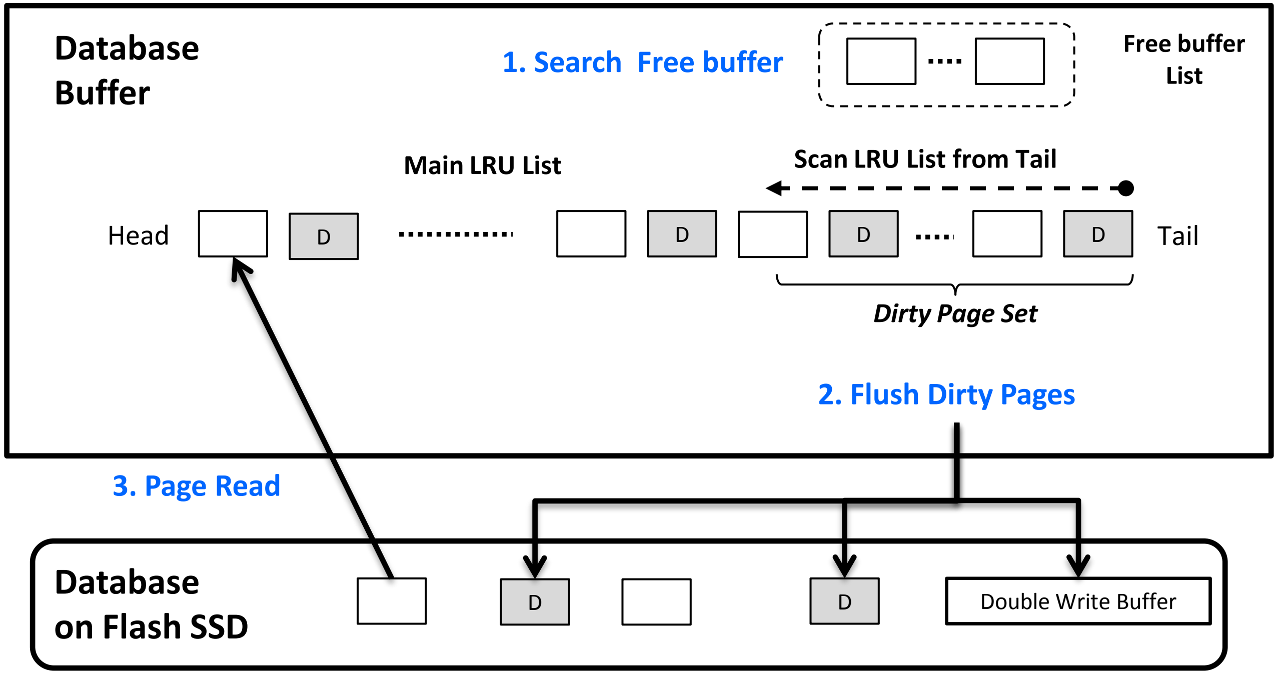
SSD Performance with MySQL

- Running MySQL on top of SSD
  - `$> run LinkBench - MySQL`



Read + write IOPS = 1,000 degradation almost 1/20

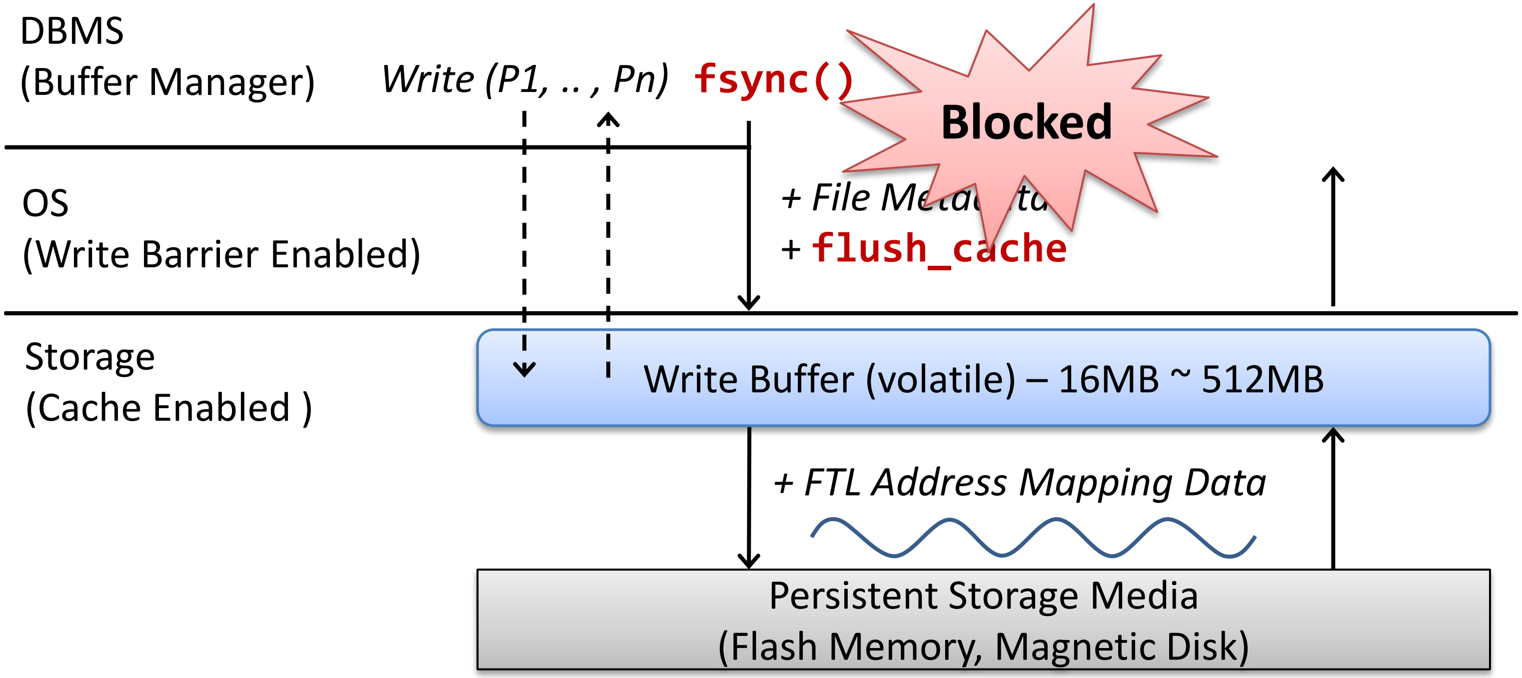
MySQL/InnoDB I/O Scenario



Issue	Technique	Problem
Latency	Buffer pool	Read is blocked until dirty pages are written to storage
Atomicity	Redundant writes	One to double write buffer, the other to data pages
Durability	Write barrier	Flush dirty pages from OS to device and then from write cache to media

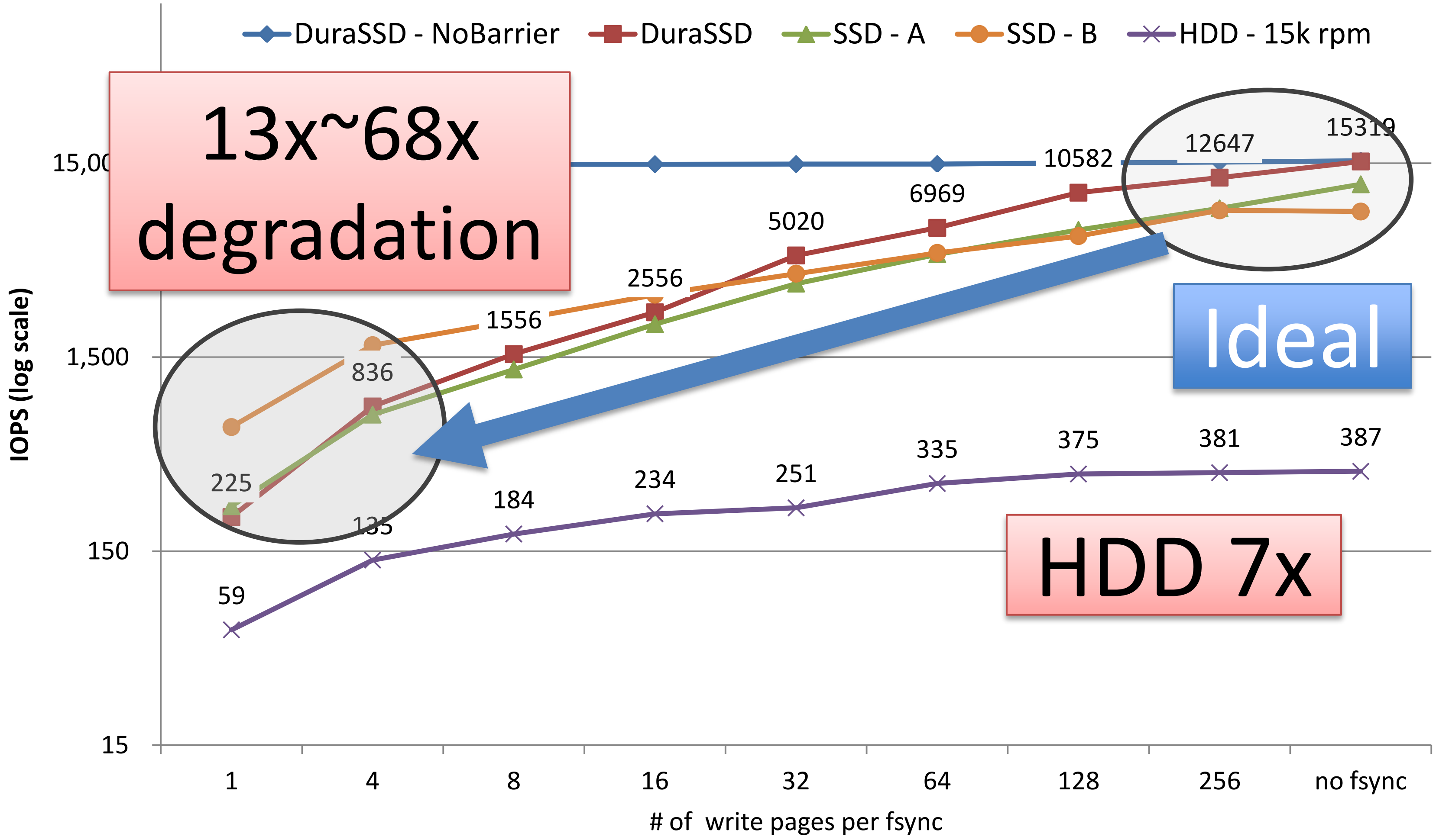
Persistency by WRITE_BARRIER

- fsync() - “ordering and durability”
  - Flushes dirty pages from OS to device
  - If WRITE_BARRIER is enabled, OS sends a FLUSH_CACHE command to storage device and flushes the write cache to persistent media:



Impact of fsync with Barrier

- High performance degradation due to fsync
  - **SSD - 70x** ↓ **HDD – 7x** ↓



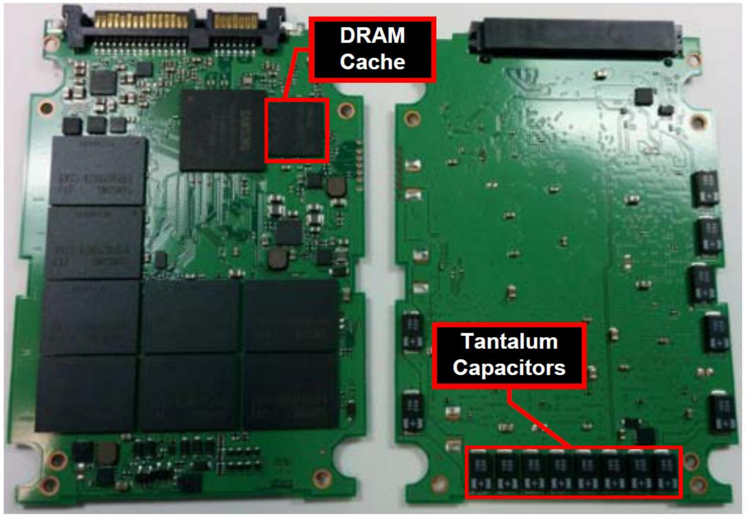
13x~68x degradation

Ideal

HDD 7x

DuraSSD

- DuraSSD
  - Samsung SM843T with a durable write cache
  - Economical solution
    - DRAM cache backed by tantalum capacitors
    - HDD with battery-backed



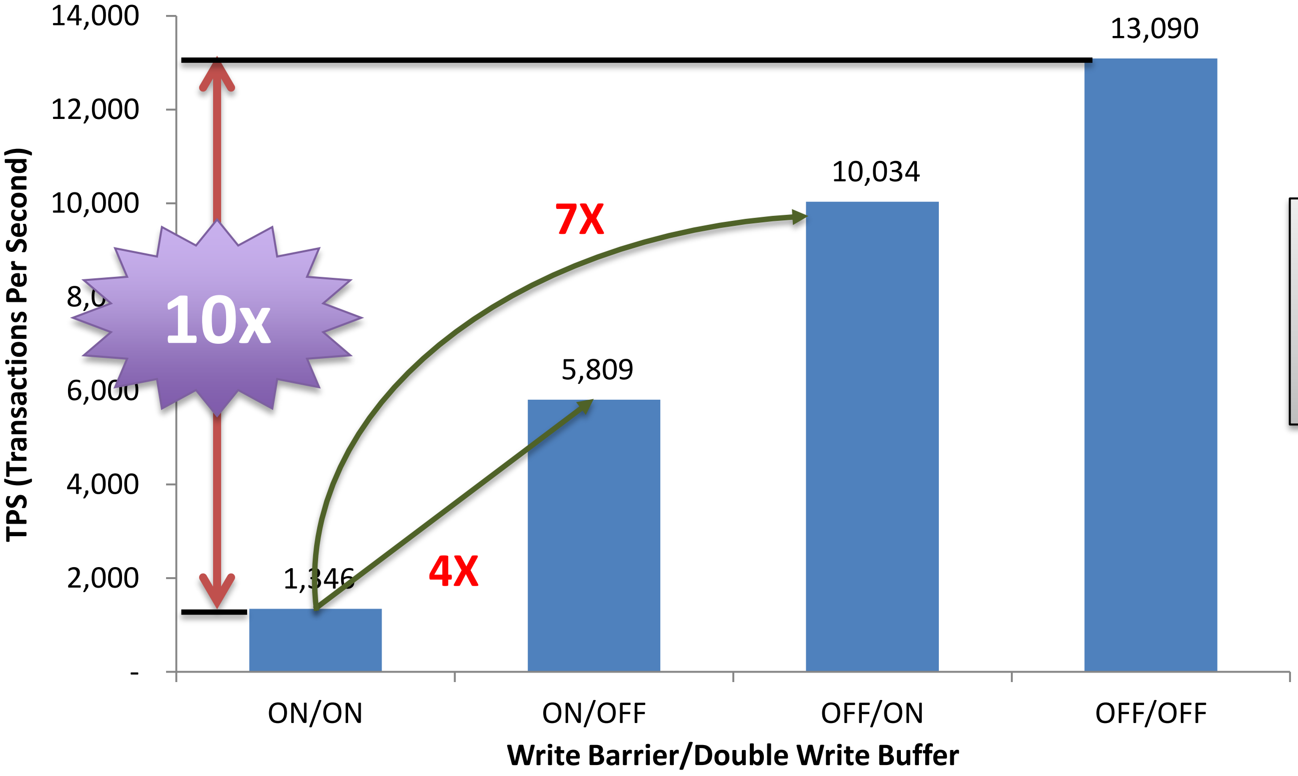
Issue	Existing Technique	Solution
Latency	Buffer pool	Fast write with a write cache
Atomicity	Redundant writes	Single atomic write for small pages (4KB or 8KB)
Durability	Write barrier	• Durability: battery-backed write cache without WRITE_BARRIER • Ordering: NOOP scheduler and in-order command queue

Experiment Setup

- System configuration
  - Linux Kernel 3.5.10
  - Intel Xeon E5-4620 * 4 sockets (64 cores/with HT)
  - DDR3 DRAM 384GB (96GB/Socket)
  - Two Samsung 843T 480GB DuraSSDs (data and log)
- Workloads
  - LinkBench
    - Social network graph data benchmark (MySQL)
  - TPC-C
    - OLTP workload (Oracle DBMS)
  - YCSB
    - Key-Value store NoSQL (Couchbase)
    - Workload A

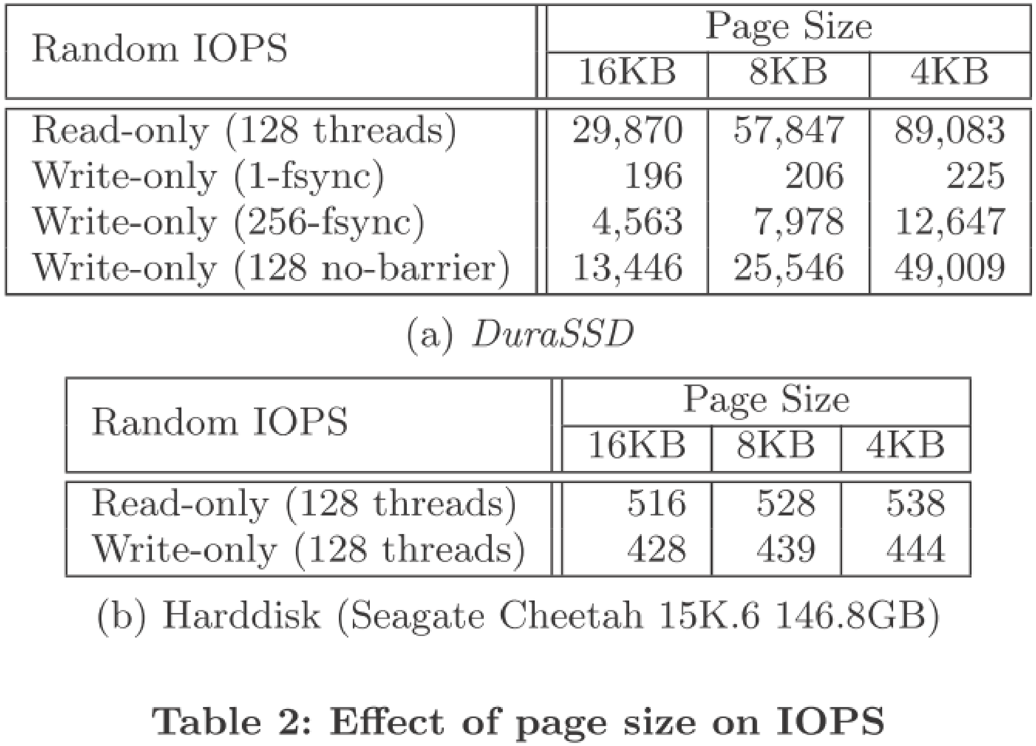
LinkBench: Storage Options

- Impacts of double write and WRITE_BARRIER
  - 100GB DB, 128 clients
  - 6.4 Million transactions (50K TXS per client)



Pagesize 16KB
Buffer 10GB
DB 100GB
Clients 128

Page Size Tuning



Random IOPS	Page Size		
	16KB	8KB	4KB
Read-only (128 threads)	29,870	57,847	89,083
Write-only (1-fsync)	196	206	225
Write-only (256-fsync)	4,563	7,978	12,647
Write-only (128 no-barrier)	13,446	25,546	49,009

(a) *DuraSSD*

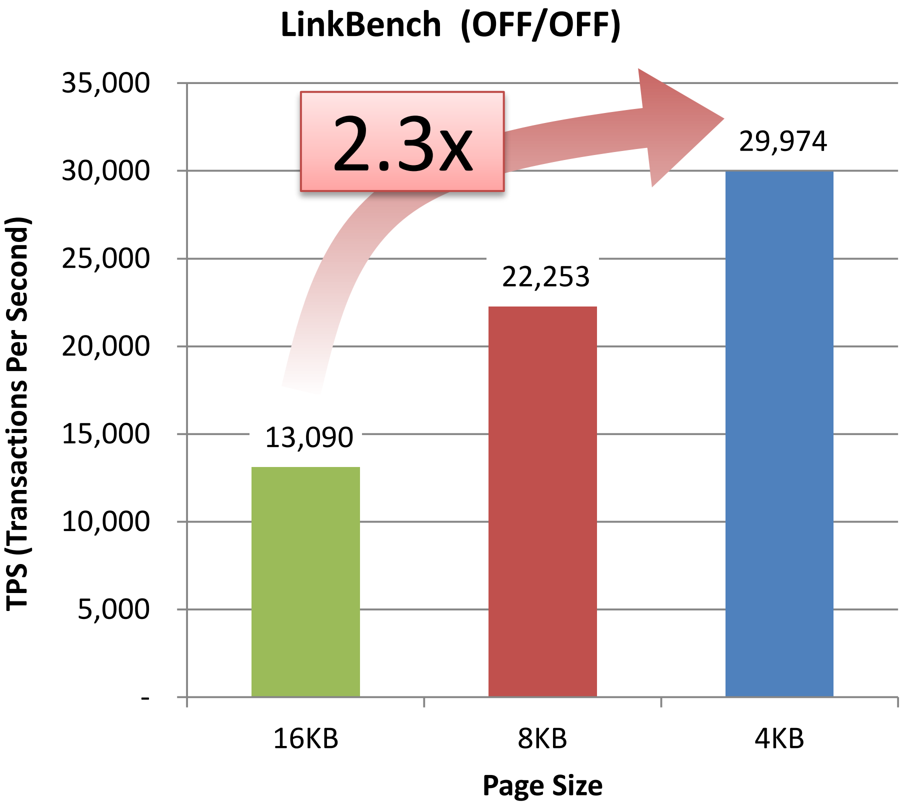
Random IOPS	Page Size		
	16KB	8KB	4KB
Read-only (128 threads)	516	528	538
Write-only (128 threads)	428	439	444

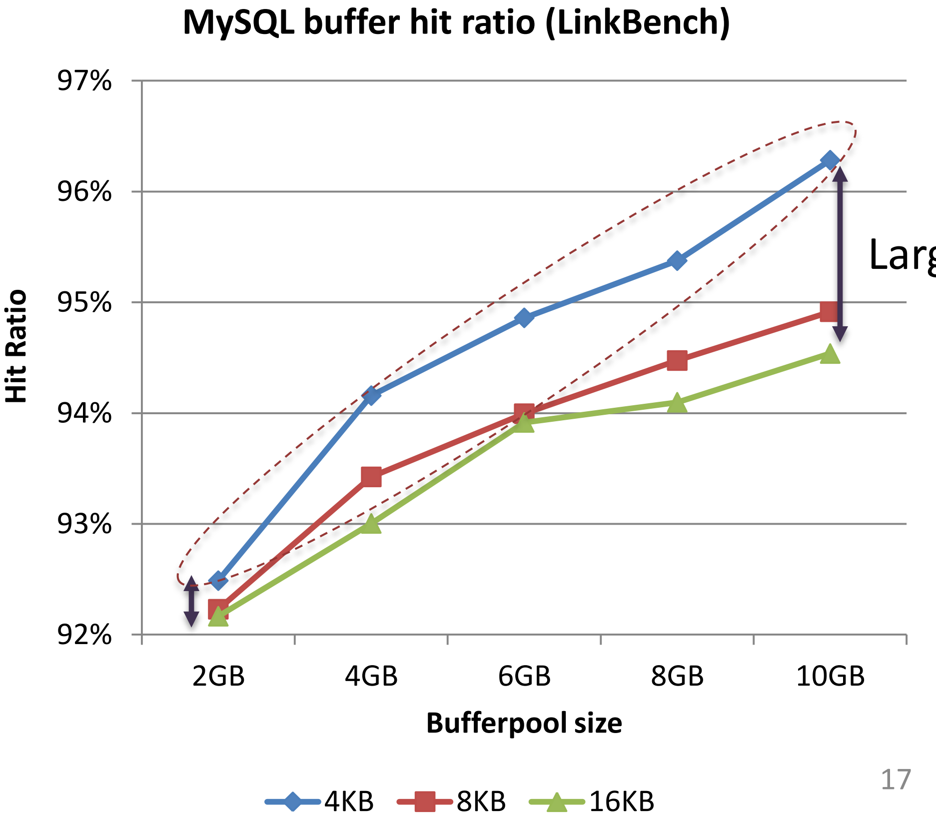
(b) Harddisk (Seagate Cheetah 15K.6 146.8GB)

Table 2: Effect of page size on IOPS

LinkBench: Page Size

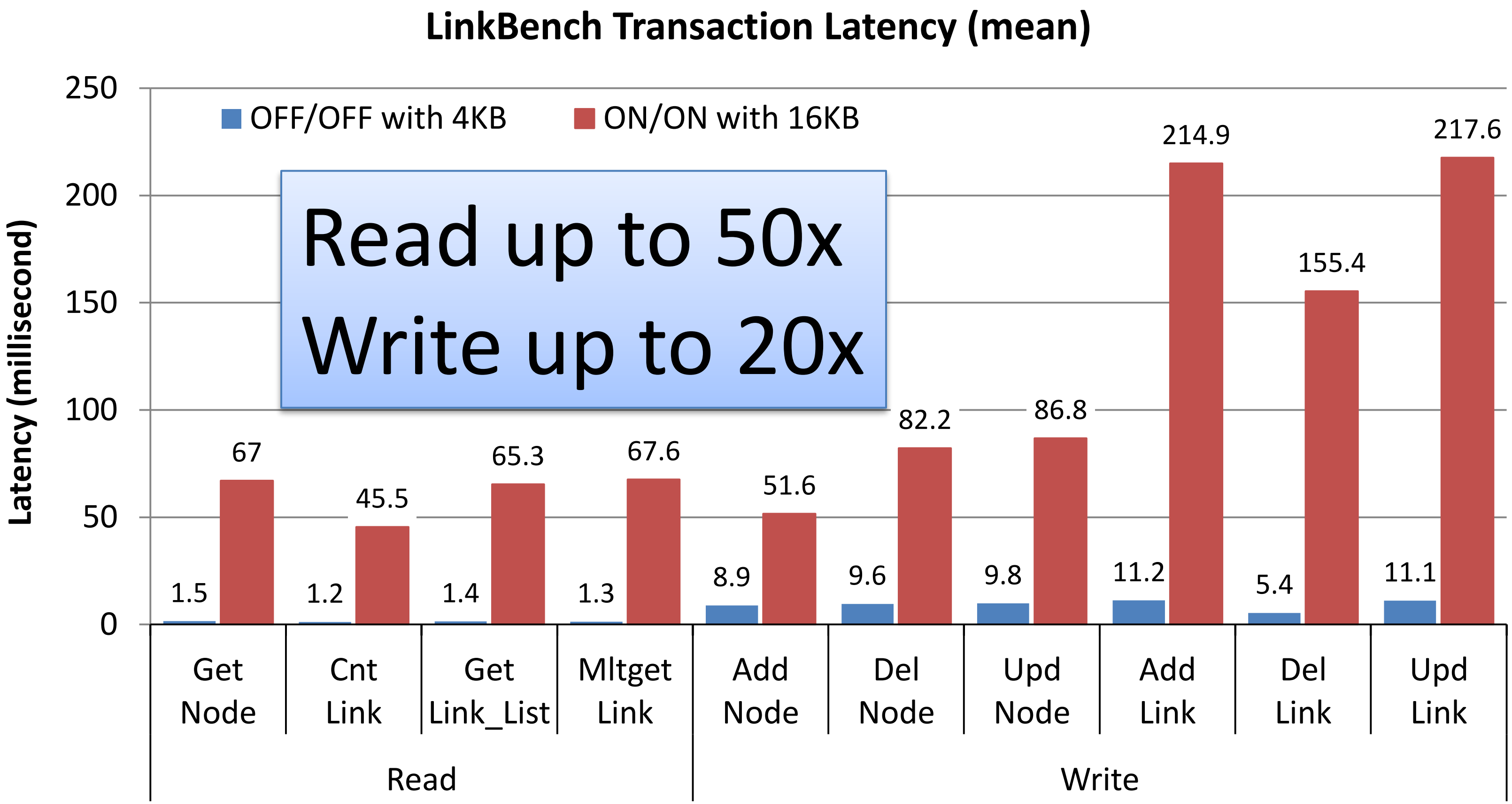
- Benefits of small page
  - Better read/write IOPS
    - Exploit internal parallelism
  - Better buffer-pool hit ratio
  - vs. [SIGMOD09] – no write opt.→ less effect of page size tuning





LinkBench: All Options Combined

- Transaction latency
  - – Write optimization → Better read latency



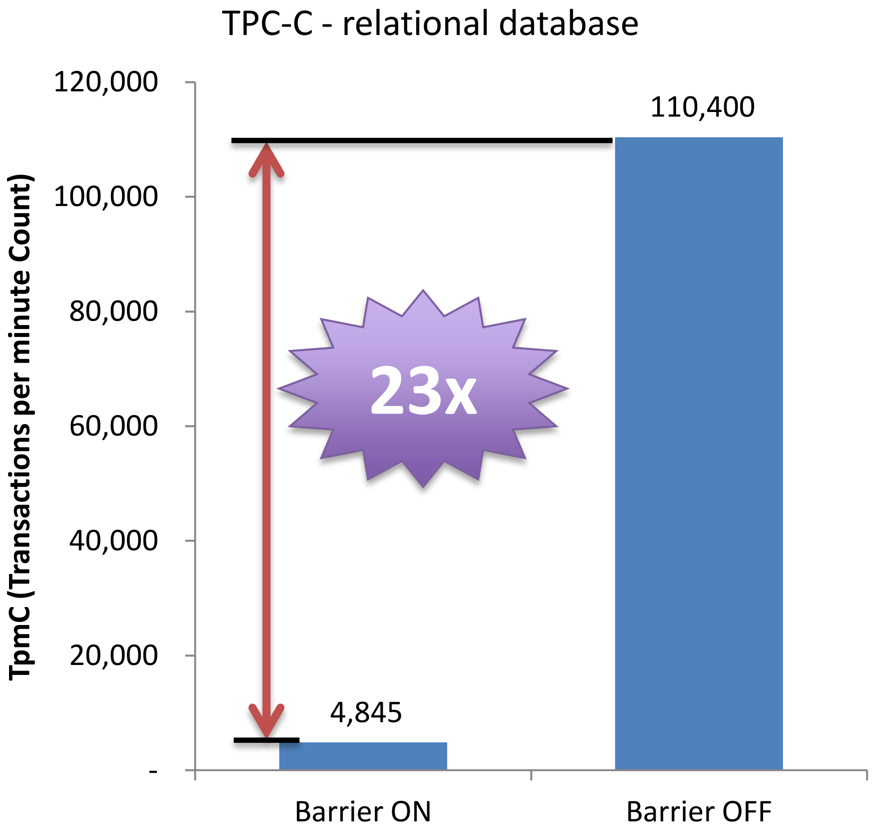
LinkBench Transaction Latency (mean)

Read up to 50x

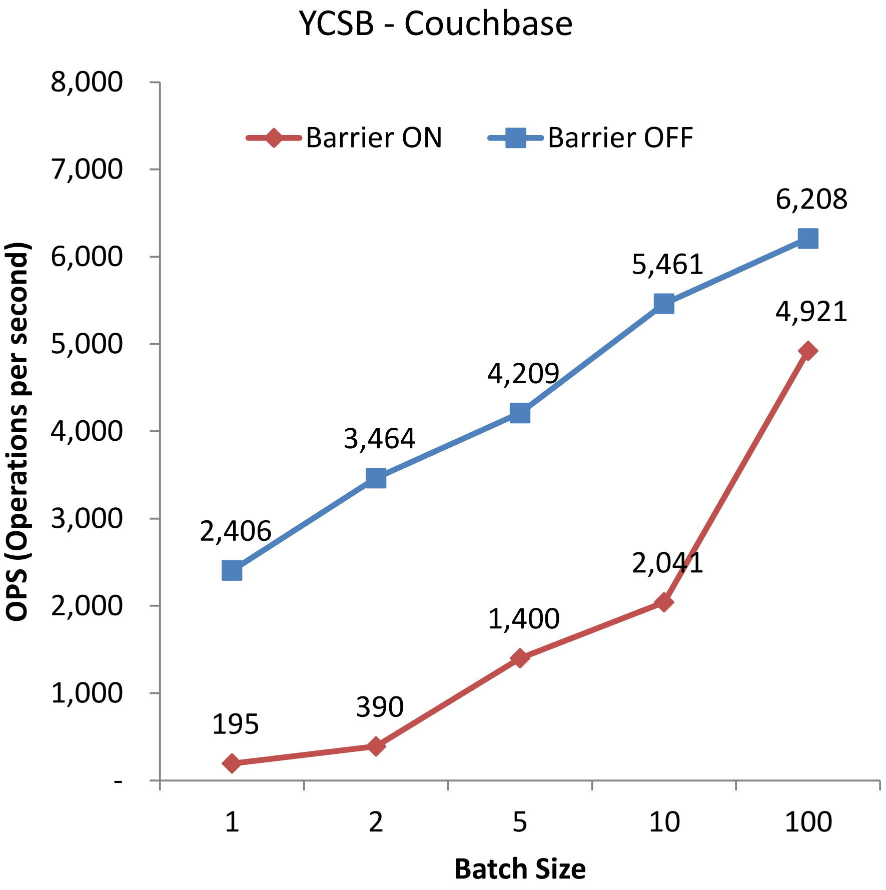
Write up to 20x

Database Benchmark

- TPC-C for MySQL: up to **23x**
- YCSB for CouchDB : up to **10x**



Pagesize 8KB/Buffer 2GB/DB 100GB



Conclusions

- DuraSSD
  - SSD with a battery-backed write cache
    - 10$ → 20~30X performance improvement
  - Guarantees atomicity and durability of small pages
- Benefits
  - Avoids redundant writes of database for atomicity
  - Implements durability without costly fsync operations
  - Utilizes internal parallelism of SSDs with buffering
  - Exploits the potential of SSD
    - 10~20 times performance improvement
    - Prolonged device lifetime

Conclusions

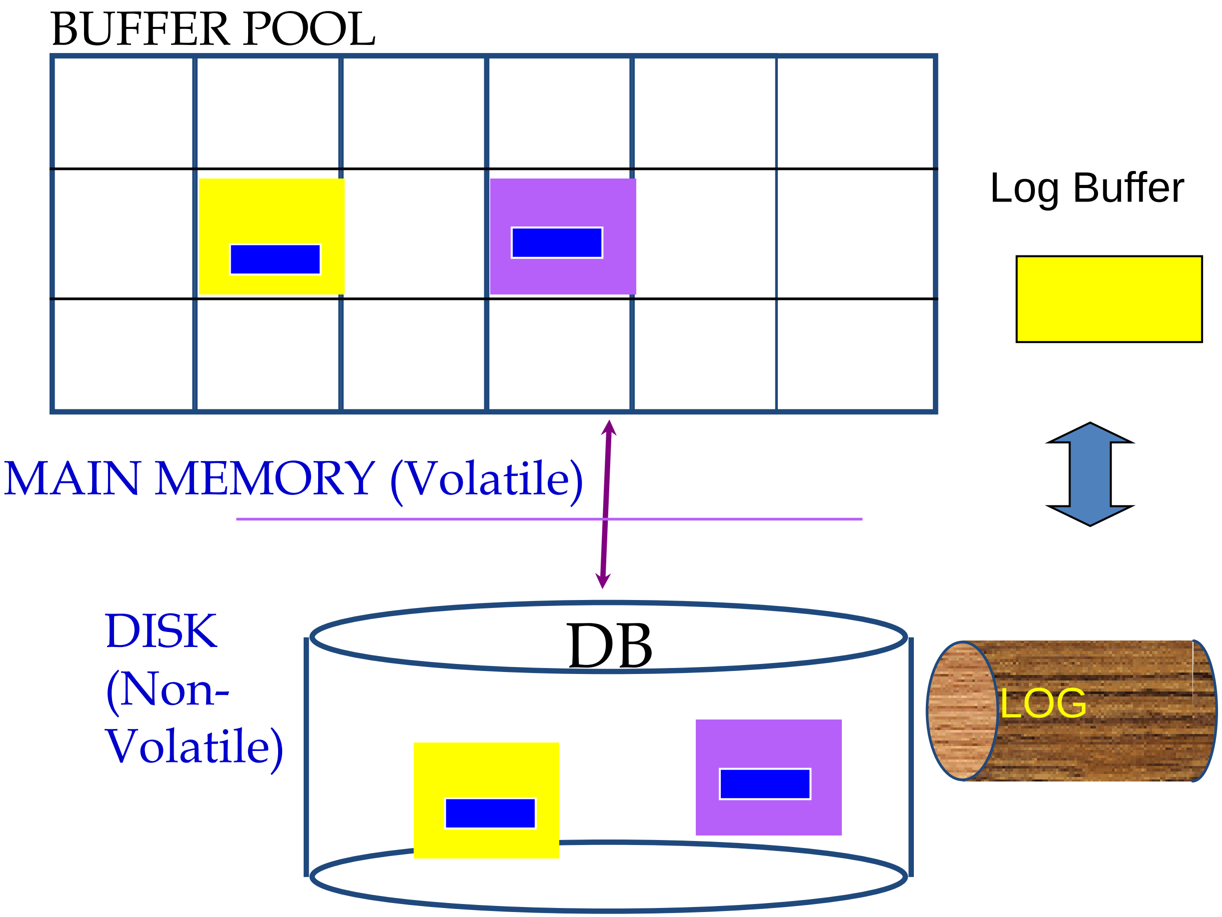
- DuraCache in DuraSSD
  - Gap filler between the latency for the durability and the bandwidth
- One DuraSSD can saturate Dell 32 core machine (when running LinkBench)
  - IOPS crisis is solved?
  - NVMe = Excessive IOPS/GB ?
- MMDBMS vs. All-flash DBMS: Who wins?
  - 5 min rule (Jim Gray)
    - 3hr rule with hdd @ 2014 → MMDBMS
    - 10 sec rule with NVMe @ 2014 → All-flash DBMS with less DRAM

Contents

- DuraSSD
- Latency in WAL log
  - WAL paradigm is ubiquitous!!!
  - DuraSSD vs. Ideal Case in TPC-B
  - DuraSSD vs. Ideal Case in NoSQL YCSB
- Future directions

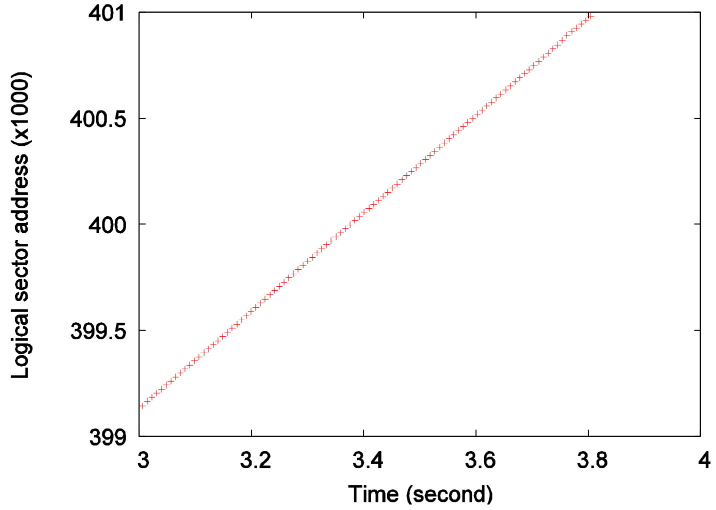
Ubiquitous WAL Paradigm

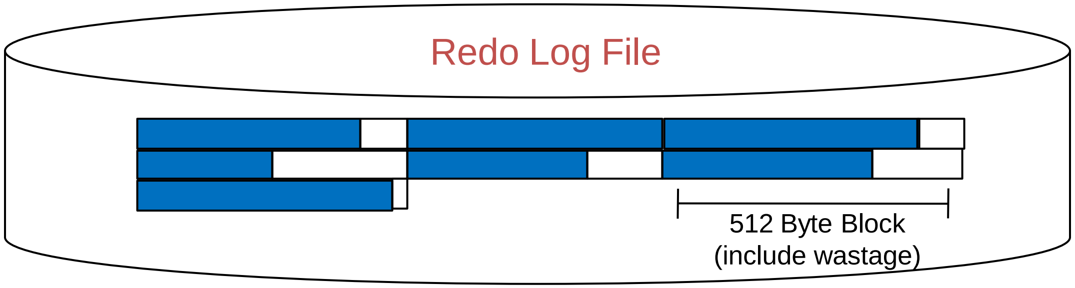
- OLTP DB
- NoSQL and KV Store
  - WAL log in BigTable, MongoDB, Cassandra, Amazon Dynamo, Netflix Blitz4j, Yahoo WALNUT, Facebook, Twitter
- Distributed Database
  - Two Phase Commit
  - SAP HANA, Hekaton
- Distributed System
  - Eventual consistency
  - Replication



Ubiquitous WAL Paradigm

- Append-only write pattern

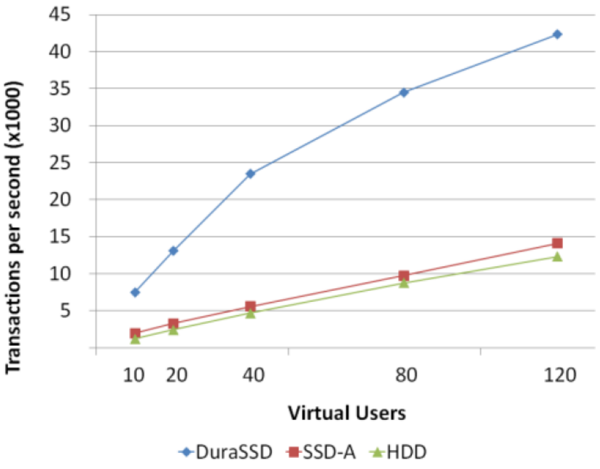


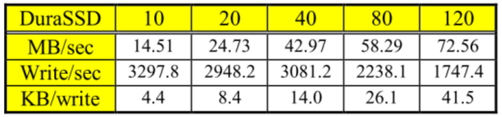


- Trade-off b/w performance and durability
  - DBMS, NoSQL: sync vs. async commit mode

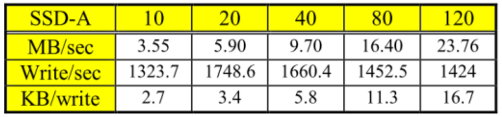
TPC-B: Various WAL Devices

- Intel Xeon E7-4850
  - 40 cores: 4 sockets, 10 cores/socket, 2GHz/core
  - 32GB 1333MHz DDR3 DRAM
- 15K rpm HDD vs. MLC SSD vs. DuraSSD

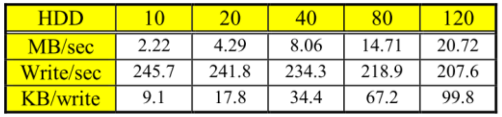




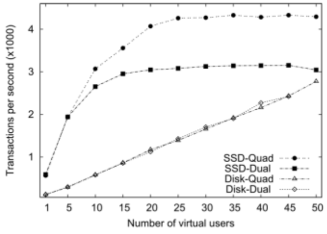
DuraSSD	10	20	40	80	120
MB/sec	14.51	24.73	42.97	58.29	72.56
Write/sec	3297.8	2948.2	3081.2	2238.1	1747.4
KB/write	4.4	8.4	14.0	26.1	41.5



SSD-A	10	20	40	80	120
MB/sec	3.55	5.90	9.70	16.40	23.76
Write/sec	1323.7	1748.6	1660.4	1452.5	1424
KB/write	2.7	3.4	5.8	11.3	16.7

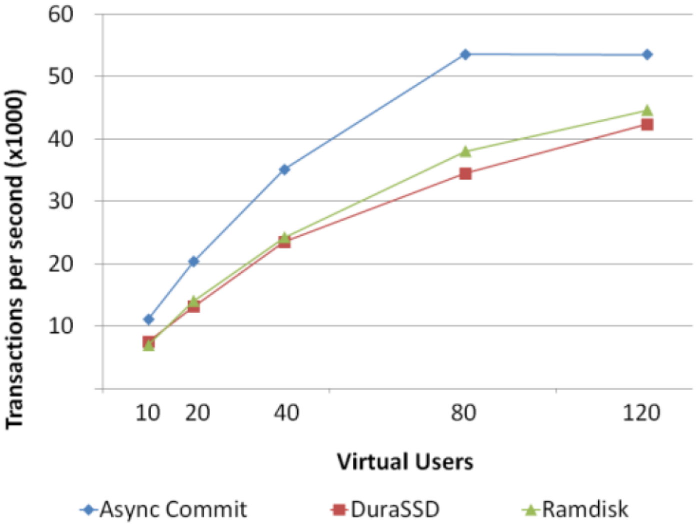


HDD	10	20	40	80	120
MB/sec	2.22	4.29	8.06	14.71	20.72
Write/sec	245.7	241.8	234.3	218.9	207.6
KB/write	9.1	17.8	34.4	67.2	99.8



TPC-B: Various WAL Devices

- Async Commit vs. RamDisk vs. DuraSSD



- Polling vs. Interrupt

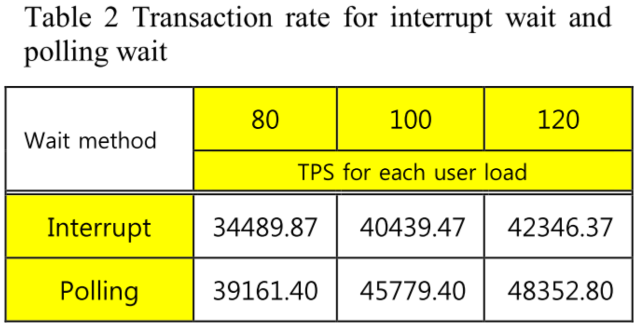
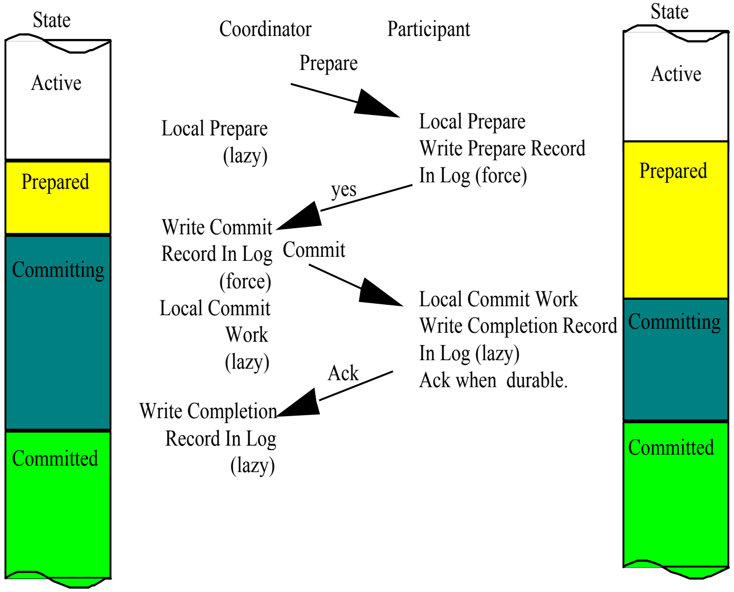


Table 2 Transaction rate for interrupt wait and polling wait

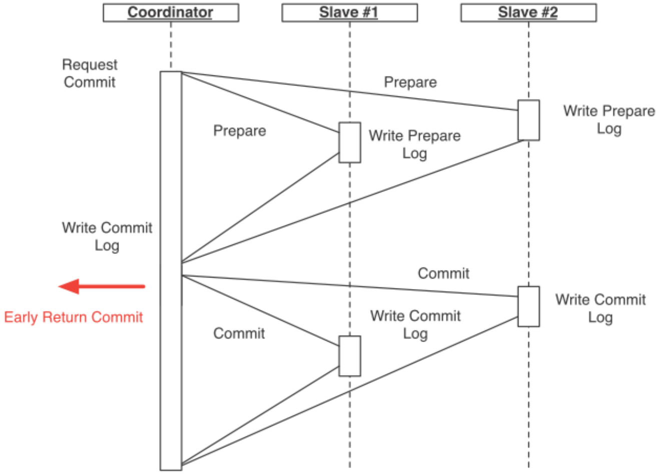
Wait method	80	100	120
	TPS for each user load		
Interrupt	34489.87	40439.47	42346.37
Polling	39161.40	45779.40	48352.80

Distributed Main Memory DBMS

- Two-phase commit in distributed DBMSs

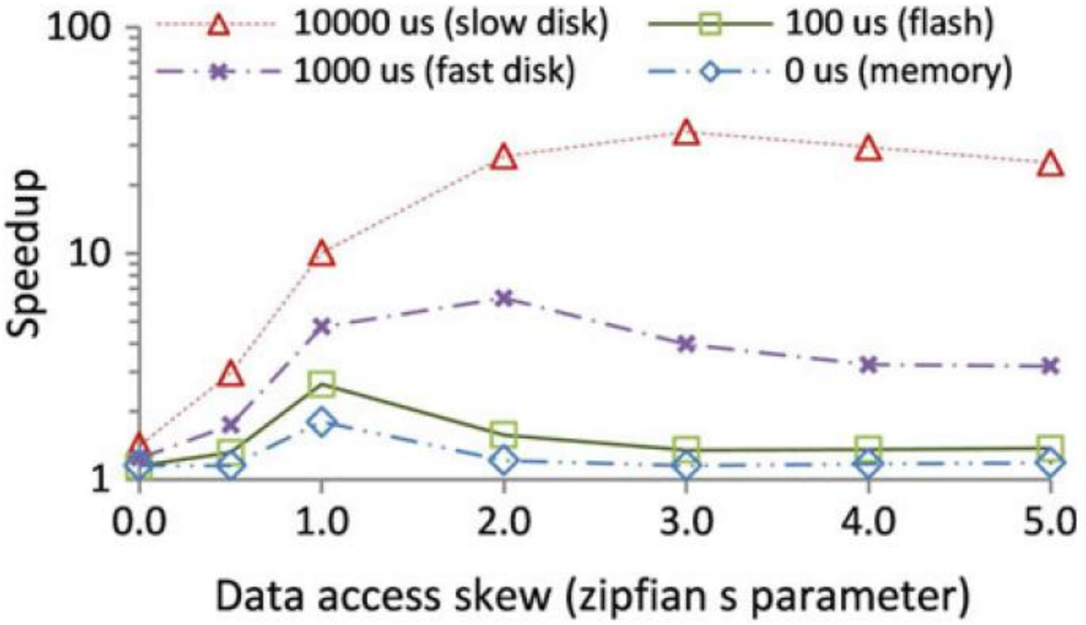


- "High Performance Transaction Processing in SAP HANA", IEEE DE Bulletine, 2013 June



The Effect of Fast Durability on Concurrency in DBMS

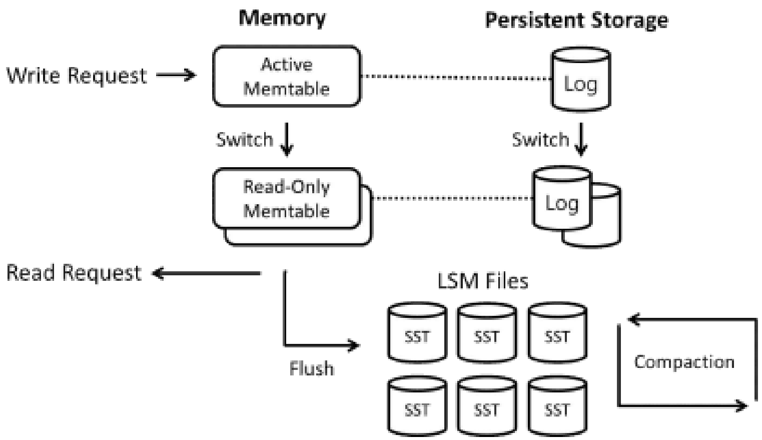
- Other TXs are waiting for the lock held by a committing TX



- Source: Aether [VLDB 2011, VLDB J. 2013]

YCSB@RocksDB

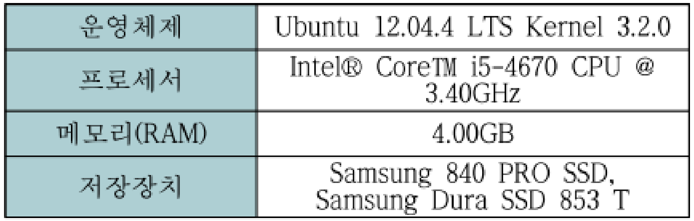
- Random update against 1M KV documents
  - Each document: 10B key + 800B value



[그림 1] RocksDB 기본 구조

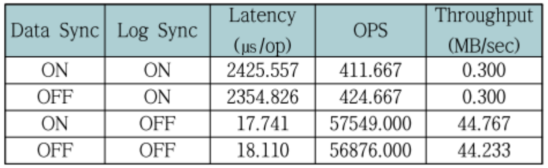
(출처 : www.rocksdb.org)

[표 1] 실험 환경



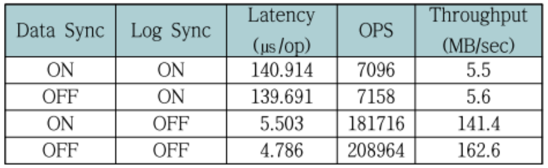
운영체제	Ubuntu 12.04.4 LTS Kernel 3.2.0
프로세서	Intel® CoreTM i5-4670 CPU @ 3.40GHz
메모리(RAM)	4.00GB
저장장치	Samsung 840 PRO SSD, Samsung Dura SSD 853 T

[표 2] 840 PRO SSD에서의 성능 측정 결과



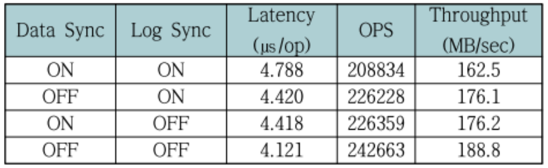
Data Sync	Log Sync	Latency (μs/op)	OPS	Throughput (MB/sec)
ON	ON	2425.557	411.667	0.300
OFF	ON	2354.826	424.667	0.300
ON	OFF	17.741	57549.000	44.767
OFF	OFF	18.110	56876.000	44.233

[표 3] DuraSSD에서의 성능 측정 결과



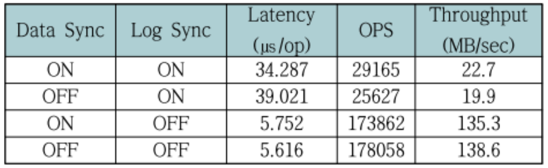
Data Sync	Log Sync	Latency (μs/op)	OPS	Throughput (MB/sec)
ON	ON	140.914	7096	5.5
OFF	ON	139.691	7158	5.6
ON	OFF	5.503	181716	141.4
OFF	OFF	4.786	208964	162.6

[표 4] RAMDISK(tmpfs)에서의 성능 측정 결과



Data Sync	Log Sync	Latency (μs/op)	OPS	Throughput (MB/sec)
ON	ON	4.788	208834	162.5
OFF	ON	4.420	226228	176.1
ON	OFF	4.418	226359	176.2
OFF	OFF	4.121	242663	188.8

[표 5] RAMDISK(file system 추가)에서의 성능 측정 결과

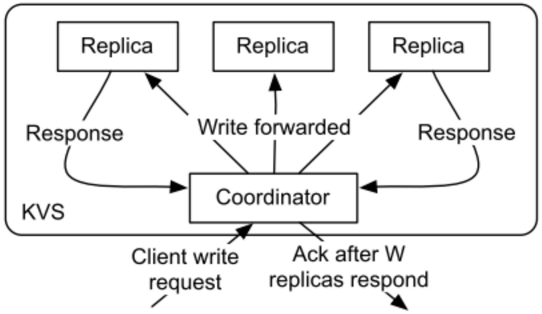


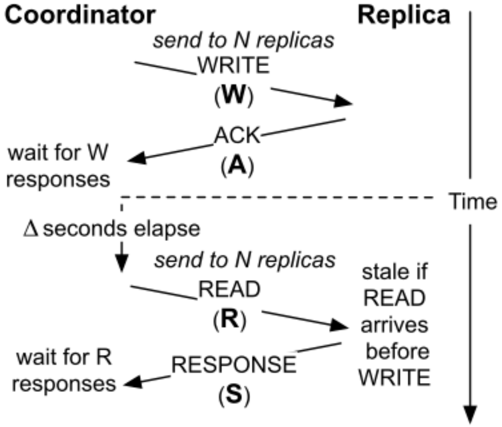
Data Sync	Log Sync	Latency (μs/op)	OPS	Throughput (MB/sec)
ON	ON	34.287	29165	22.7
OFF	ON	39.021	25627	19.9
ON	OFF	5.752	173862	135.3
OFF	OFF	5.616	178058	138.6

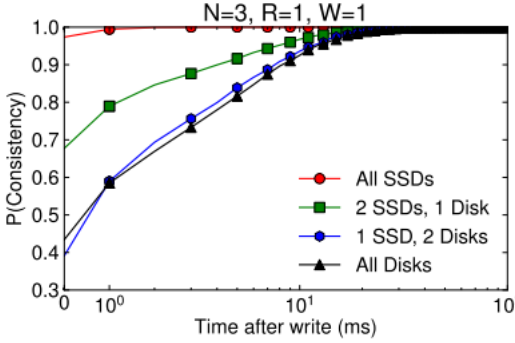
Modern Distributed Database

- Effect of SSD on Eventual Consistency [PBS - VLDB 2013, CACM / VLDBJ 2014]

LNKD-SSD and LNKD-DISK demonstrate the **importance of write latency** in practice. Immediately after write commit, LNKD-SSD had a 97.4% probability of consistent reads, reaching over a 99.999% probability of consistent reads after 5 ms. LNKD-DISK had only a 43.9% probability of consistent reads and, 10 ms later, only a 92.5% probability. This suggests that SSDs may greatly improve consistency due to reduced write variance.







Contents

- DuraSSD
- Latency in WAL log
  - WAL paradigm is ubiquitous!!!
  - DuraSSD vs. Ideal Case in TPC-B
  - DuraSSD vs. Ideal Case in NoSQL YCSB
- Future directions

QnA

Thank you!
Any Question?