Flash-based Database Systems

Experiences from the FlashDB Project

Xiaofeng Meng
Renmin University of China

National Database Conference of China, Hefei, 2012,10,13

Outline

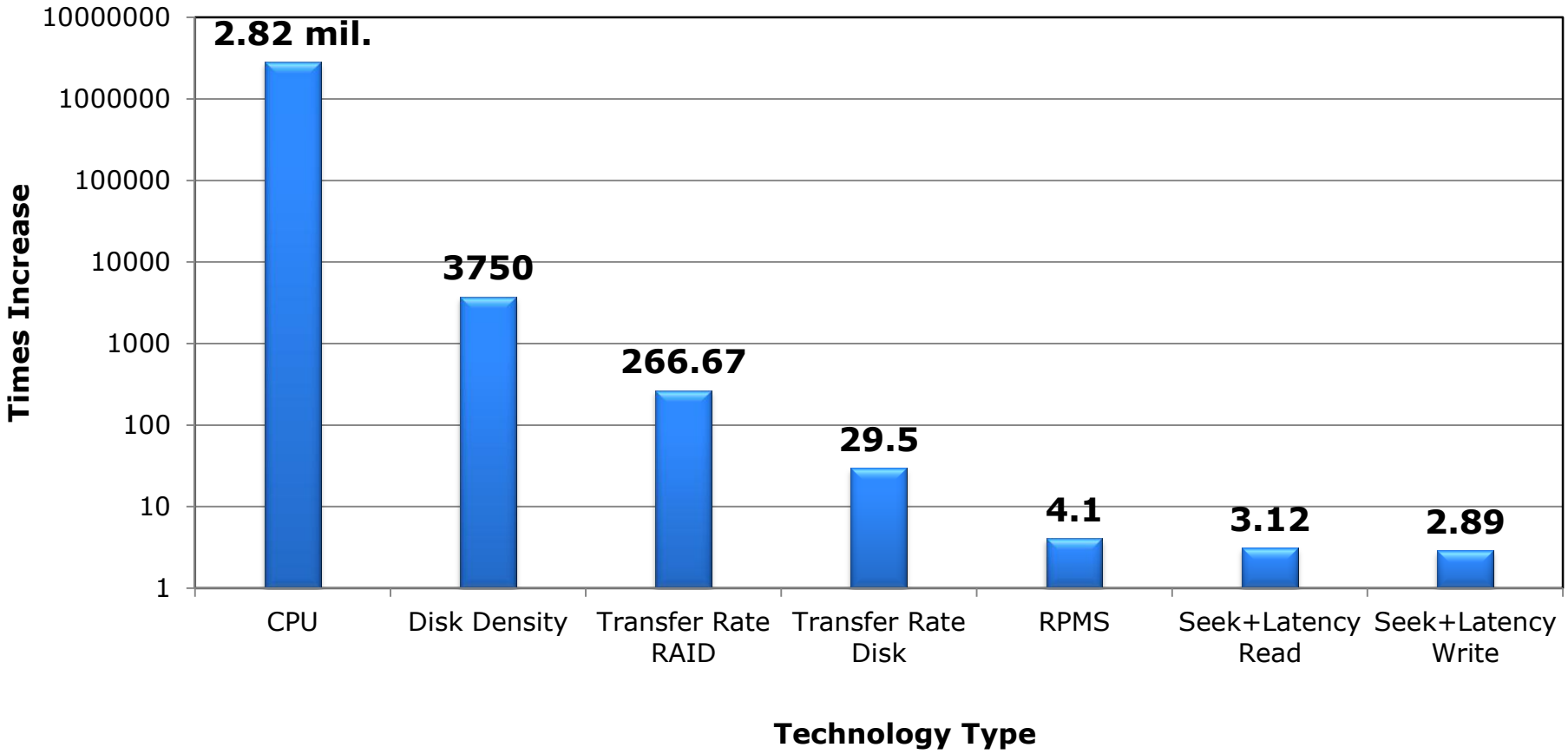
1. New Storage
2. Flash-based DBMSs
3. SSD Hybrid Systems
4. Future Work

Outline

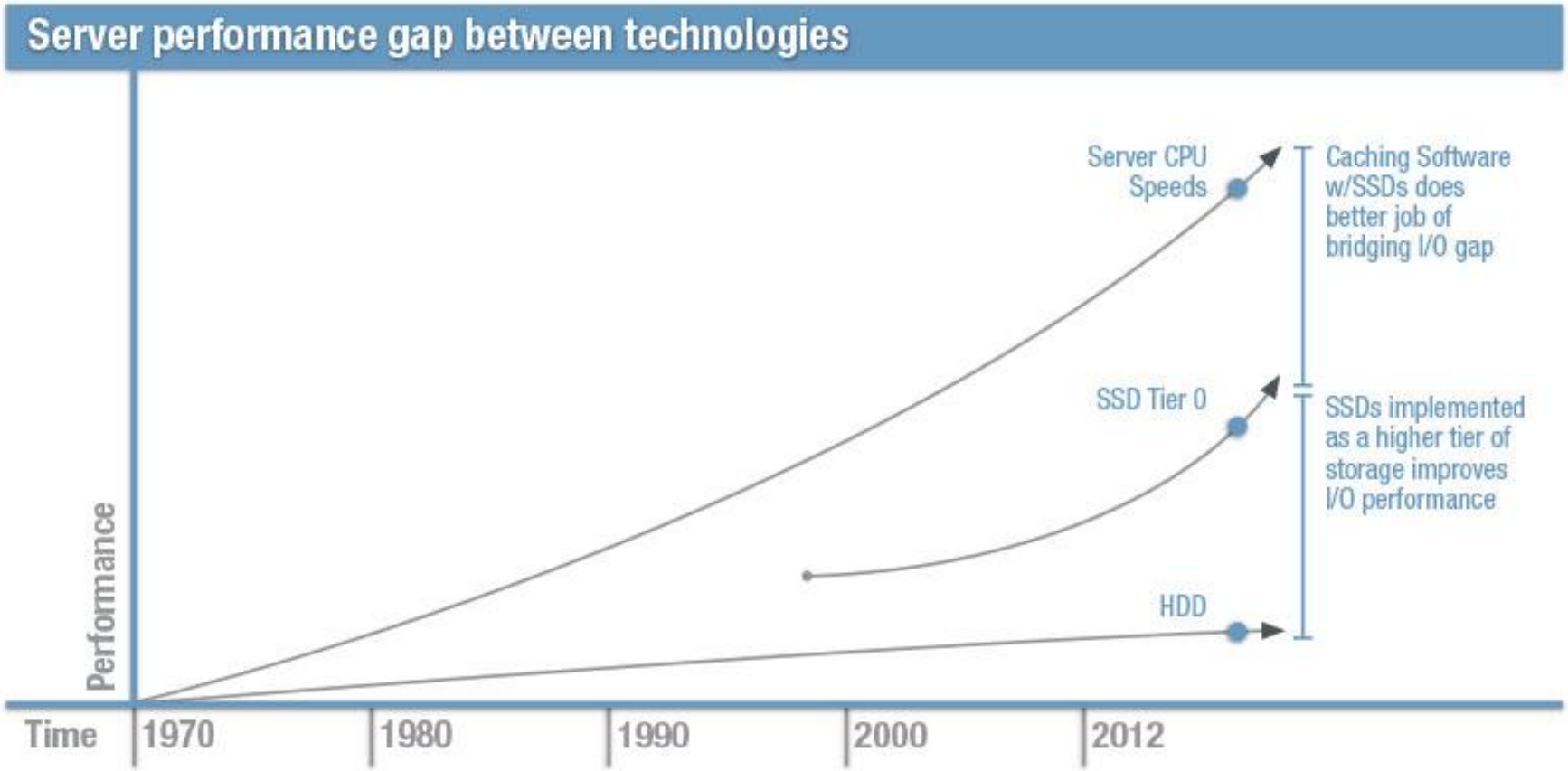
1. New Storage

2. Flash-based DBMSs
3. SSD Hybrid Systems
4. Future Work

Performance Improvement: Disk vs. CPU Over the Last 30 Years



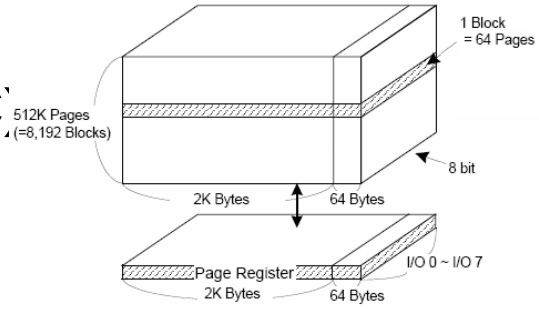
SSD will Bring Storage Performance Back in line with CPU Performance



WE NEED SSD!

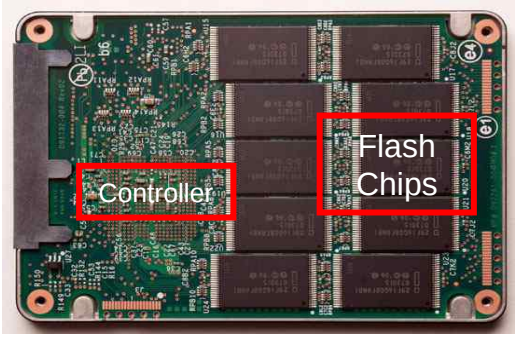
Flash Memory Chip

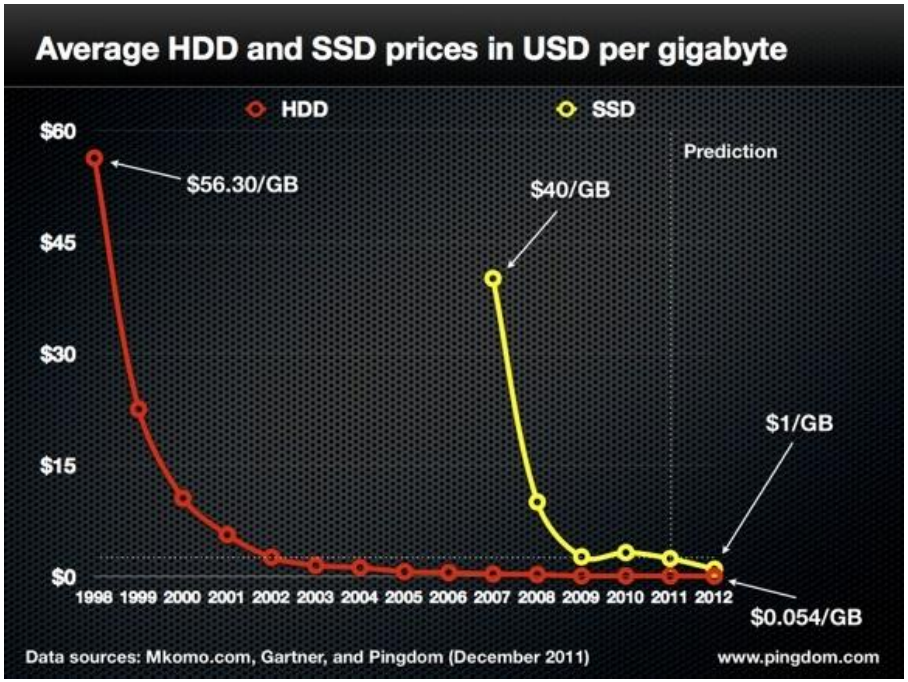
- Flash Memory
  - NOR Flash: Procedure
  - NAND Flash: Data
- NAND Flash
  - Density increases
    - SLC (1bit), MLC (2bits), TLC (3bits)
  - Lifetime decreases
    - 100,000 (SLC), 10,000 (MLC), 5,000 (TLC)
  - Flash chip layout and structure
    - Larger blocks (32 -> 256 Pages)
    - Larger pages: 512 B (old SLC) -> 16KB (future TLC)



Solid State Disk (SSD)

- Flash Translation Layer (FTL)
- Interface
  - SATA, SAS, PCIE
- Manufactures
  - Intel, OCZ, Samsung…
- Capacity/Price





Application of Flash Devices

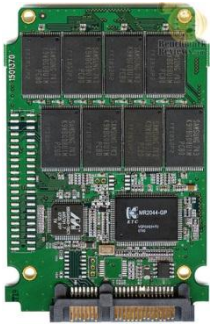
Mobile Devices
Personal Computer

Aerospace

Data Center

Embedded Devices

SSD vs. HDD



	Enterprise SSD	Ratio	Enterprise HDD
IOPs (4 KB)	10^5	150x	10^2
Seq. Read BW (MB/s)	>450	3x	>150
Ran. 80/20 BW (MB/s)	>450	22x	20
Avg. Random I/O latency	μs	>1000x	ms
Active Power	10w	60%	17w
Typical Capacity	300GB	2/3	450GB

Research Motivation (1)

- 60～150x faster data r/w speed (vs. 7200RPM HDD)

Media	Access time		
	Read	Write	Erase
Magnetic† Disk	12.7 ms (2 KB)	13.7 ms (2 KB)	N/A
NAND Flash‡	80 μs (2 KB)	200 μs (2 KB)	1.5 ms (128 KB)

- Query processing time is improved only up to 10x

Read Queries	Query processing time (sec)		Write Queries	Query processing time (sec)	
	Disk	Flash		Disk	Flash
Sequential (Q_1)	14.04	11.02	Sequential (Q_4)	34.03	26.01
Random (Q_2)	61.07	12.05	Random (Q_5)	151.92	61.76
Random (Q_3)	172.01	13.05	Random (Q_6)	340.72	369.88

Sang-Won Lee et al. Design of Flash-Based DBMS: An In-Page Logging Approach. SIGMOD'*07*

Research Motivation (2)

- Transaction processing performance is improved 2～10x

no. of concurrent transactions	hard disk		flash SSD	
	TPS	%CPU	TPS	%CPU
4	178	2.5	2222	28
8	358	4.5	4050	47
16	711	8.5	6274	77
32	1403	20	5953	84
64	2737	38	5701	84

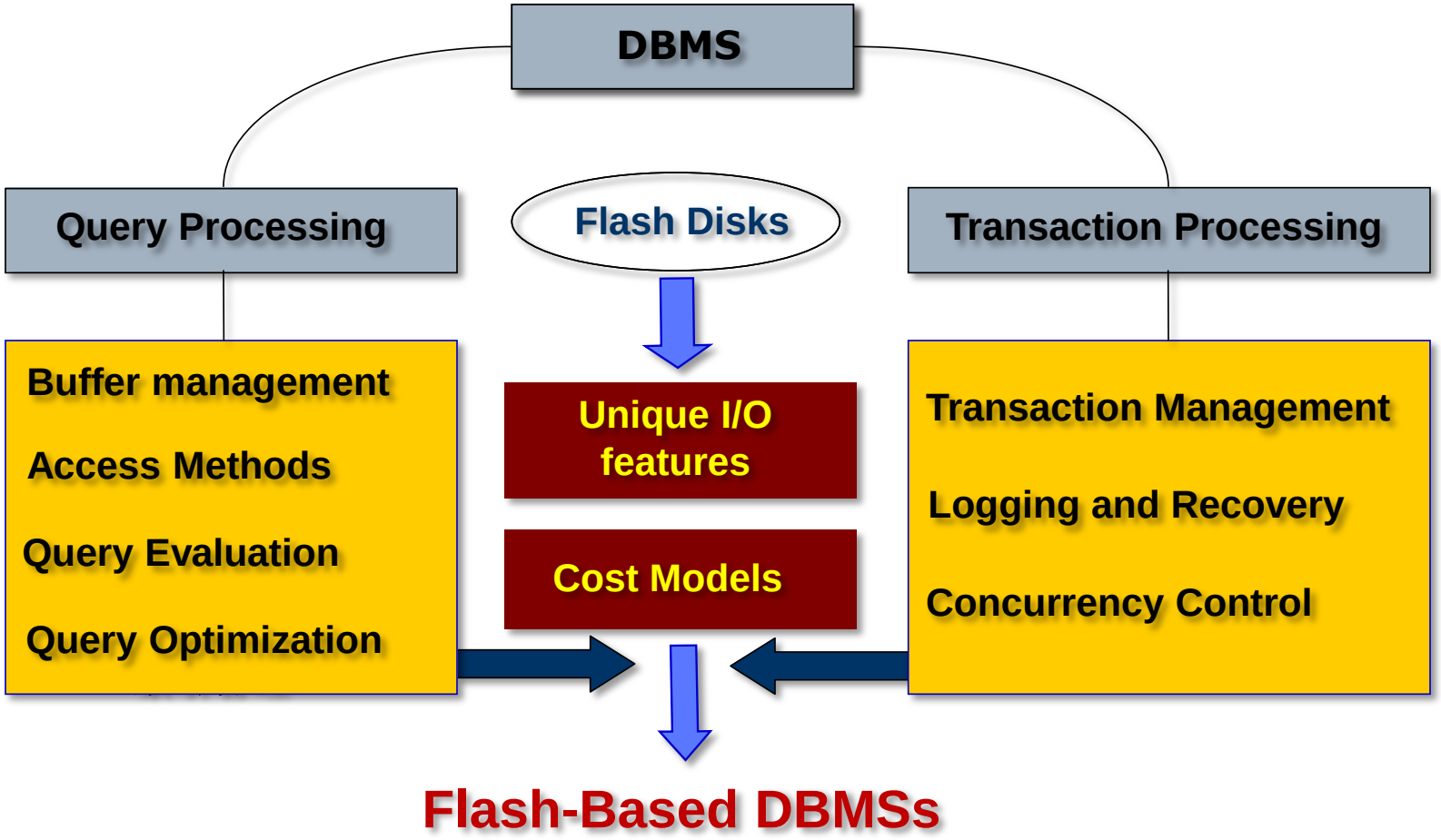
TPS: Transactions-per-Second

The fast access performance of flash memory is not fully exploited by existing database algorithms!

Sang-Won Lee et al. A Case for Flash Memory SSD in Enterprise Database Applications. SIGMOD'08

Research Goal of FlashDB Project

- To boost database performance by exploiting unique flash I/O characteristics



FlashDB Project - Background

- National key project funded by NSFC
  - To investigate novel database technologies for flash-memory based DBMSs
  - To explore applications for flash-based DBMSs
  - Funding period: 2009-2012

- Participating institutions
  - Renmin University of China (Leading)
  - University of Science & Technology of China
  - Hong Kong Baptist University

FlashDB Meetings

- Bi-annual meetings (every semester in 2009-2012)
  - Progress reporting
  - Experience sharing
  - Brainstorm new ideas
  - Participations from academia and industry (IBM, Baidu, Huawei)
- International workshop FlashDB(Hong Kong 2011, Bushan 2012)

Outline

1. New Storage
2. Flash-based DBMSs
3. SSD Hybrid Systems
4. Future Work

Flash is Coming

- The age of flash-based DBMSs is coming
  - Oracle's TPC-C BM result @ 2010 using Exadata
    - Oracle + Sun Flash Storage
    - Total cost: 49M $
      - Storage: 23M $
      - Sun Flash Array: 22M $
      - 720 2TB 7.2K HDD: 0.7M$
  - IBM proposed SSD Buffer (VLDB 10)
  - And MS SQL Server @ Jim Gray Lab ..

ORACLE

SPARC SuperCluster with T3-4 Servers

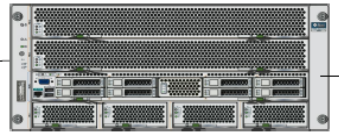
TPC-C 5.11.0
TPC-Pricing 1.5.0
Report Date December 2, 2010

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$30,528,863USD	30,249,688 tpmC	$1.01USD/tpmC	June 1, 2011

Database Server Processors/Cores/Threads	Database Manager	Operating System	Other Software	Number of Users
SPARC T3 1.65GHz 108 / 1,728 / 13,824	Oracle Database 11g Release 2 Enterprise Ed. With Oracle Real Application Clusters and Partitioning	Oracle Solaris 10 09/10	Tuxedo CFS-R Tier 1 Oracle iPlanet Web Server	24,300,000

Clients
81 Sun Fire X4170M2
2.93GHz Intel Xeon X5670 HC
48GB Memory
2 146GB SAS disk

Database Nodes

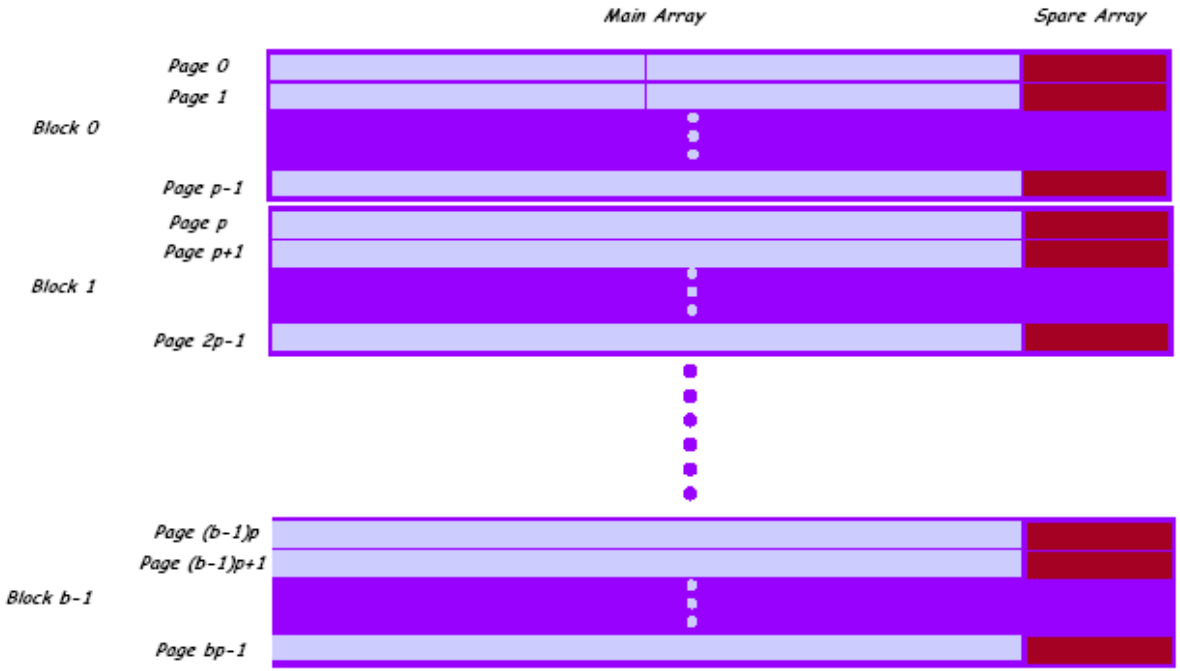


27 Sun SPARC T3-4 Servers
4 1.65GHz SPARC T3
512GB Memory
3 300GB 10K RPM SAS
4 8Gb/s FC HBA, 2 port
10GbE SFP+
5RU High

Storage
67 X4270M2 DATA COMSTAR
6 2TB 7.2K RPM SAS
2 Sun F5100 Flash Arrays
2 X4270M2 DATA COMSTAR
5 2TB 7.2K RPM SAS
2 Sun F5100 Flash Arrays
28 X4270M2 REDO COMSTAR
11 2TB 7.2K RPM SAS

System Component	Each Server Node		Each Client	
Processors/Cores/Threads and cache	4/64/512	SPARC T3 1.65GHz 6 MB L2 Cache	2/12/24	Intel Xeon X5670 12MB Smart Cache
Memory		512GB (13.5TB Total)		48GB
Disk Controllers	4	8Gb/s FC HBA 2 Port	1	8 port Internal SAS
OS Disks (each system)	3	300GB 10K RPM SAS	2	146GB 10K RPM SAS
External Storage (Equally visible to all T3 4 Server nodes)	11,040 720	24GB SSD Flash Modules 2TB 7.2K RPM SAS		
Total Storage		1.76PB		

Flash Organization



- Page size: 2/4/8 KB
- Block = 64 ~ 128 pages: 128/256/512 KB
- A page has a *data* area and a *spare* area
  - *Data* area: for mass data storage
  - *Spare* area: for storing metadata like ECC and LBA

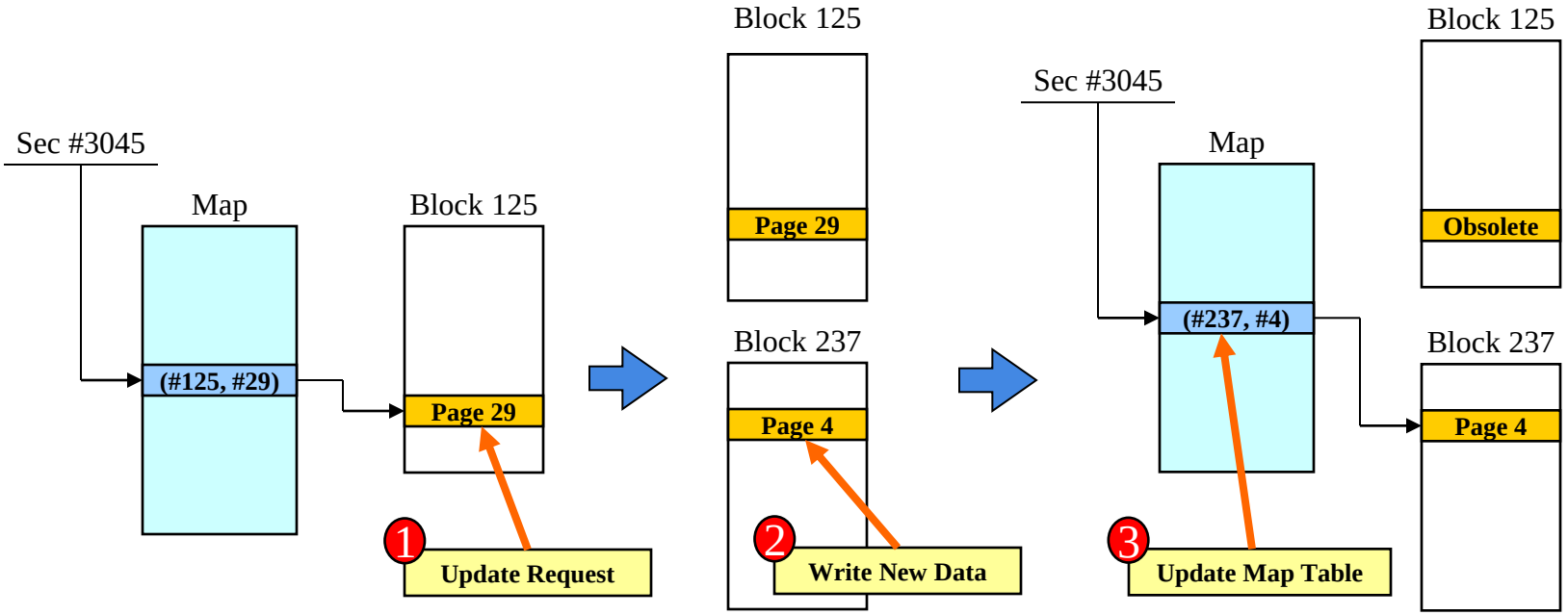
Characteristics of NAND Flash

Media	Access time		
	Read	Write	Erase
Magnetic† Disk	12.7 ms (2 KB)	13.7 ms (2 KB)	N/A
NAND Flash‡	80 μs (2 KB)	200 μs (2 KB)	1.5 ms (128 KB)

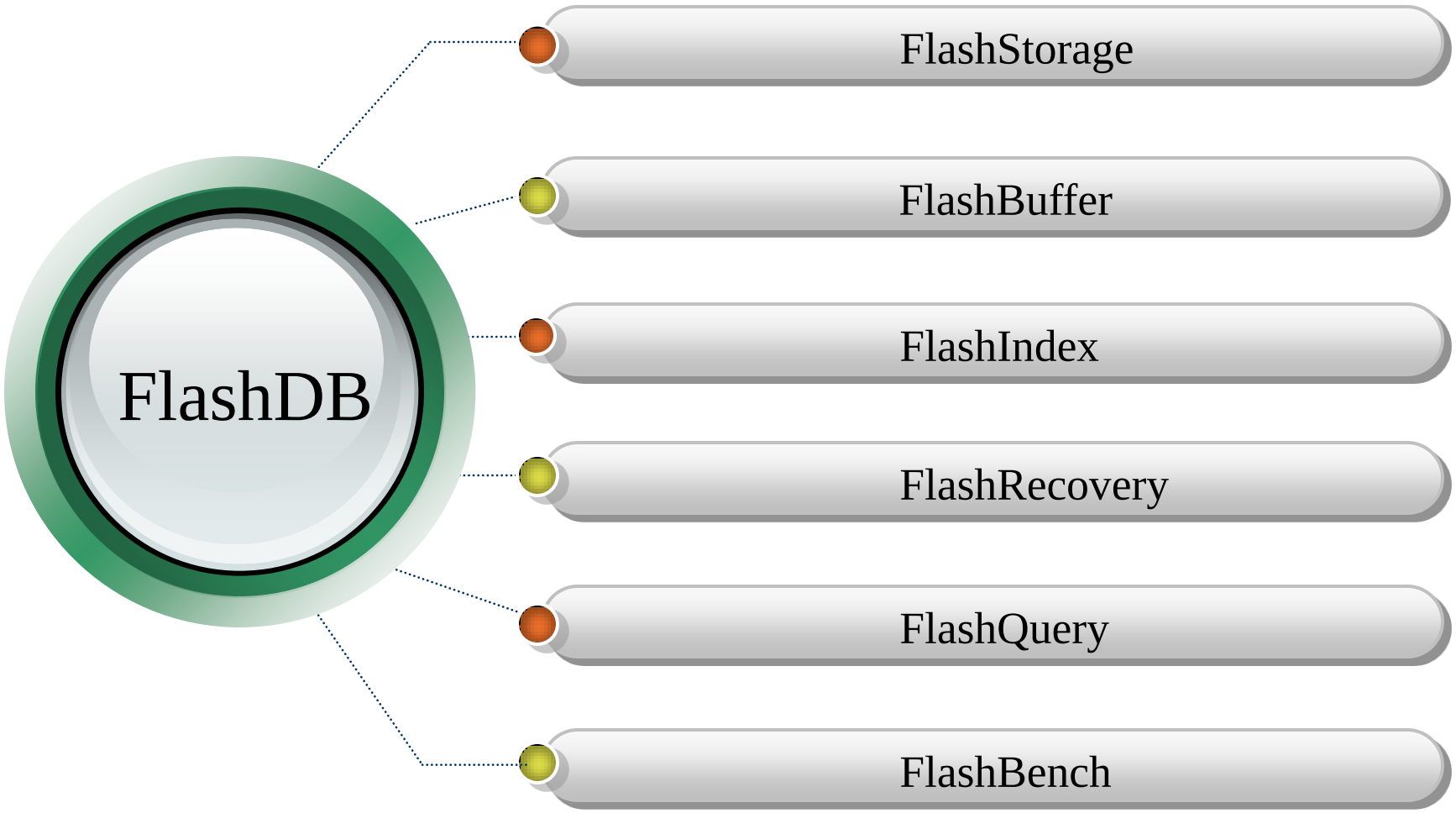
- Asymmetric read/write speed (by pages)
- Random read fast(No mechanical latency)
- Erase-before-overwrite(by Blocks)
- Out-of-Place update

Out-of-Place Update in Flash

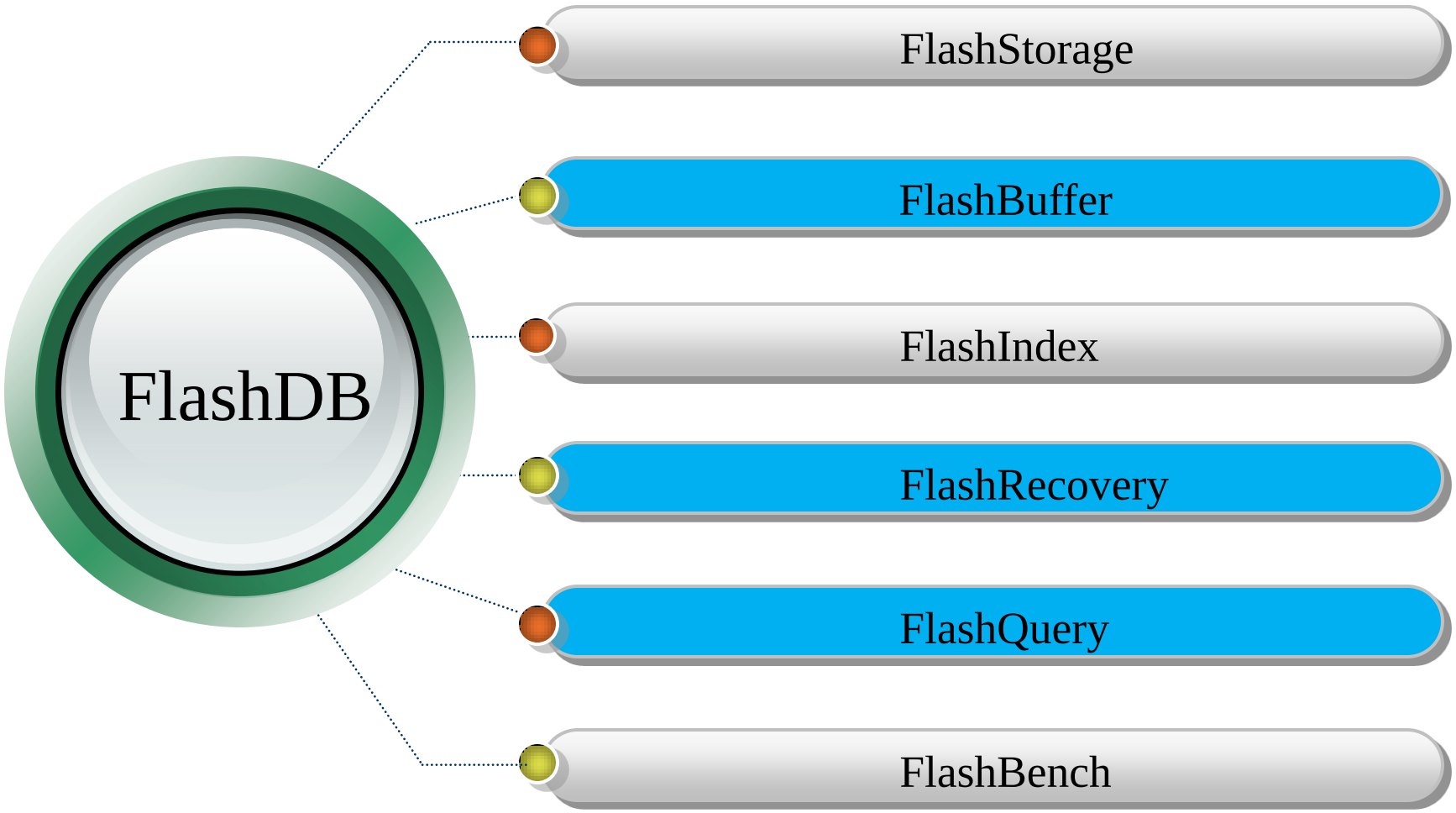
- Flash Transaction Layer (FTL):
  - Addresses Mapping
  - Garbage Collection
  - Wear Leveling



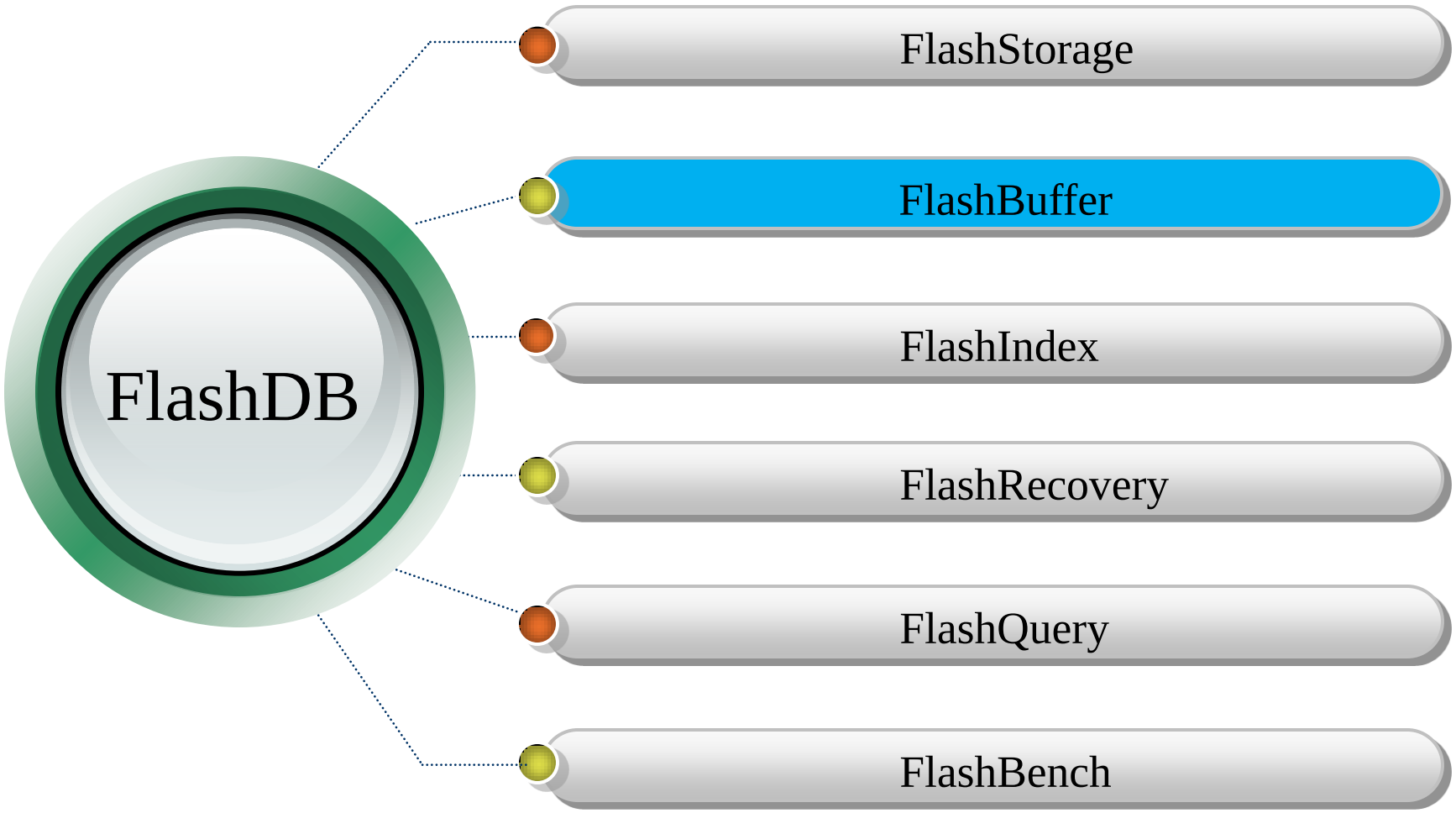
Research on FlashDB



Research on FlashDB



Research on FlashDB

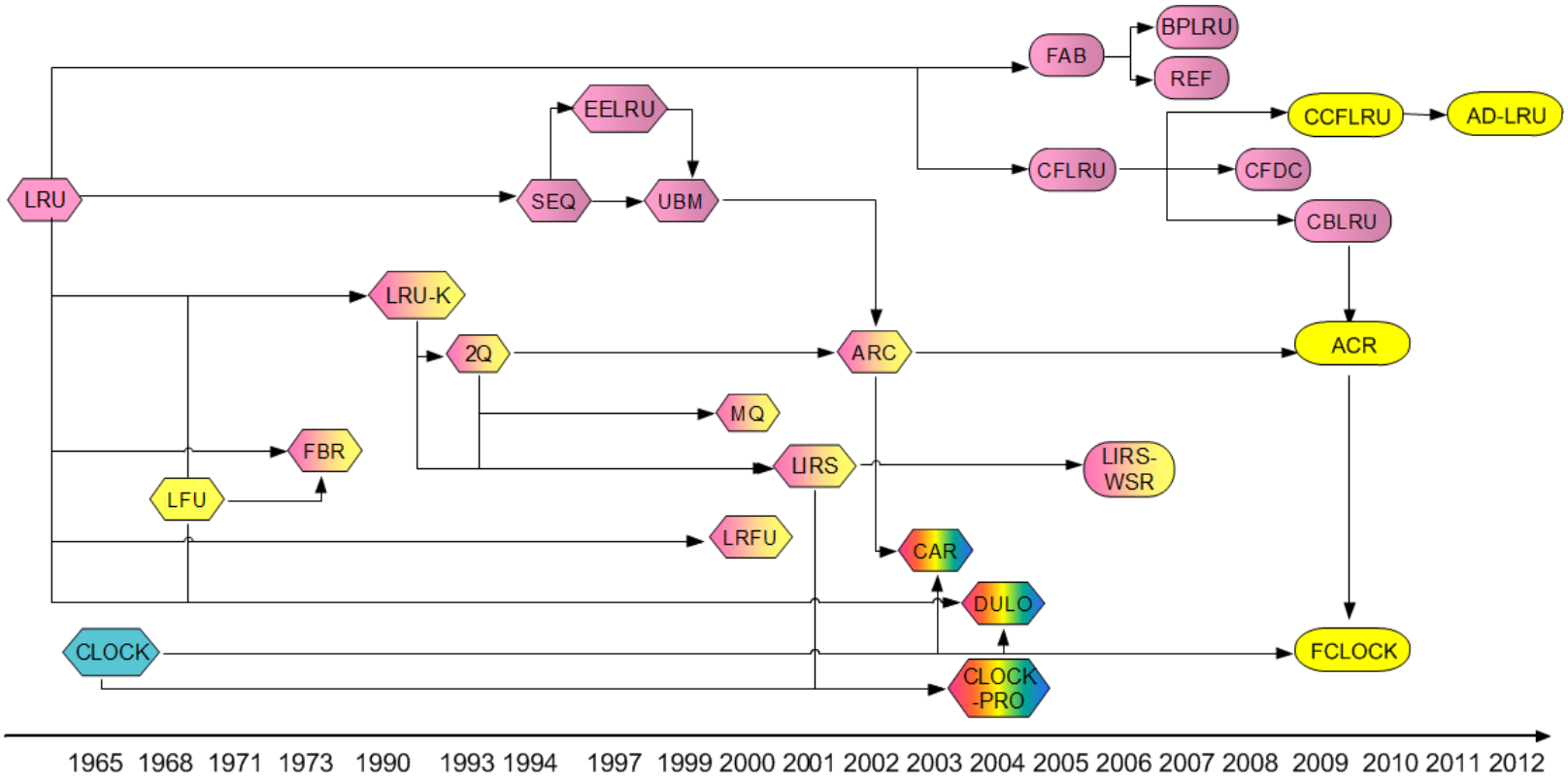


Buffer Management

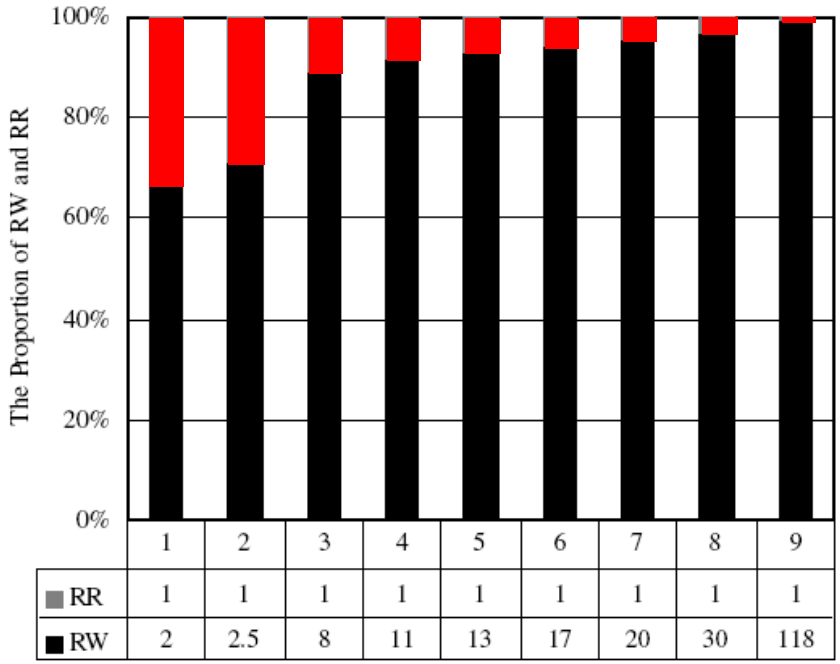
- LRU/CLOCK

Assumption:

$$Cost_{read} = Cost_{write}$$



Motivation

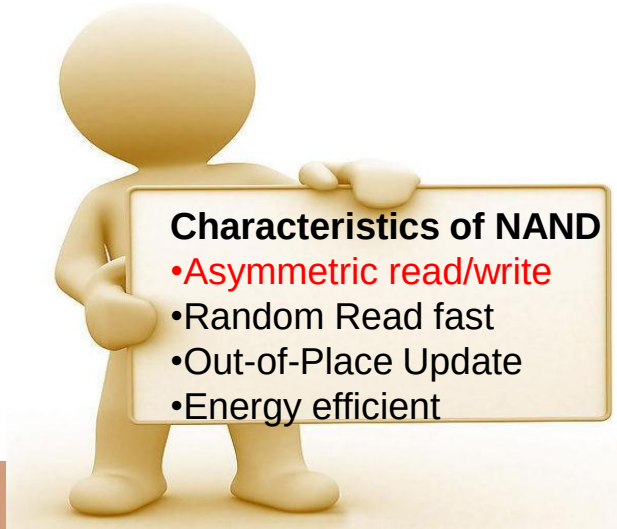


	1	2	3	4	5	6	7	8	9
RR	1	1	1	1	1	1	1	1	1
RW	2	2.5	8	11	13	17	20	30	118

"1" is MCAQE32G8APP-0XA, "2" is K9WAG08U1A,
"3" is K9XXG08UXM, "4" is K9F1208R0B,
"5" is K9GAG08B0M, "6" is Hynix HY27SA1G1M,
"7" is K9K1208U0A, "8" is K9F2808Q0B,
"9" is MCAQE32G5APP

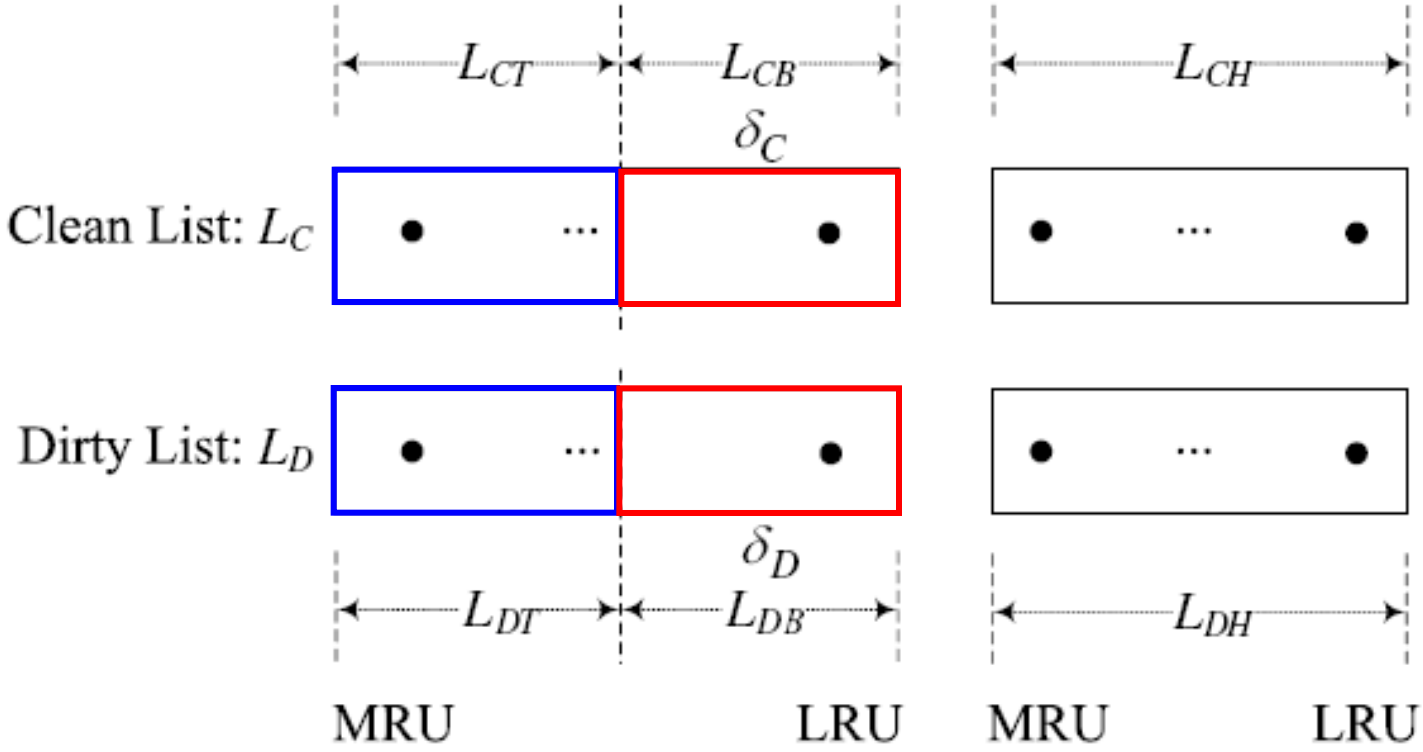
- **Asymmetry** of IO operation cost
- **Discrepancy** of the asymmetry of different SSDs
- Assumption:

$Cost_{read} << Cost_{write}$



ACR [MDM2010]

- Adaptive Cost-aware Replacement
  - 针对闪存的不同读写代价，在内存维护两个LRU队列，Clean队列（Lc）和dirty队列(Ld)



ACR [MDM2010]

置换策略

- ACR根据SSD不同的读写代价来调节Lc和Ld的长度
- L_C (L_D)的长度和**L_C (L_D)的置换代价占整个缓冲区置换代价的比值β**相对应。

$$\beta = C_{L_C} / (C_{L_C} + C_{L_D})$$

$$s = |L_C| + |L_D|$$

LC的置换代价

整个缓冲区的置换代价

$|L_C| < \beta \cdot s$

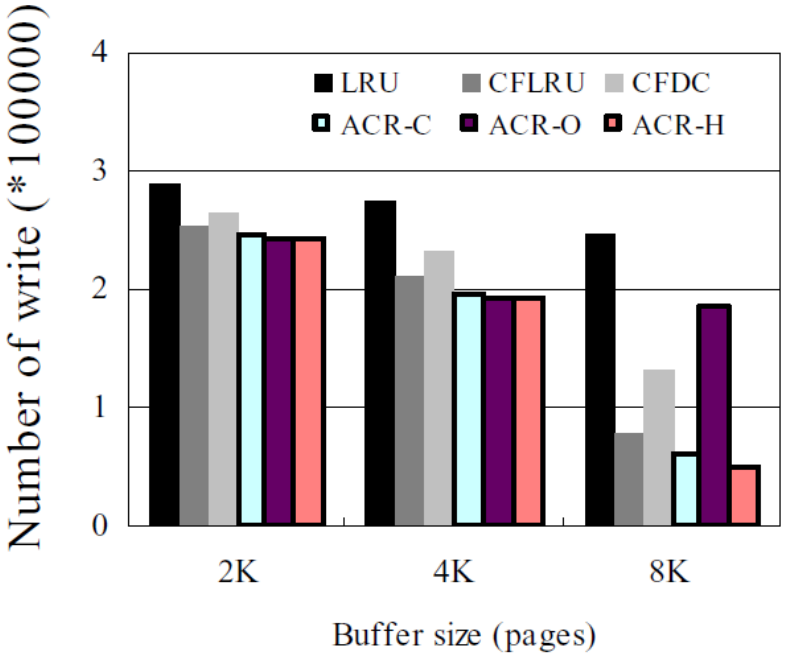
- **L_C太短，选择L_D的LRU位置的数据页进行置换**

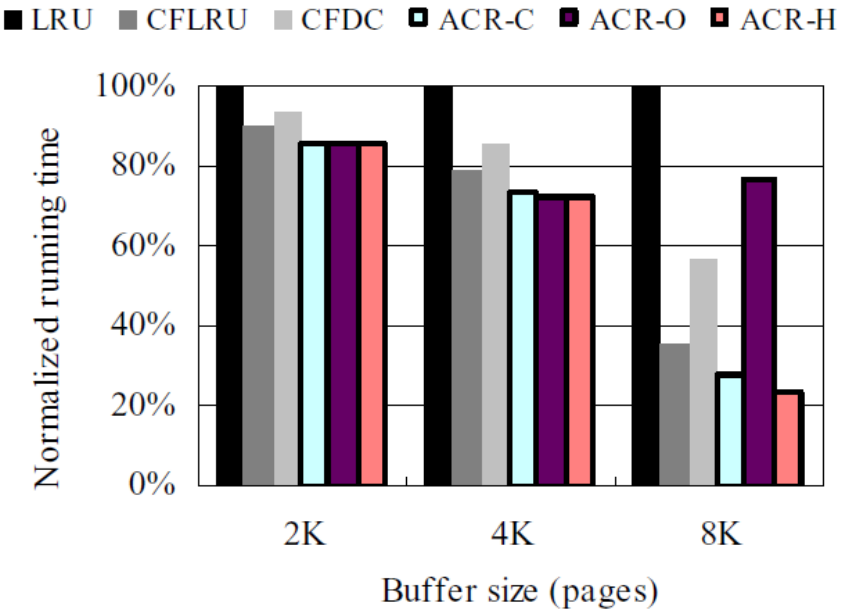
$|L_C| \geq \beta \cdot s$

- **L_C太长，选择L_C的LRU位置的数据页进行置换**

ACR [MDM2010]

- 按照队列计算置换代价
- 考虑不同的闪存的读写差异，能适用于不同类型的闪存





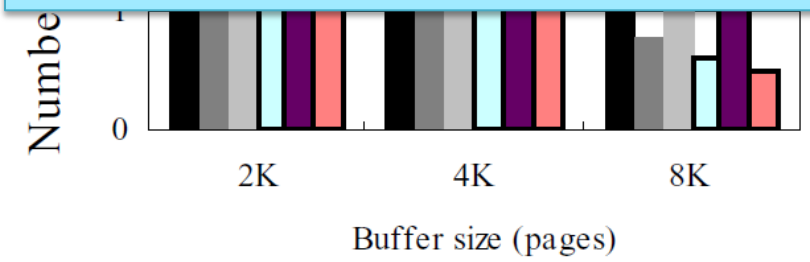
ACR [MDM2010]

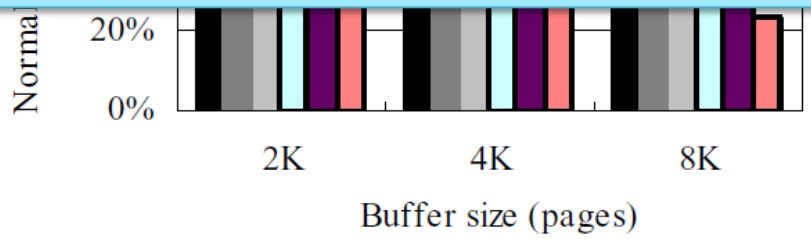
- 按照队列计算置换代价
- 考虑不同的闪存的读写差异，能适用于不同类型的闪存

“This paper targets at an interesting and important topic by considering developing a new buffer cache replacement algorithm on flash disks to solve this problem…”

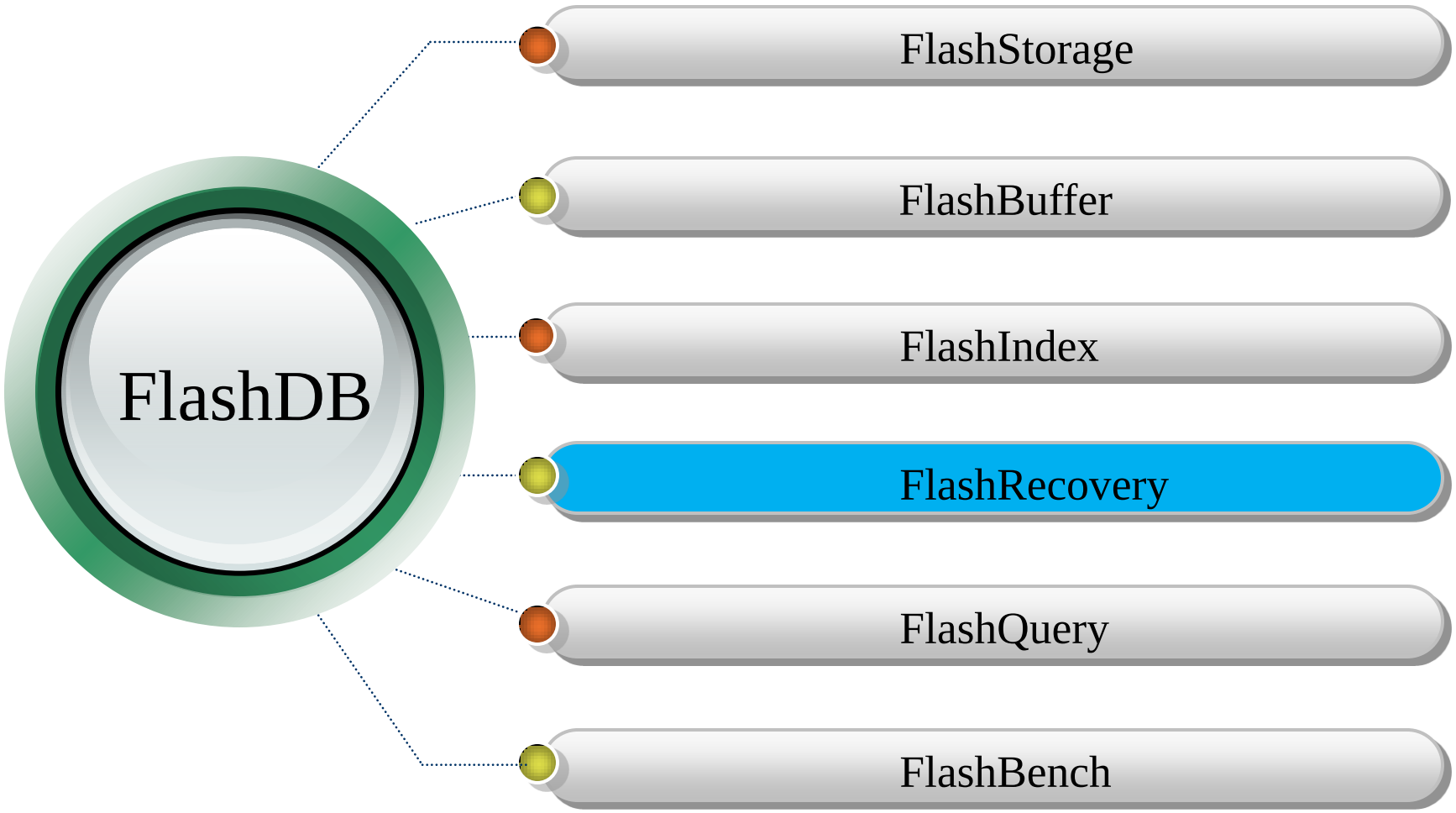
（本文解决的是一个重要并且有趣的问题...）

--美国俄亥俄州立大学张晓东教授的评价



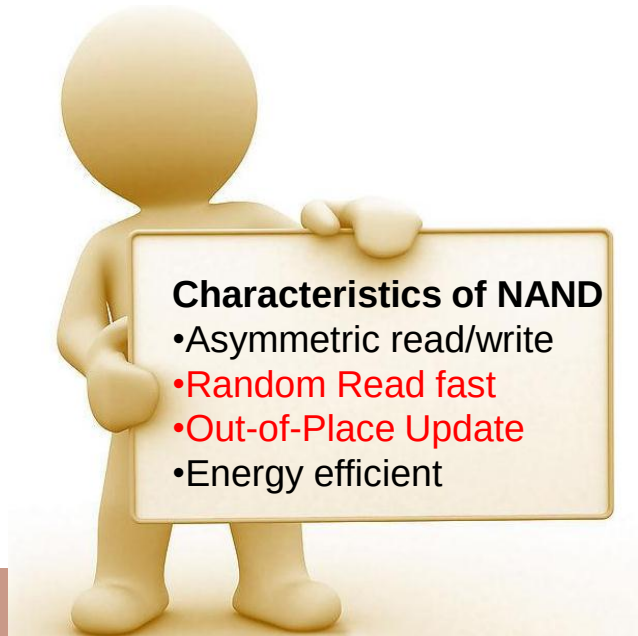


Research on FlashDB

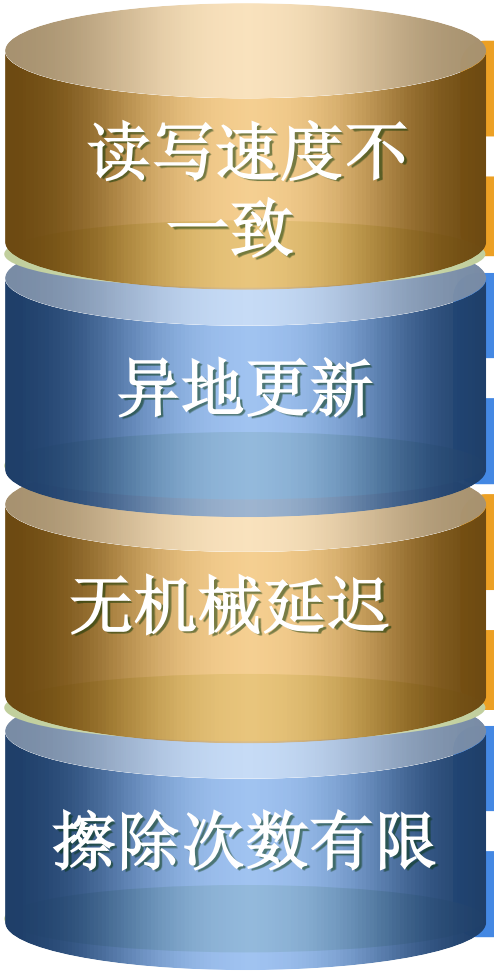


Transaction Recovery

- Write Ahead Log (WAL)
  - Basic Ideas:
    - Updates can be written only after logged;
    - Force log records to disk before a commit is finished;
    - Perform undo/redo operations during abort or recovery
  - Disadvantage: frequent write operations
    - May not preferable for write-expensive flash-based DBMSs



闪存数据库中日志设计思路



- 读速度比写速度快
- 考虑用较多

利用数据的历史版本地址记日志

- 闪存要求
- 利用天然存在的历史版本

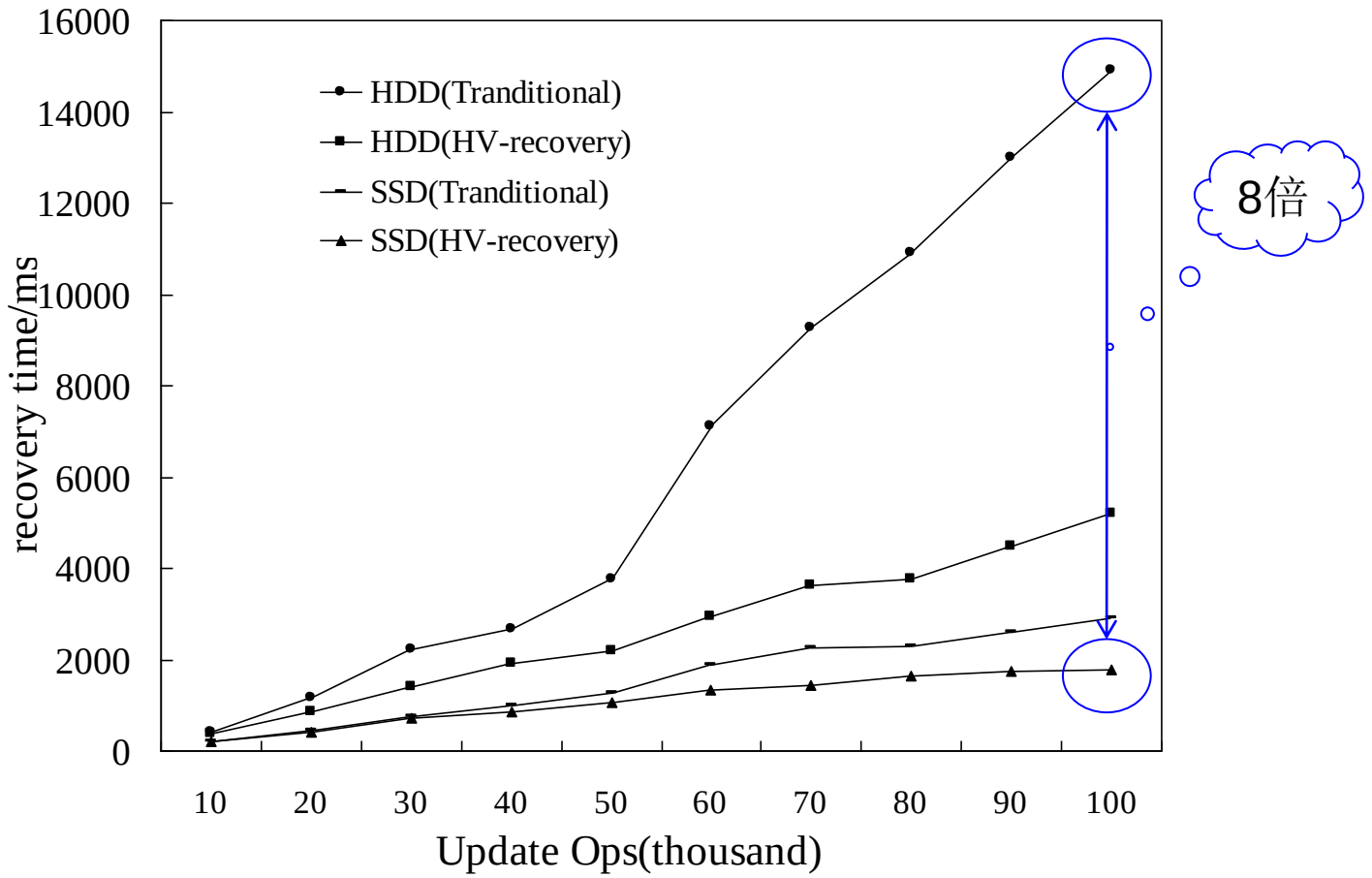
将日志文件由顺序结构转变成链表结构

- 随机和连续访问速度相
- 可以用随机读来代替连续读

- 闪存寿命有限，不可无限制的擦除
- 尽量减少写操作，间接减少擦除

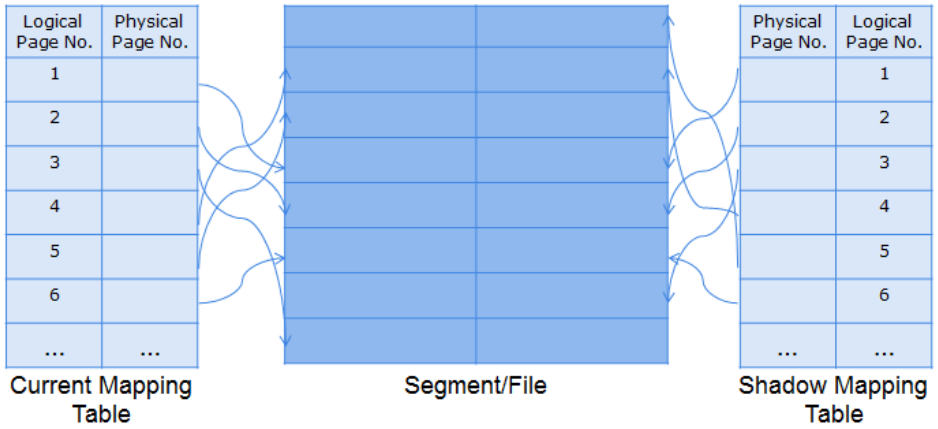
Performance Evaluation

- 利用该设计实现在Berkeley DB中，与传统数据库的日志恢复时间进行比较



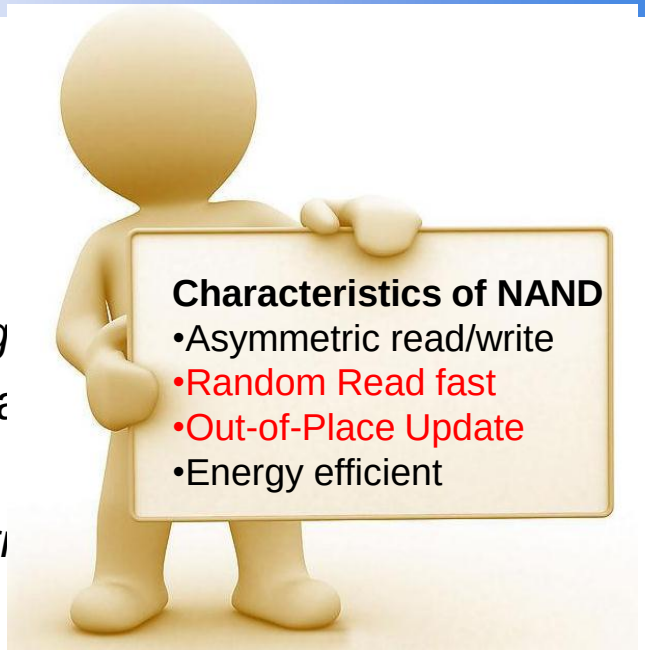
Transaction Recovery

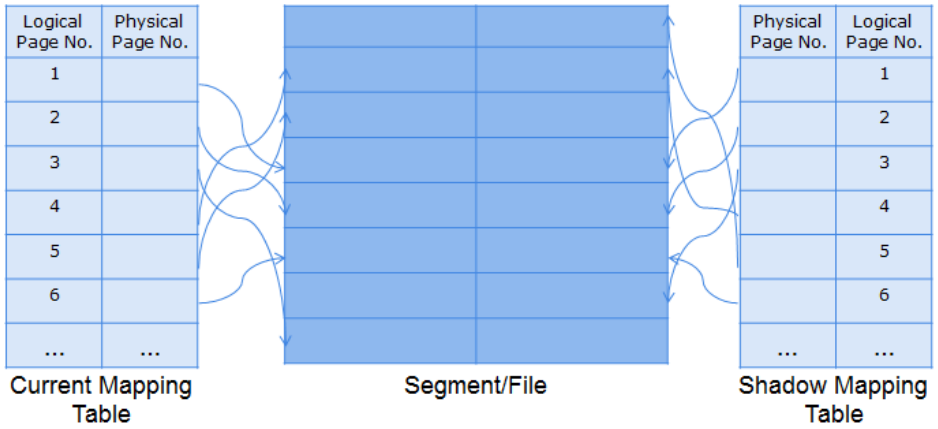
- Write Ahead Log (WAL)
- Shadow Paging[TODS, 1977]
  - Basic idea: out-of-place update
    - Access data pages through a *page mapping table;*
    - Update a page: allocate a *shadow page*, change the *current* mapping
    - Commit: force *current* mappings of updated pages to disk
    - Abort: discard the *shadow page* and the *current* mapping



Transaction Recovery

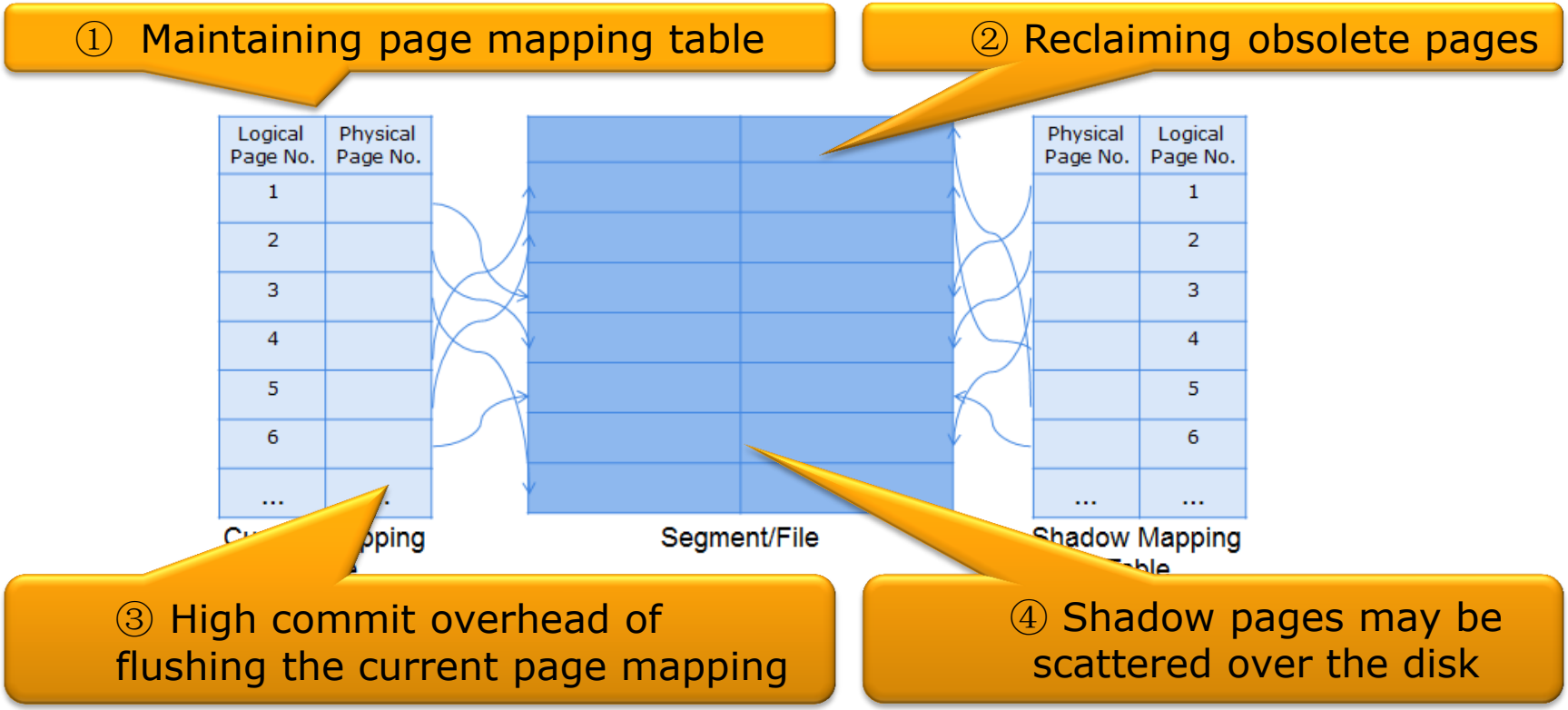
- Write Ahead Log (WAL)
- Shadow Paging[TODS, 1977]
  - Basic idea: out-of-place update
    - Access data pages through a *page mapping*
    - Update a page: allocate a *shadow page*, cha
    - Commit: force *current* mappings of updated
    - Abort: discard the *shadow page* and the *curr*





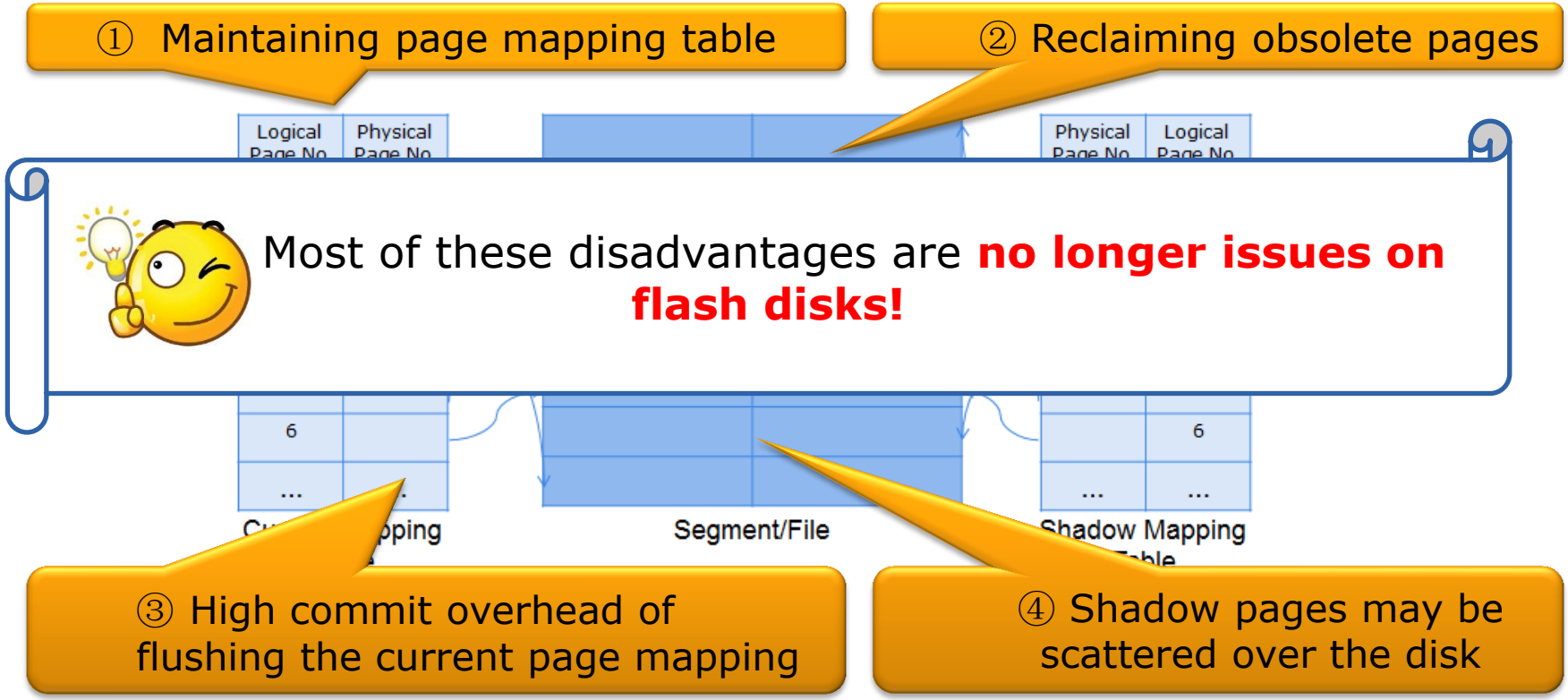
Shadow Paging[TODS, 1977]

- Basic idea: out-of-place update
- Advantage: no need to write log records
- Disadvantages on hard disk



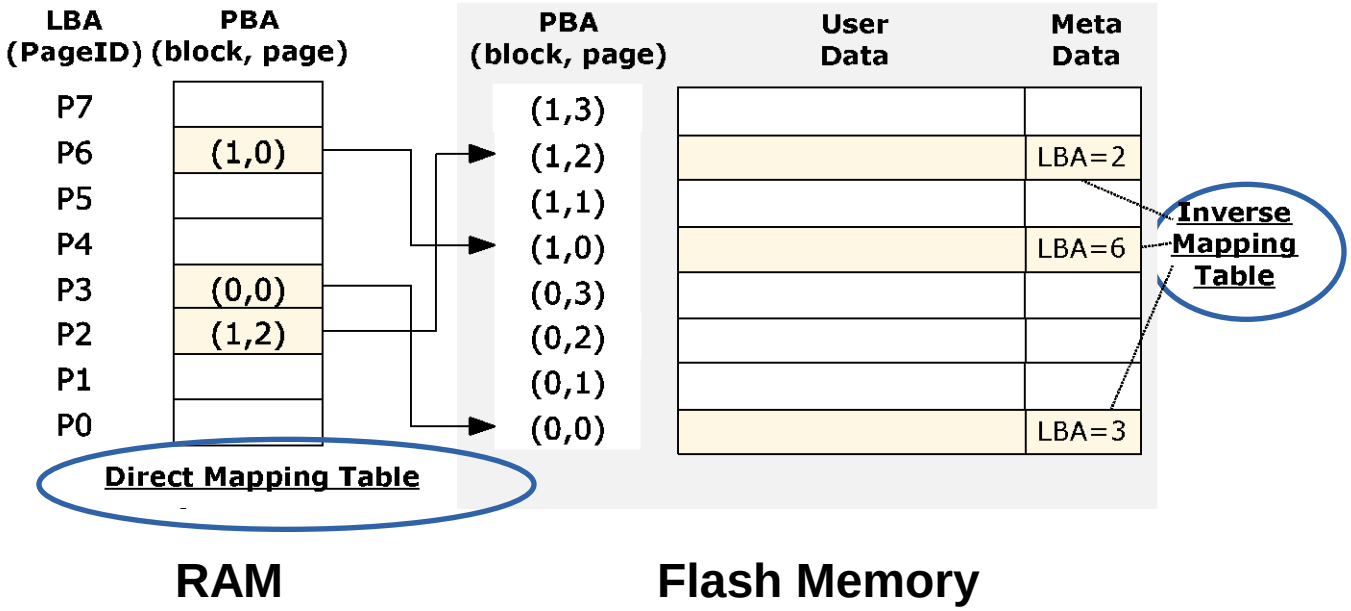
Shadow Paging[TODS, 1977]

- Basic idea: out-of-place update
- Advantage: no need to write log records
- Disadvantages on hard disk



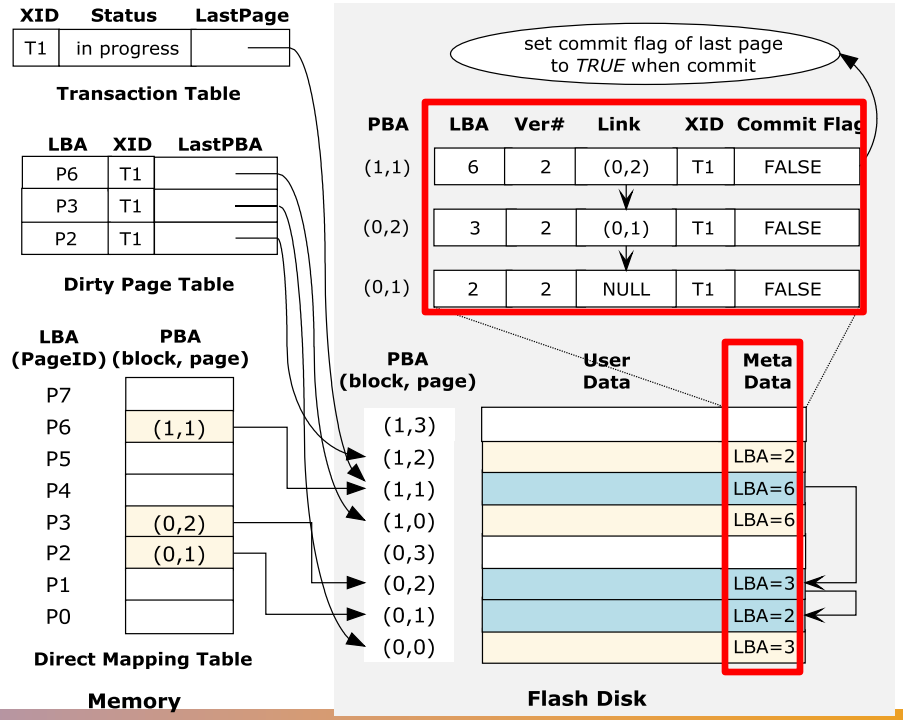
Shadow Paging for SSD

- Two mapping tables in FTL
  - Direct mapping table
  - Inverse mapping table



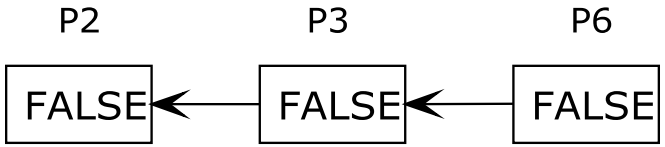
FlagCommit [TKDE, 2011]

- Inverse mapping table(Spare area):
  - Extend to keep Transaction States

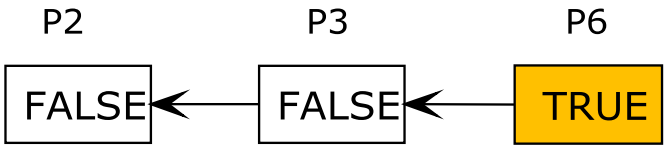


FlagCommit [TKDE, 2011]

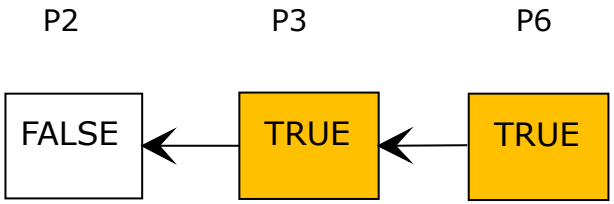
- Inverse mapping table(Spare area):
  - Extend to keep Transaction States
- Flag-base Protocol
  - Commit-based Flag Commit (CFC)
  - Abort-based Flag Commit (AFC)



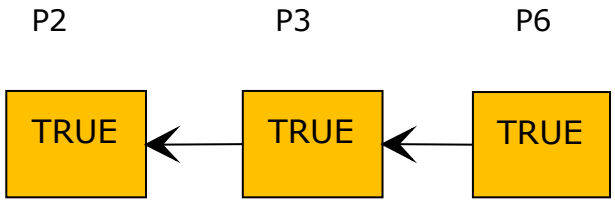
(a) An in-progress / aborted transaction



(b) A committed transaction



(a) An in-progress / aborted transaction

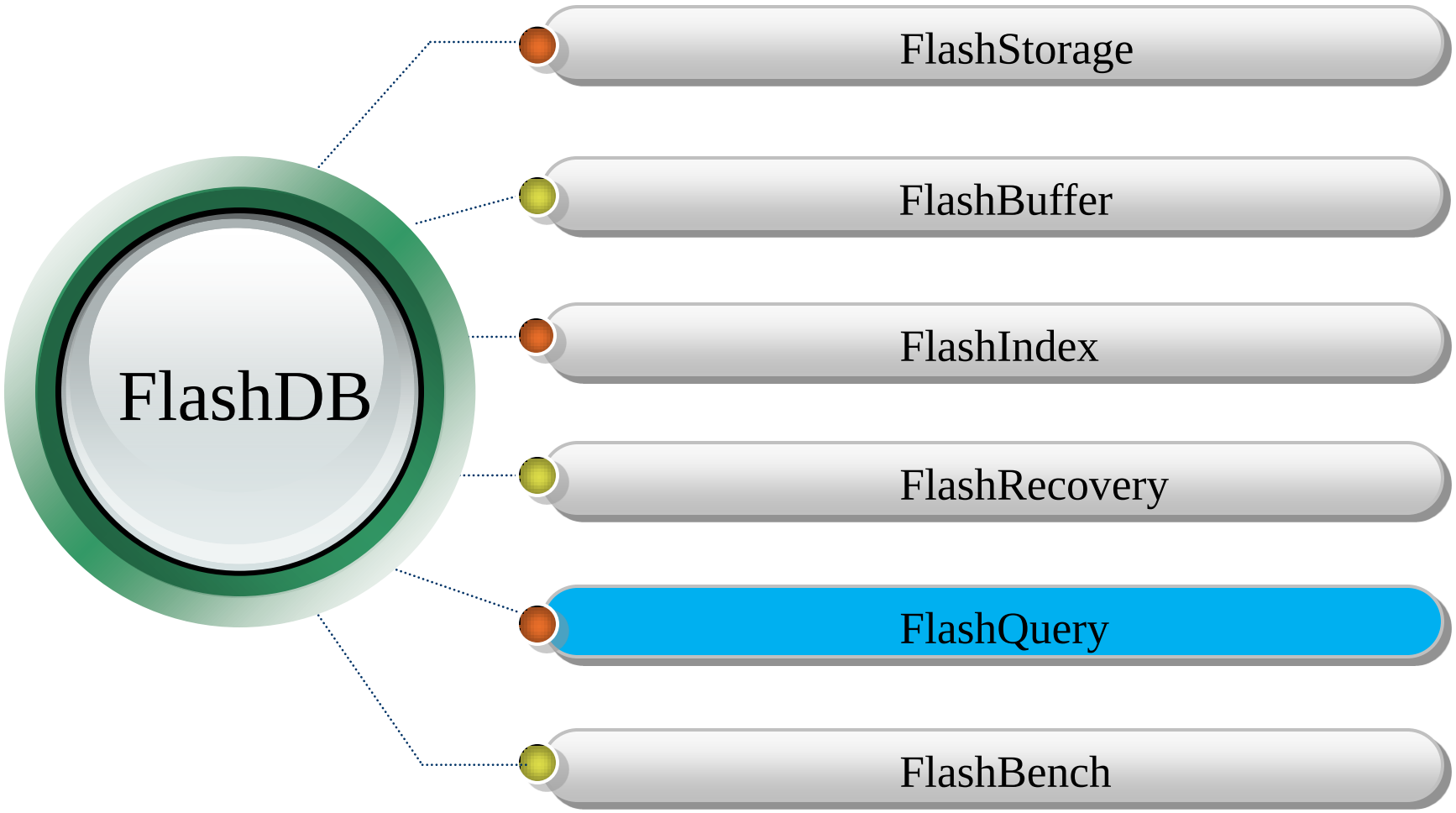


(b) A committed transaction

Shadow Paging ? WAL

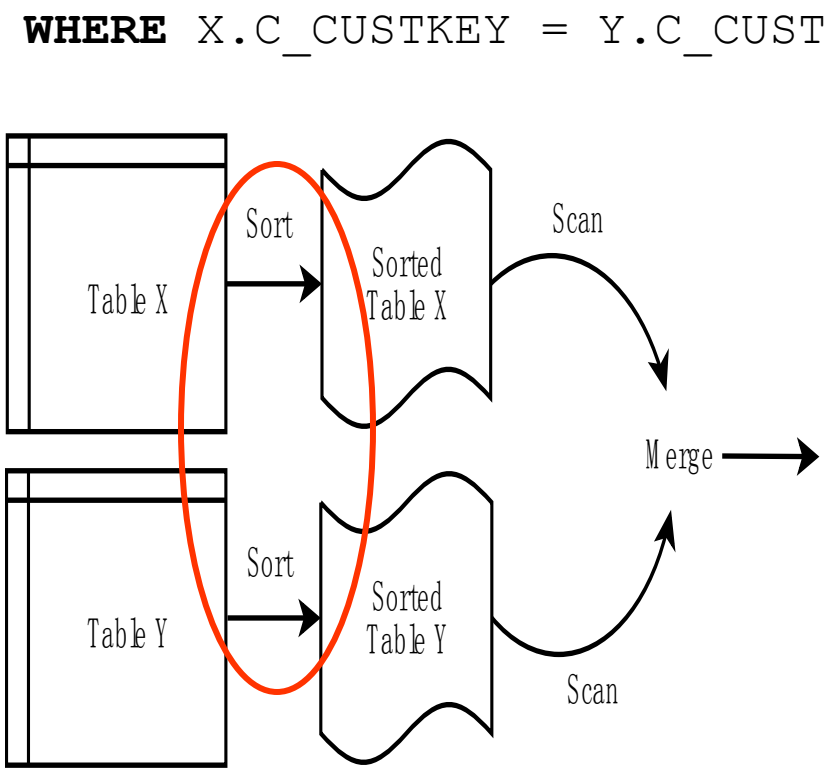
- HDD: Shadow Paging < WAL
  - Bad performance of random operations
  - Random read for recovery
  - Random write when committing
- SSD：Shadow Paging>WAL
  - Good performance of random data access
  - Out-of-place update: multi-version data
  - Garbage Collection

Research on FlashDB

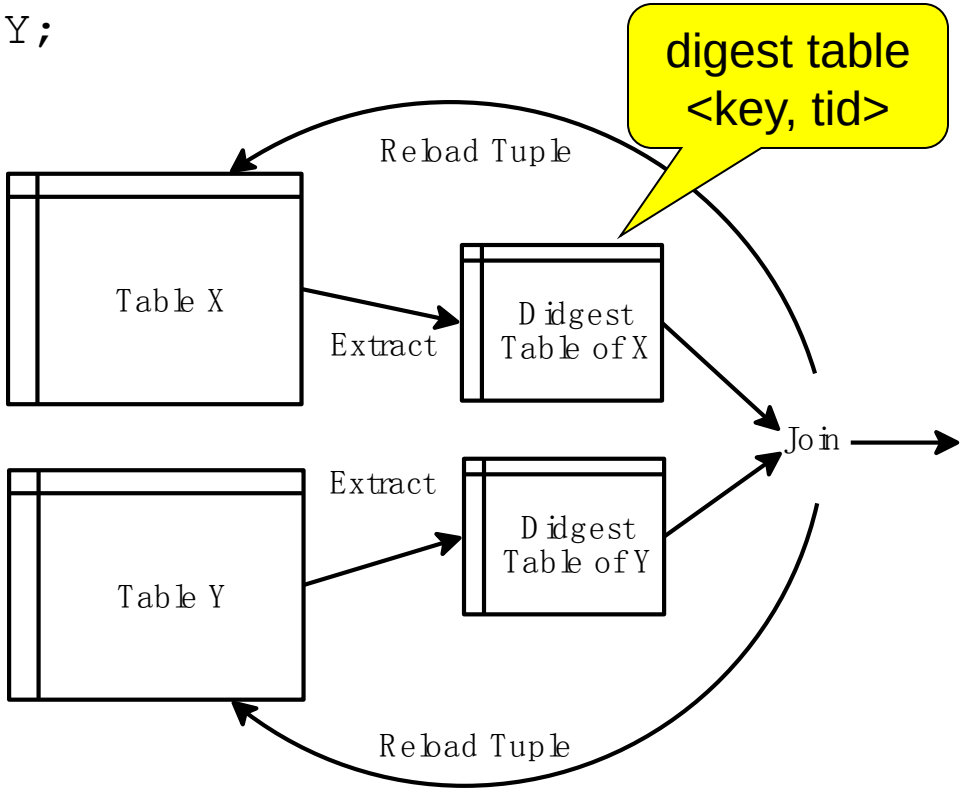


Observation on Sort-Merge-joins

```
SELECT *
FROM CUSTOMER X, ORDERS Y
WHERE X.C_CUSTKEY = Y.C_CUSTKEY;
```



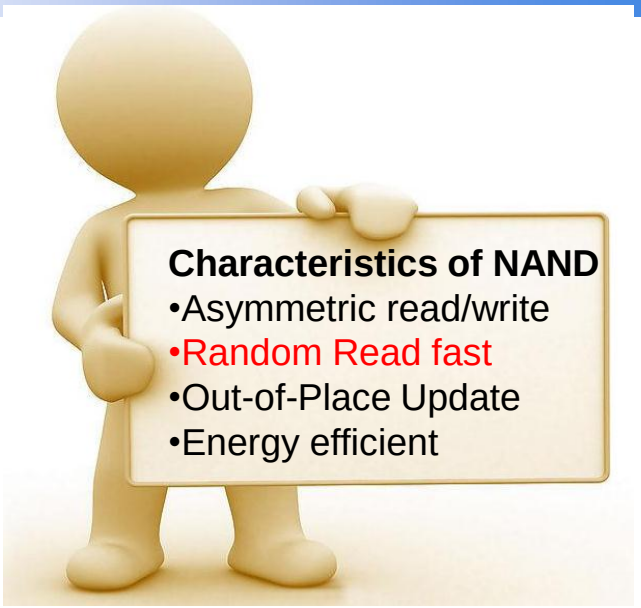
Sort-merge join



Alternative Join

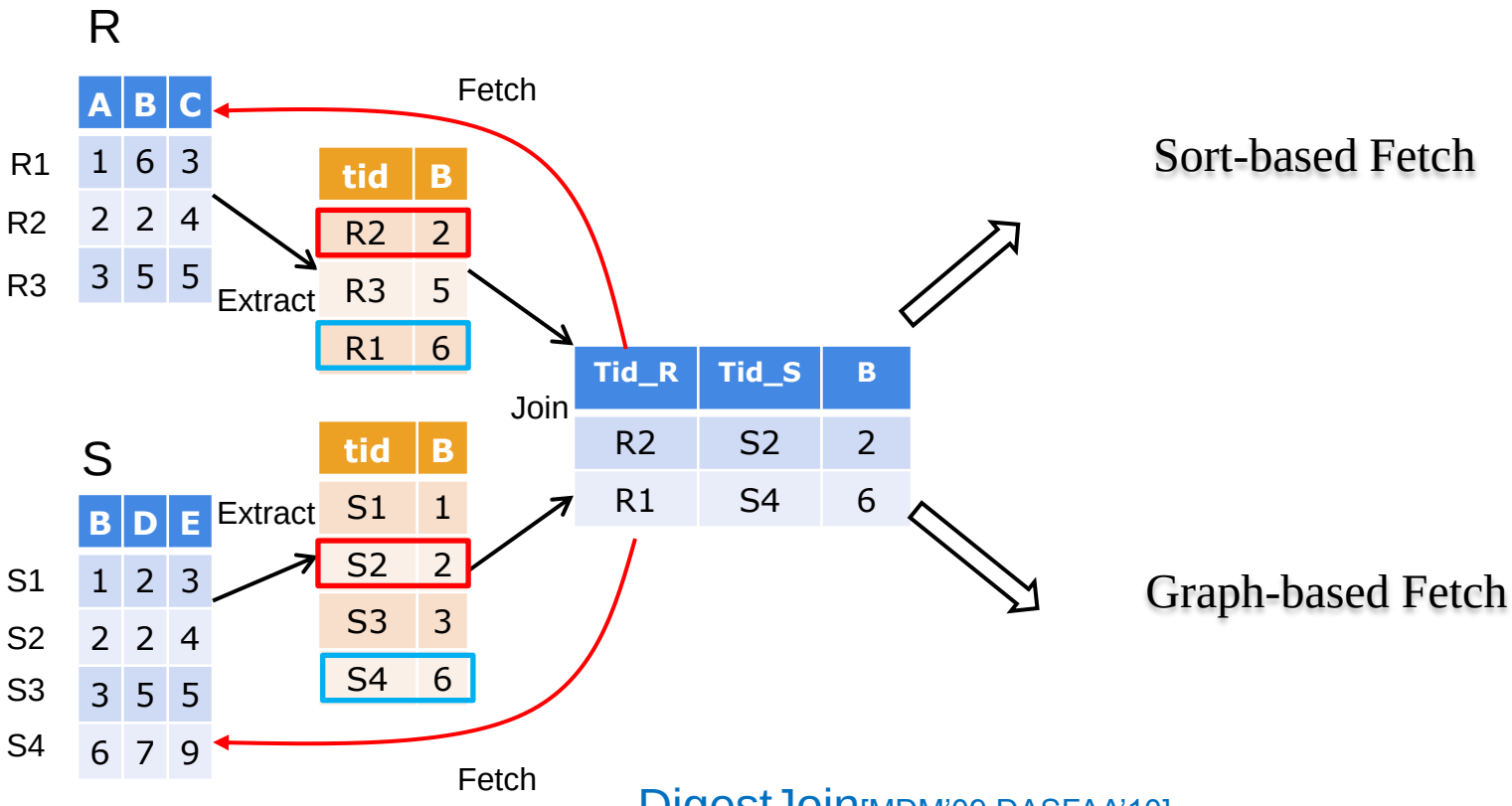
DigestJoin [MDM 2009]

- Random read is no longer an issue on flash disks.
- But writing intermediate results is expensive.



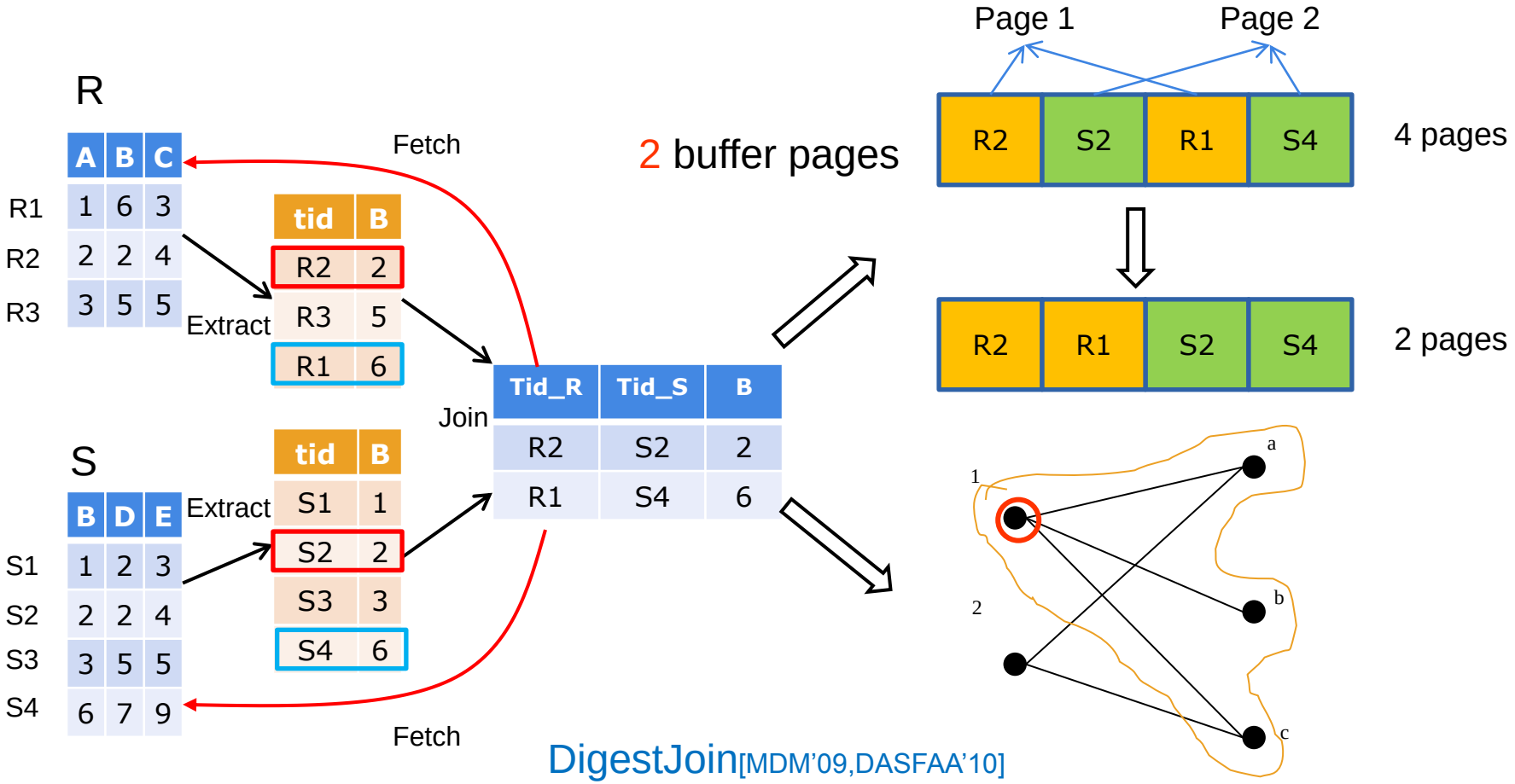
- Idea of DigestJoin:
  - 1st phase: generate digest tables <key, tid> and then join
  - 2nd phase: reload full tuples based on digest join results

DigestJoin [MDM2009]

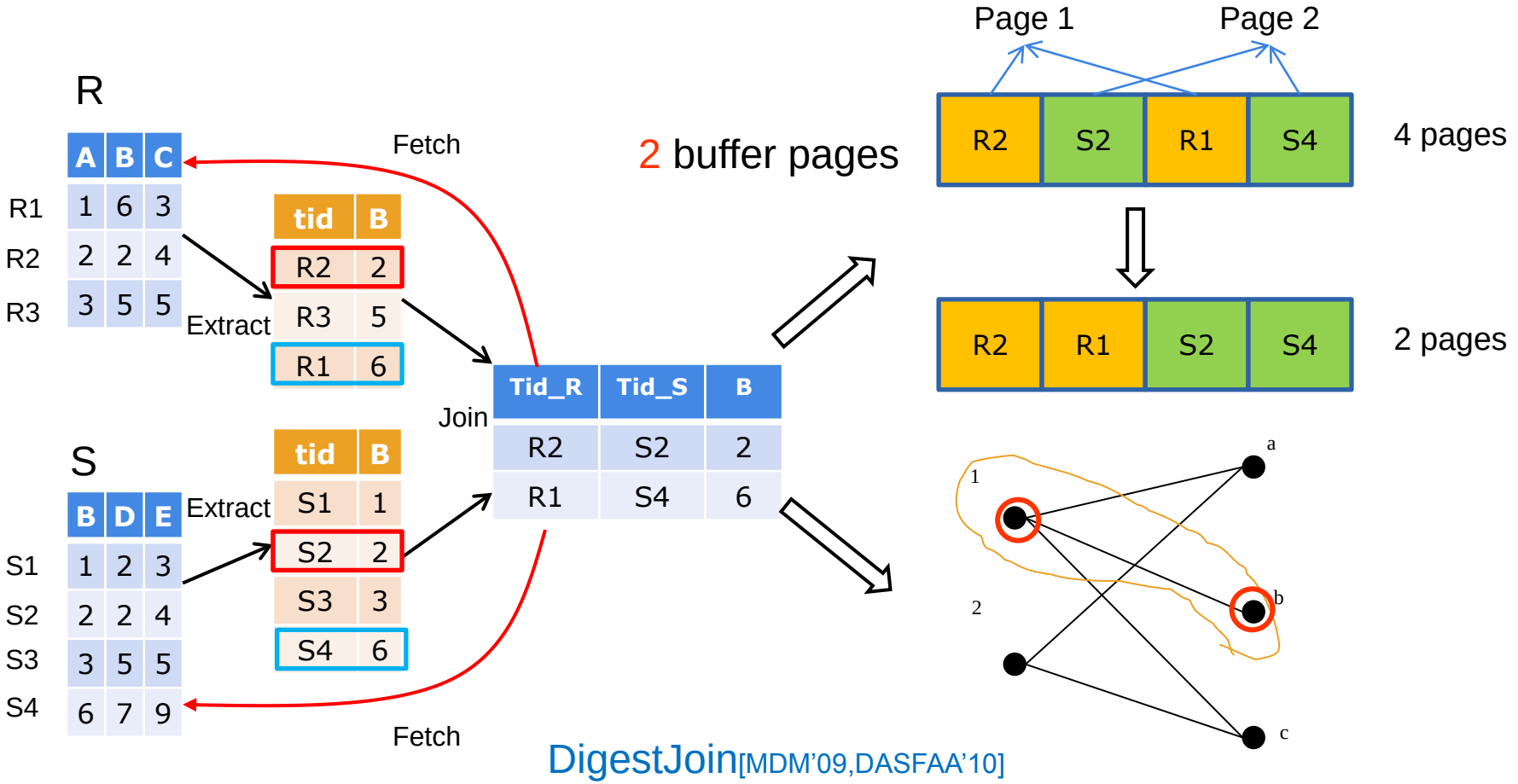


DigestJoin[MDM'09,DASFAA'10]

DigestJoin [MDM2009]

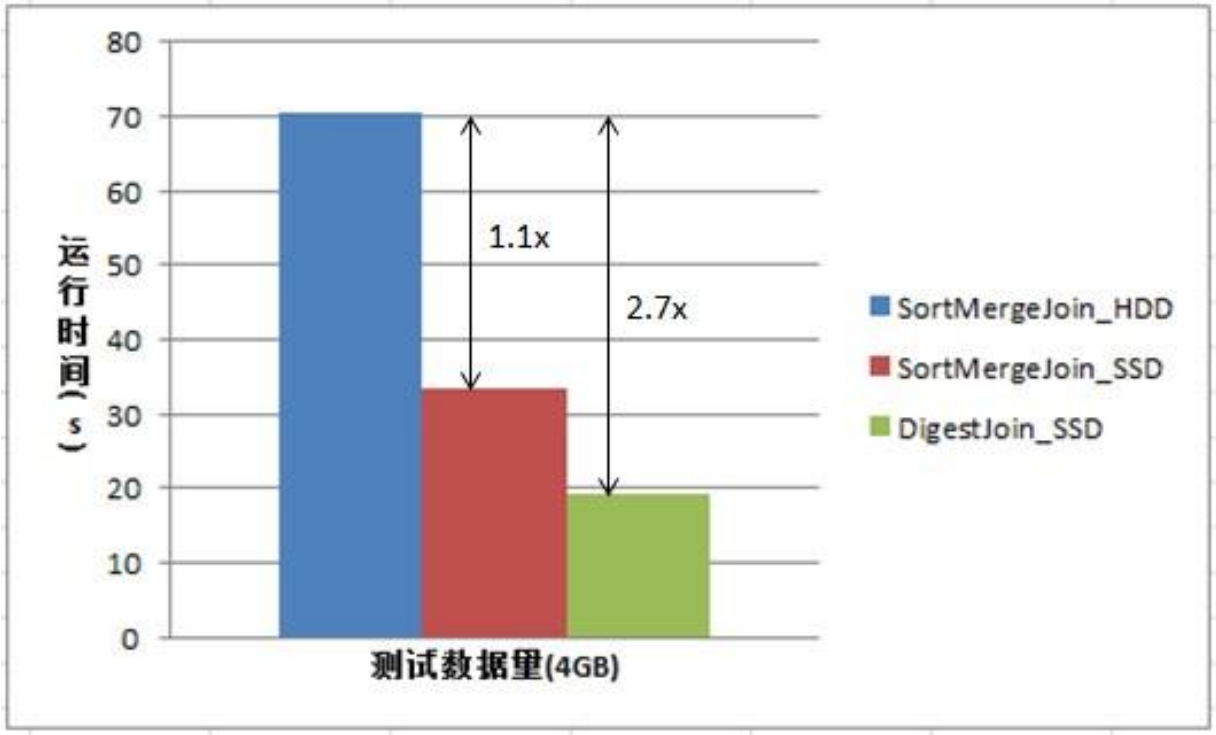


DigestJoin [MDM2009]

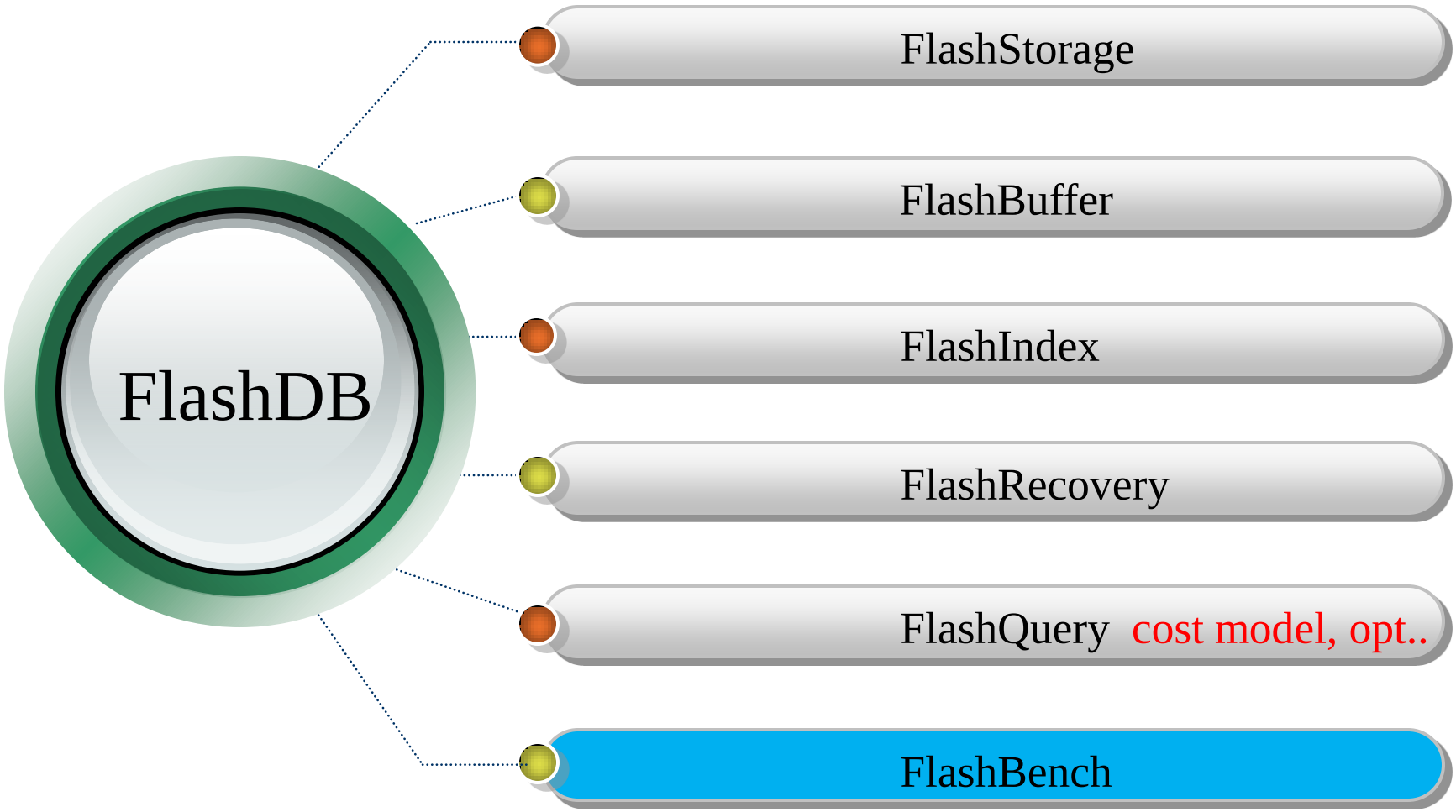


Performance Evaluation

- PostgreSQL with DigestJoin



Research on FlashDB



Flash Devices

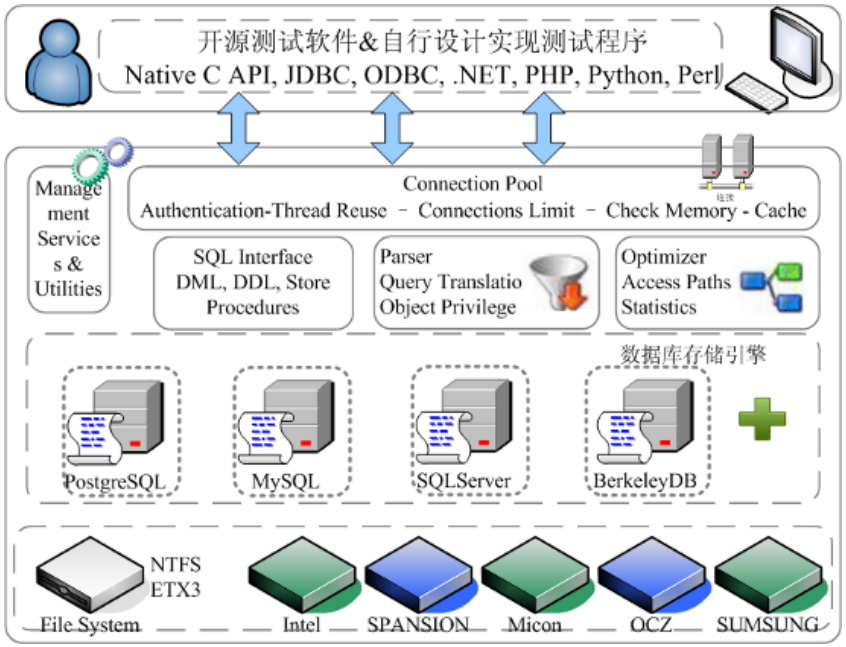
- Flash Board
  - 高速SSD的设计验证
  - 底层FTL算法的验证
  - 多芯片并行存取验证
  - Inside-SSD Cache算法验证

Flash Devices

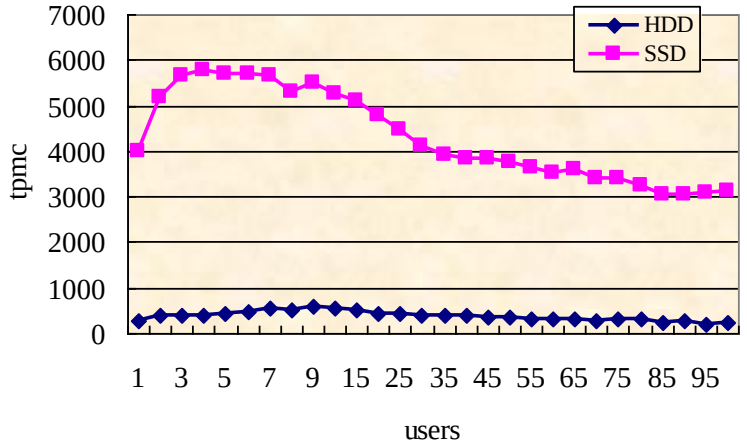
- Flash Board
  - 高速SSD的设计验证
  - 底层FTL算法的验证
  - 多芯片并行存取验证
  - Inside-SSD Cache算法验证

- FlashDBSim
  - 统一、可配置、易于使用的闪存设备仿真平台
  - 可以模拟不同类型闪存特性(SLC/MLC/NOR)
  - 提供I/O统计信息(write/read/erase)

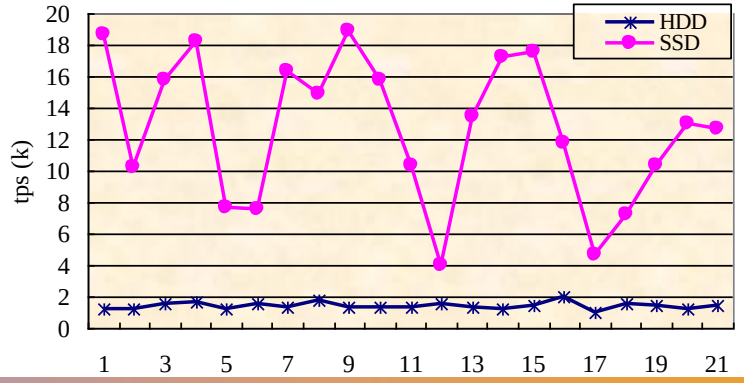
闪存数据库系统基准测试环境



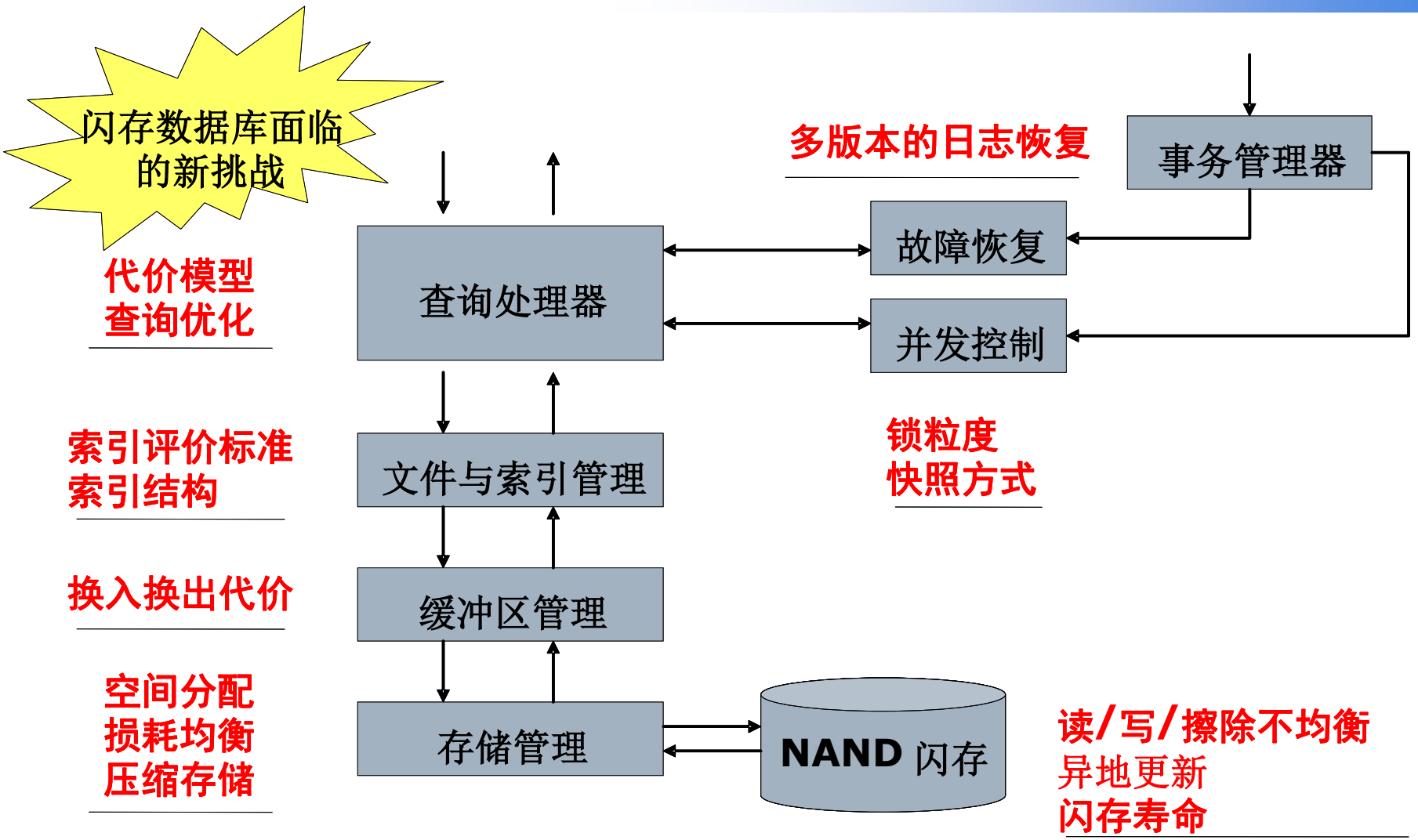
Tpmc of PostgreSQL on SSD and HDD
(warehouses=10)



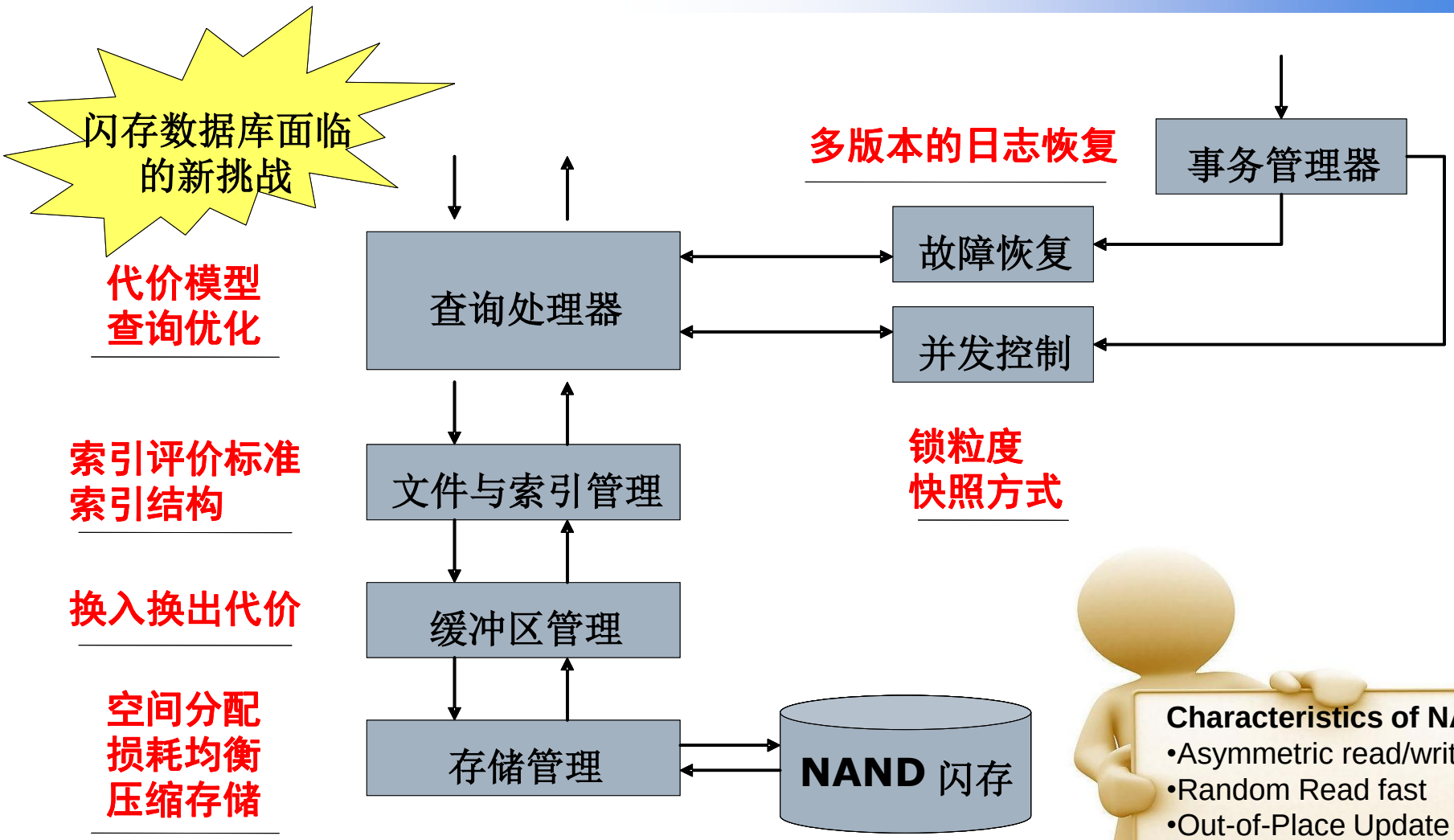
TPS of MySQL on SSD and HDD
(users=200)



闪存数据库系统


闪存数据库面临的新挑战
多版本的日志恢复
事务管理器
故障恢复
查询处理器
并发控制
代价模型
查询优化
索引评价标准
索引结构
文件与索引管理
锁粒度
快照方式
换入换出代价
缓冲区管理
空间分配
损耗均衡
压缩存储
存储管理
NAND 闪存
读/写/擦除不均衡
异地更新
闪存寿命

闪存数据库系统


闪存数据库面临的新挑战
代价模型
查询优化
索引评价标准
索引结构
换入换出代价
空间分配
损耗均衡
压缩存储
查询处理器
文件与索引管理
缓冲区管理
存储管理
NAND 闪存
多版本的日志恢复
事务管理器
故障恢复
并发控制
锁粒度
快照方式
Characteristics of NAND
•Asymmetric read/write
•Random Read fast
•Out-of-Place Update
•Energy efficient

Outline

1. New Storage
2. Flash-based DBMSs
3. SSD Hybrid Systems
4. Future Work

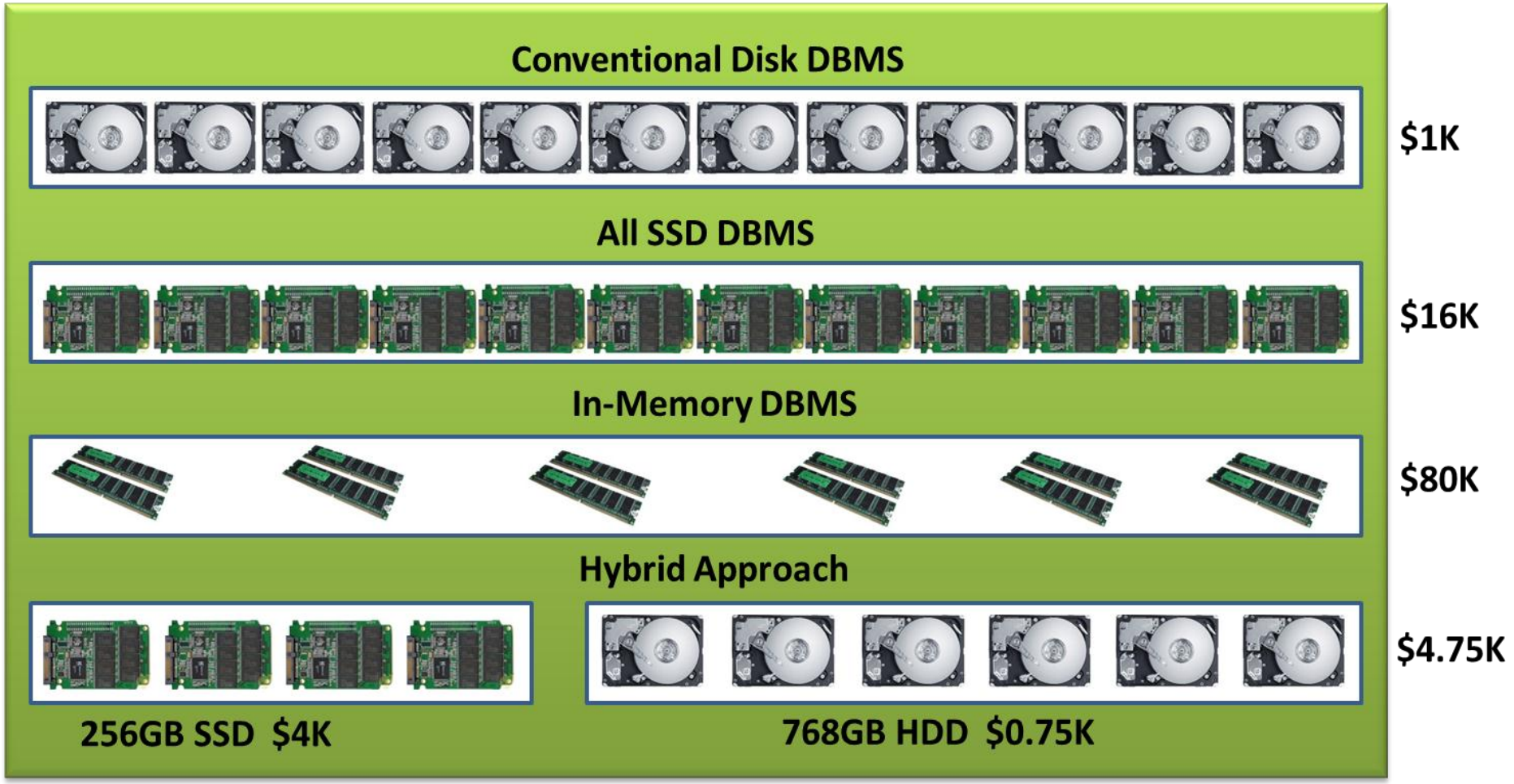
SSD Applications

- Widely used in various IT companies
  - Massive data volume
  - High throughput
  - Low latency
  - ...

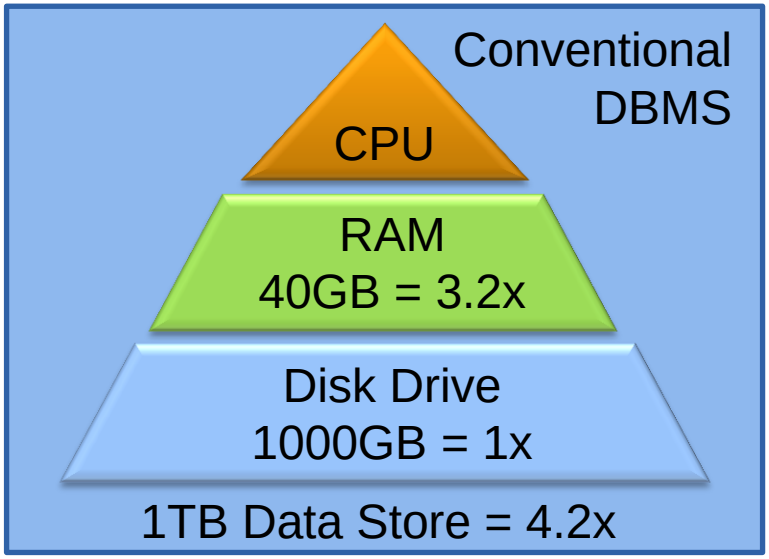
Can SSD Completely Replace Disk?

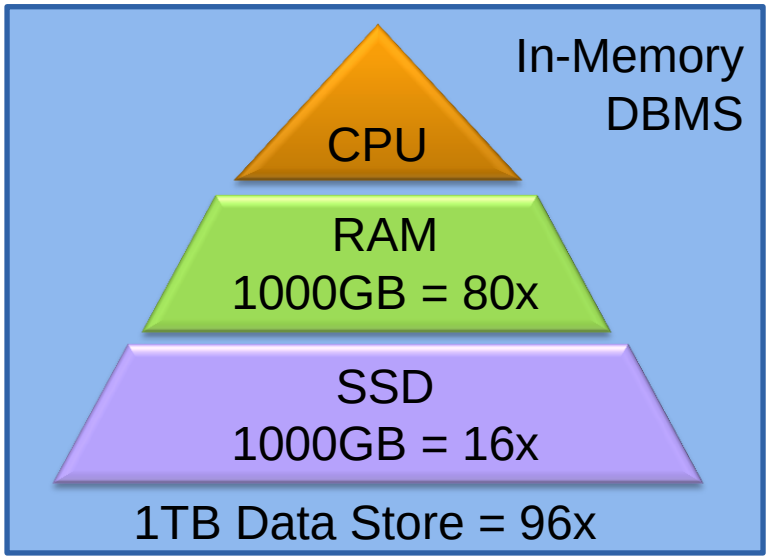
Comparison of Data Storage Approaches

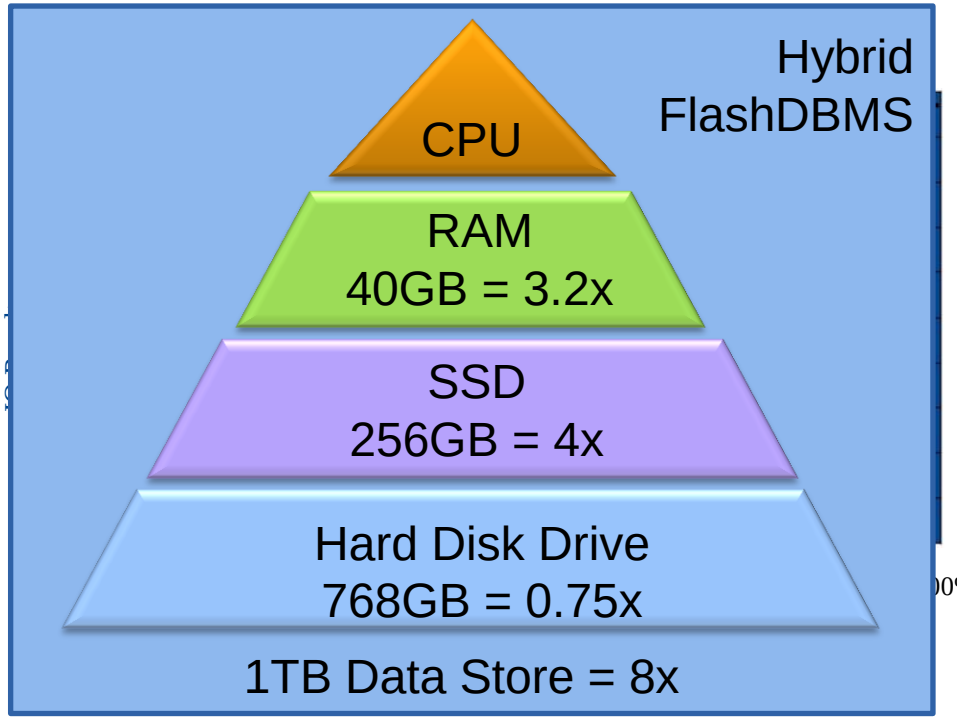
1T Data



Hierarchical Storage

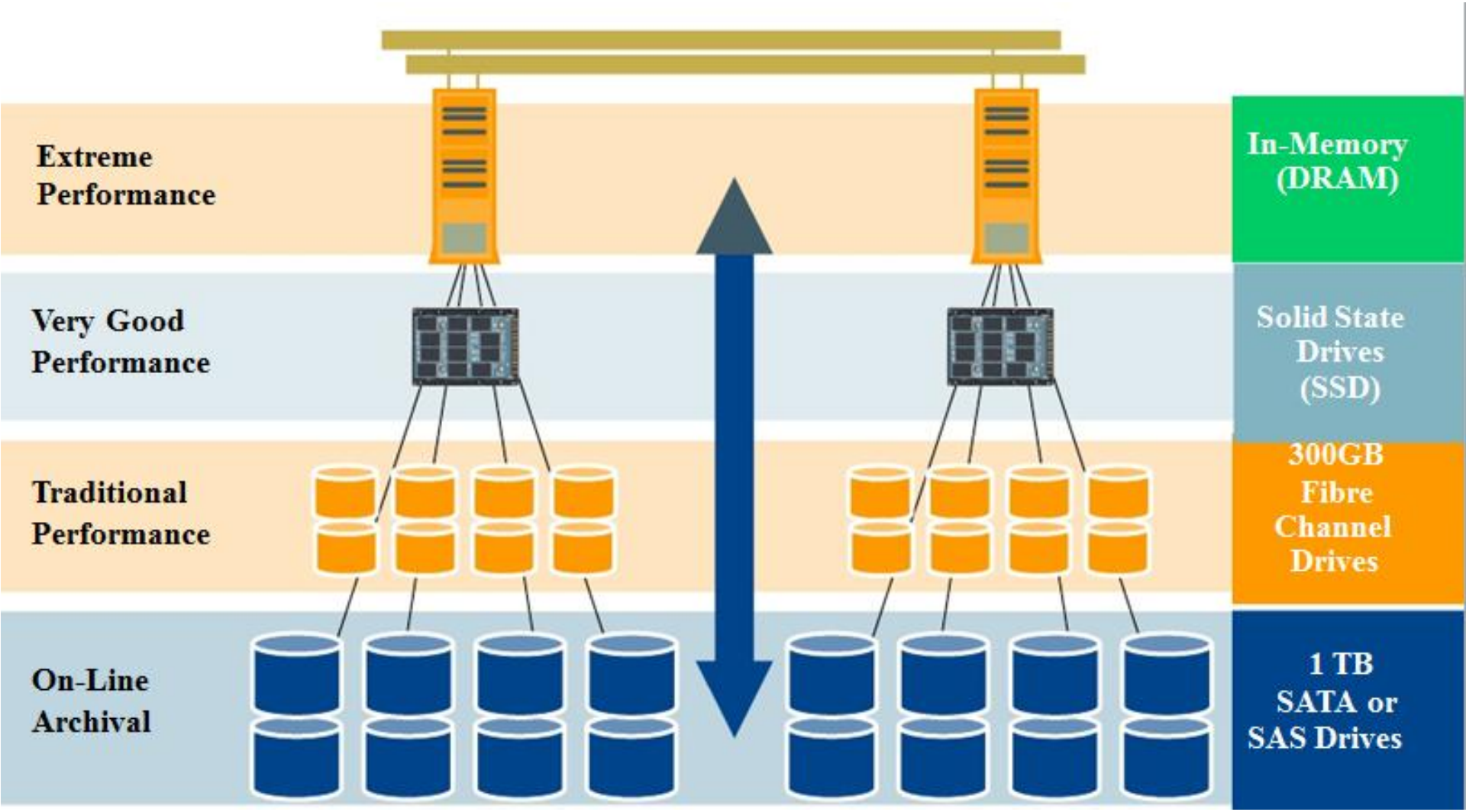




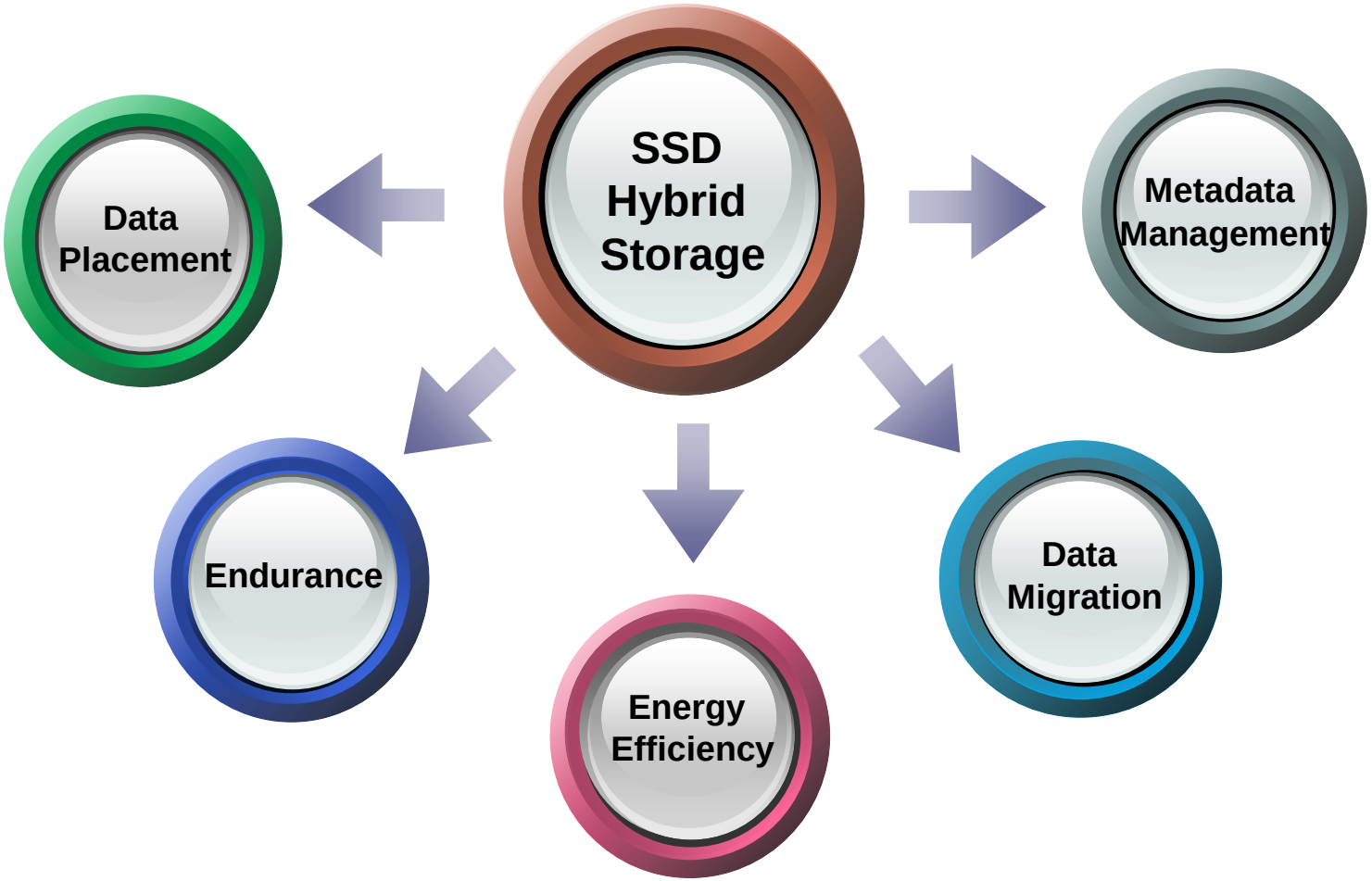


Is all of your data worth 25x ?

View of the Near Future: Leverage SSD for Big Data

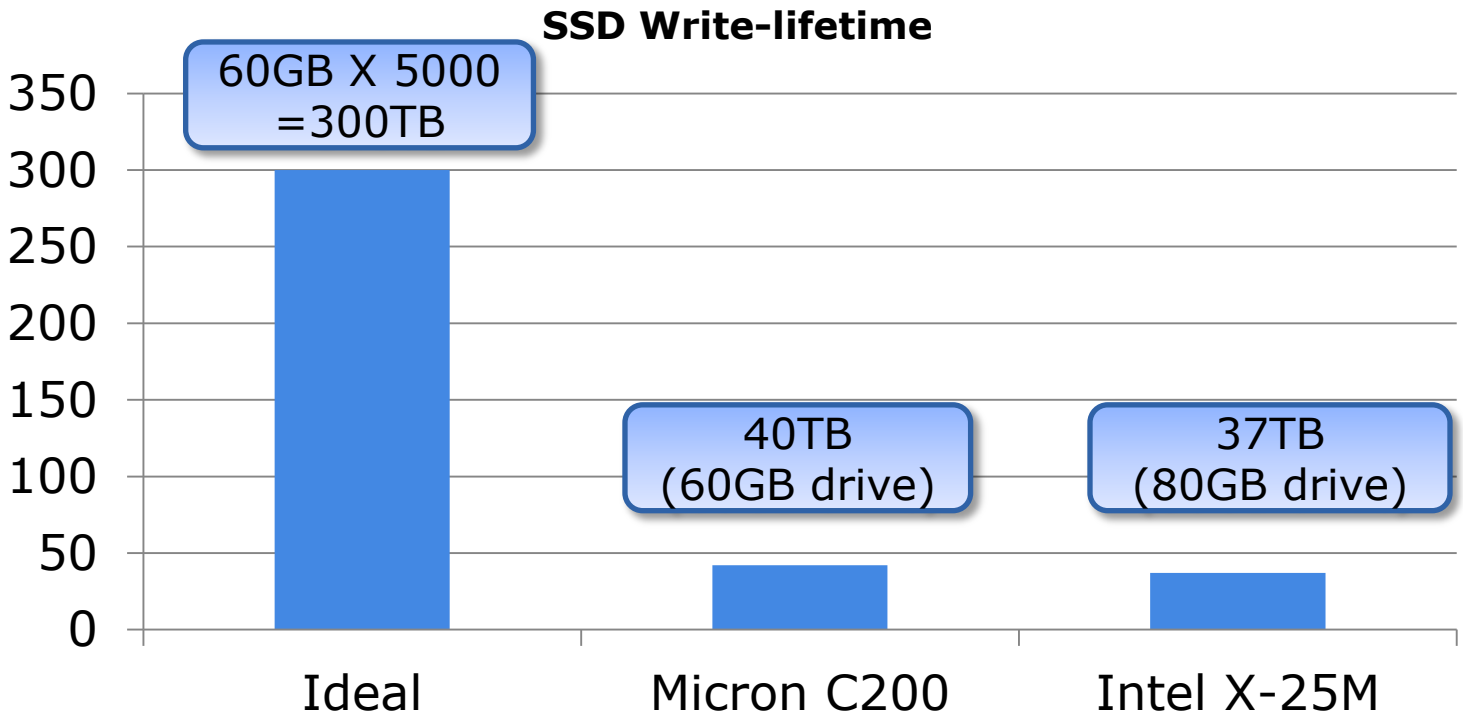


Hybrid FlashDB



Endurance

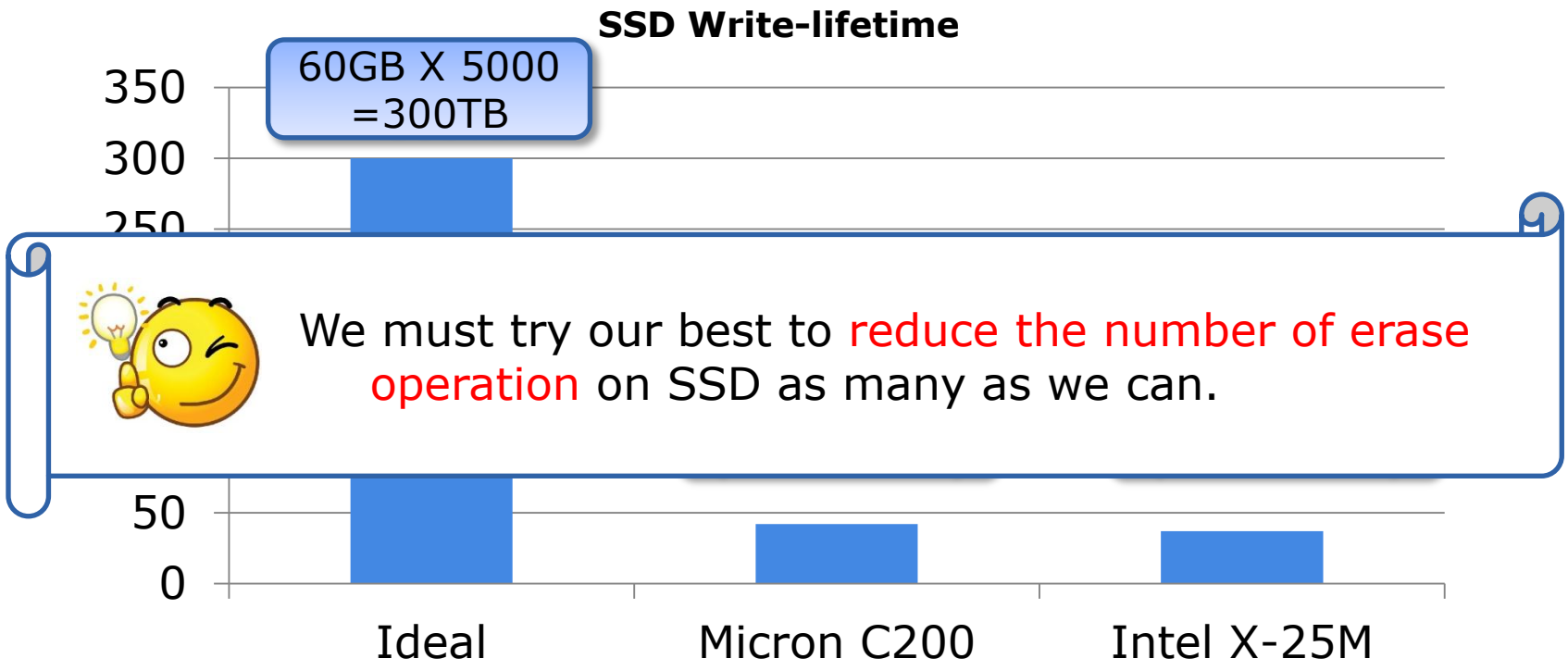
- SSD Write-lifetime



Extending SSD Lifetimes with Disk-Based Write Caches FAST'10

Endurance

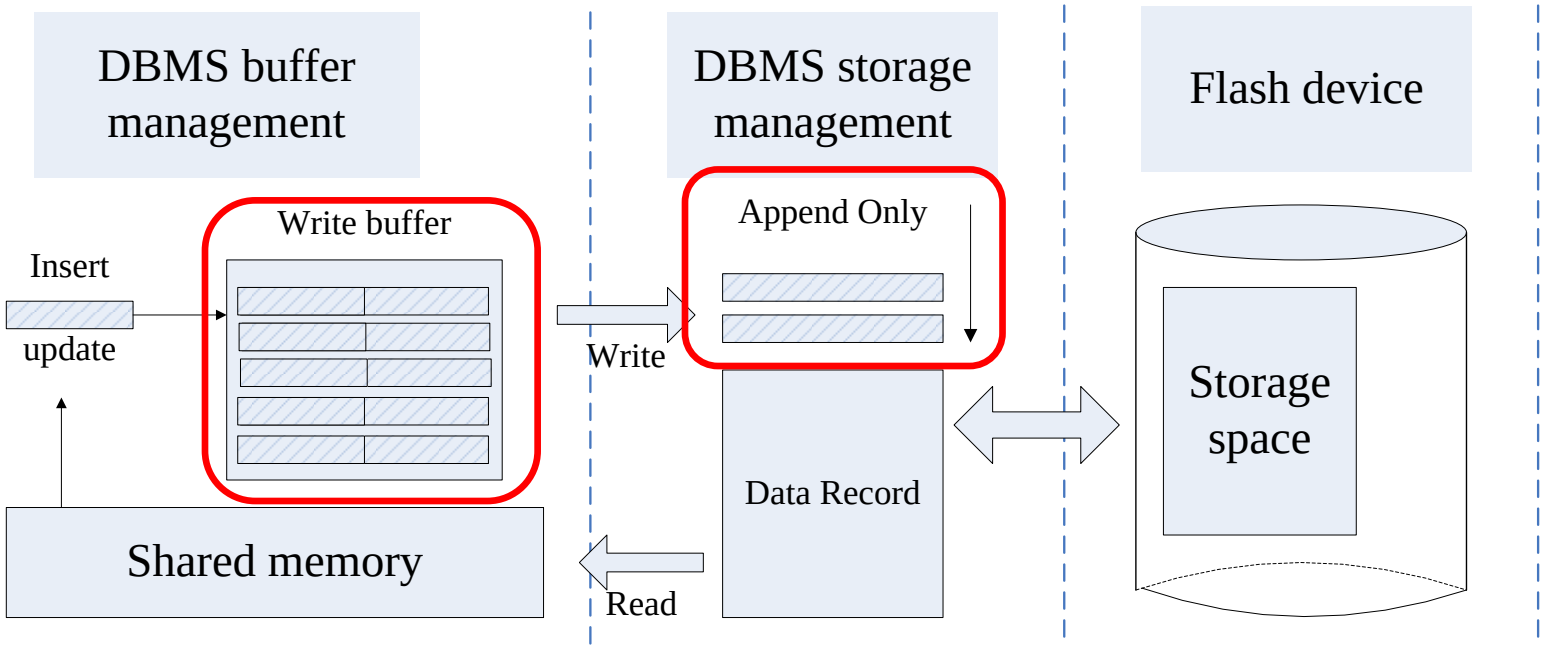
- SSD Write-lifetime



Extending SSD Lifetimes with Disk-Based Write Caches FAST'10

Extend SSD lifetime [FlashDB2011]

- Motivation
  - Small write → storage utilization declines
  - Random write → frequent erase
- Key idea



Extend SSD lifetime [FlashDB2011]

- Experiment Setup
  - Fexible simulator
  - Page size:2KB
  - Block size: 128KB
- Performance

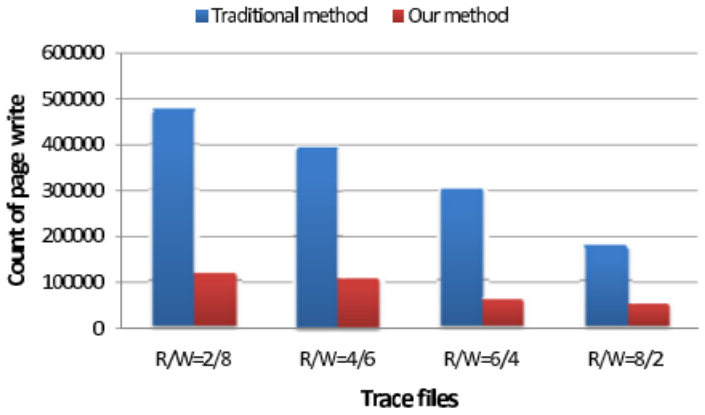


Fig. 4. Write Count for Comparison

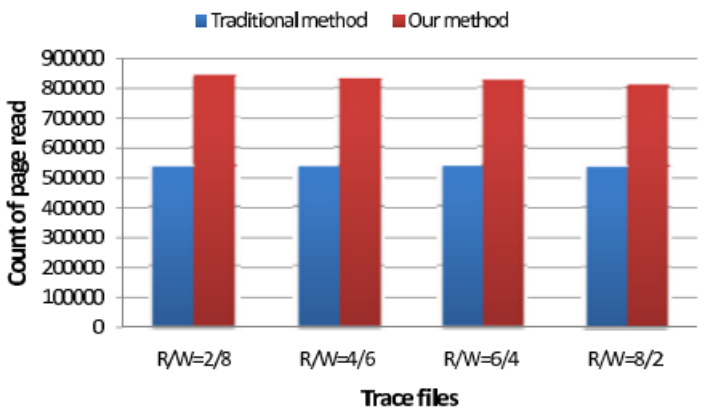
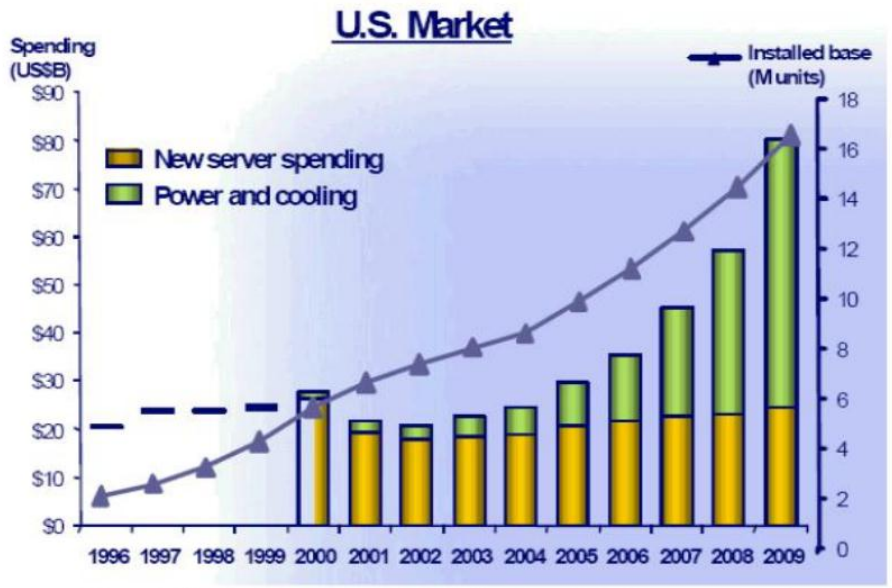


Fig. 5. Read Count for Comparison

Energy Efficiency

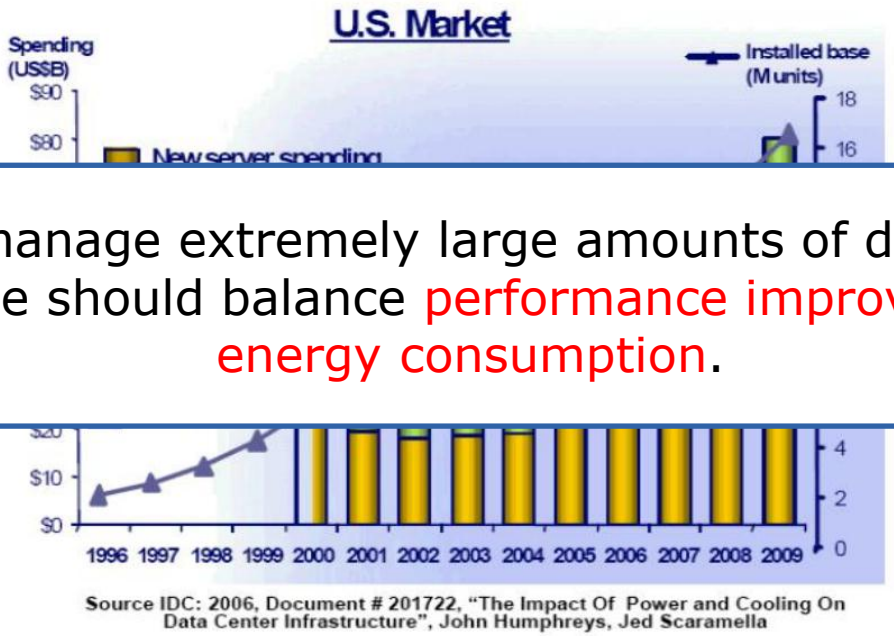
- Number of server installations is rapidly increasing
- The spending on power and cooling exceed server purchase cost



Source IDC: 2006, Document # 201722, "The Impact Of Power and Cooling On Data Center Infrastructure", John Humphreys, Jed Scaramella

Energy Efficiency

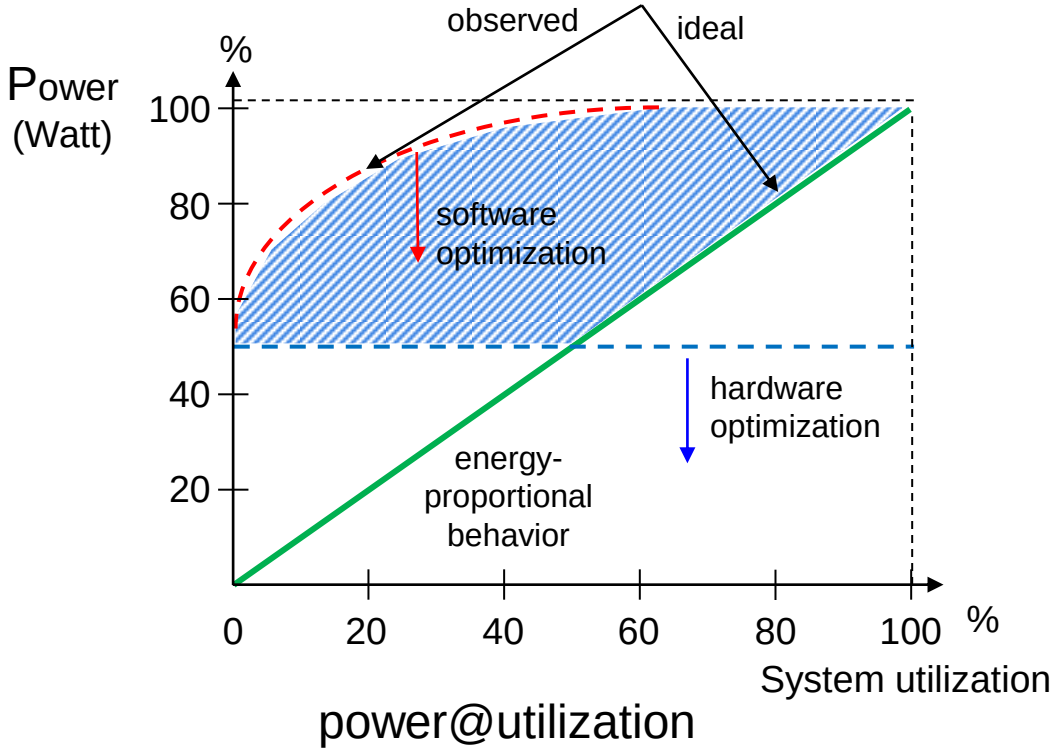
- Number of server installations is rapidly increasing
- The spending on power and cooling exceed server purchase cost



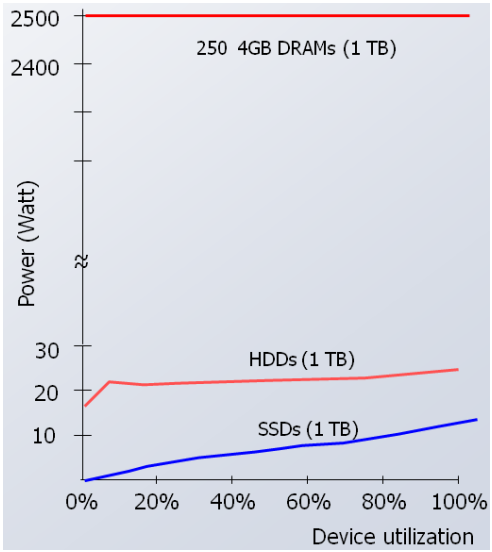
Source IDC: 2006, Document # 201722, "The Impact Of Power and Cooling On Data Center Infrastructure", John Humphreys, Jed Scaramella

To manage extremely large amounts of data efficiently, we should balance performance improvement and energy consumption.

Energy Efficiency

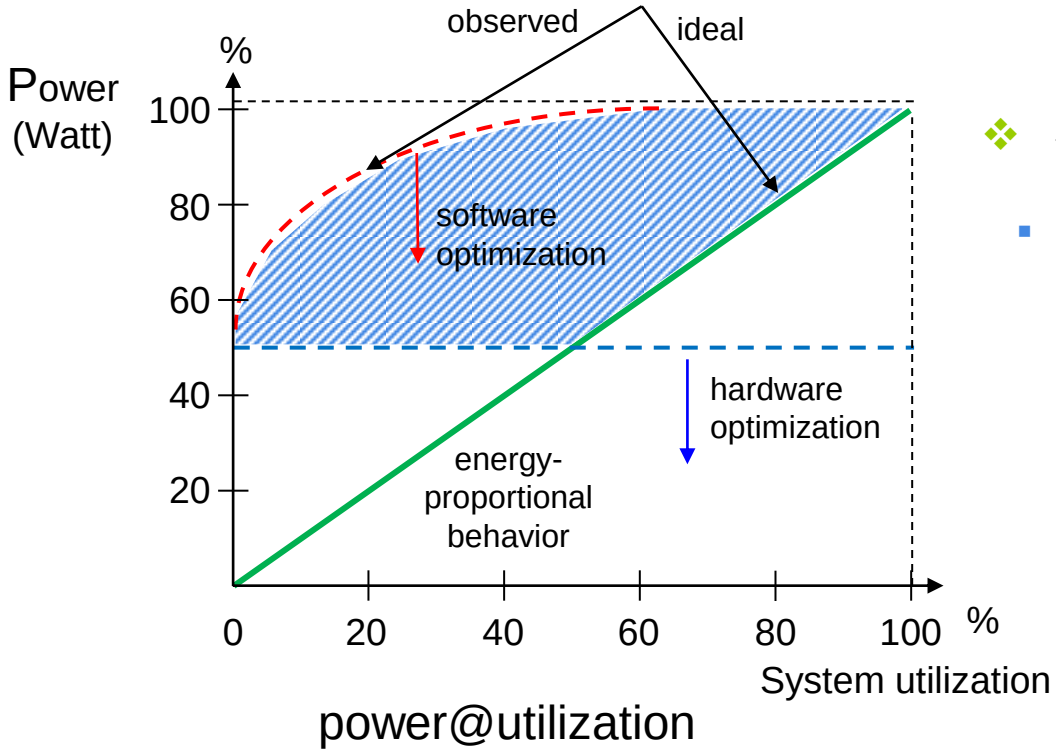


- Hardware Optimization



- SSD can reduce the use of energy-inefficient RAM-based memory without compromising the overall system performance

Energy Efficiency



power@utilization

- Software Optimization
  - Buffer Management
    - Trading Memory for Performance and Energy by Yi Ou [FlashDB2011]
  - ….

Outline

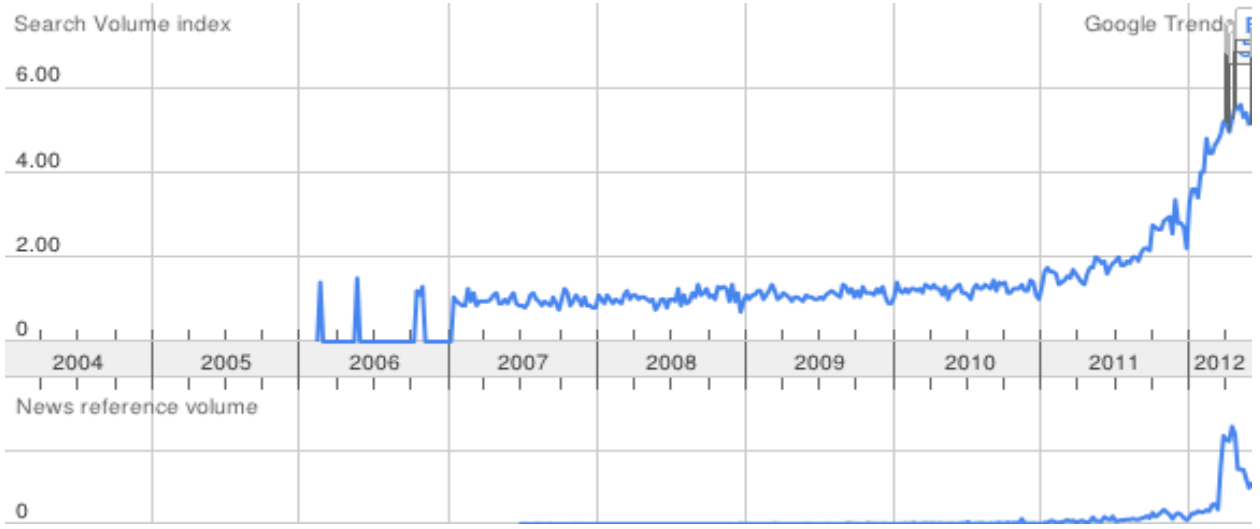
1. New Storage Era
2. Flash-based DBMSs
3. SSD Hybrid Systems
4. Future Work


the Voice of China
中国好声音


the Big Data of China
中国大数据

Big Data is so hot!

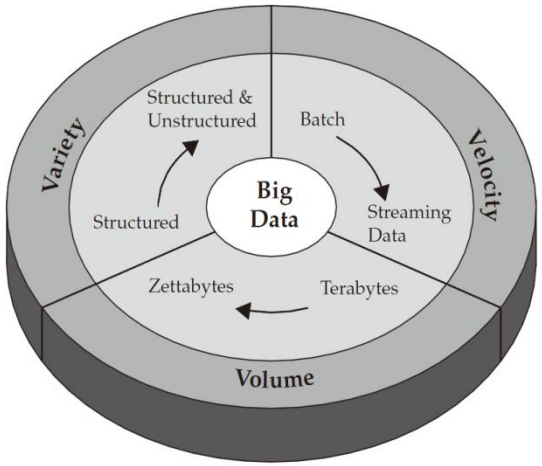
- Google Trends of Big Data



- Big Data Across the Federal Government (USA, March, 2012)

安阳殷墟遗址（公元前1300，距今3300年）

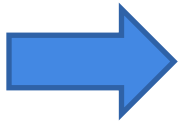
这就是大数据！



甲骨文大坑，
1万7千余片

DB(Database) vs. BD(Big Data)

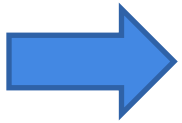
- “Small data”,
  Very Large Database
  （VLDB）
  - MB, 结构数据，
  - 运营式系统，封闭数据源
  - 以数据为对象解决其存储和管理问题



数据工程

Data Engineering

- Big Data，
  Extremely Large Database
  （XLDB）
  - >PB，非结构数据，
  - 感知式系统，开放数据源
  - 以数据为资源解决诸领域问题



数据思维

Data Thinking

Big Analytics

- Many situations need the result of analysis immediately

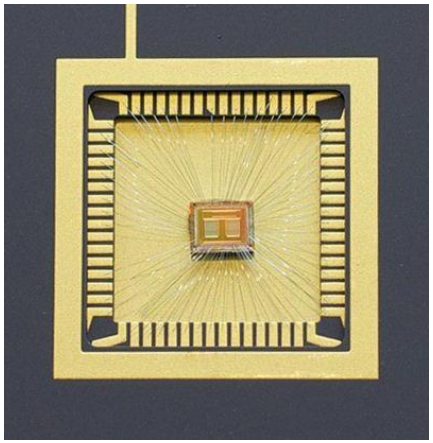
- Parallelism
  - Parallelism across nodes in a cluster
  - Parallelism within a single node
- Cloud Computing
- New hardware: SSD、PCM…

Storage Class Memory (SCM)

- A new class of data storage/memory devices
  - many technologies compete to be the 'best' SCM
- SCM *blurs the distinction* between
  - Memory (*fast, expensive, volatile*) and
  - Storage (*slow, cheap, non-volatile*)
- SCM features:
  - Non-volatile
  - Short access times (~ DRAM like)
  - Low cost per bit (disk like – by 2020)
  - Solid state, no moving parts

Phase change memory

- Phase change memory (PCM) is the leading contender for first true SCM.
- At least 18 companies are working on PCM, such as IBM, Samsung, Intel, Micro, etc.
- PCM is an electronic device using two distinct solid phases metal alloy to store a bit.



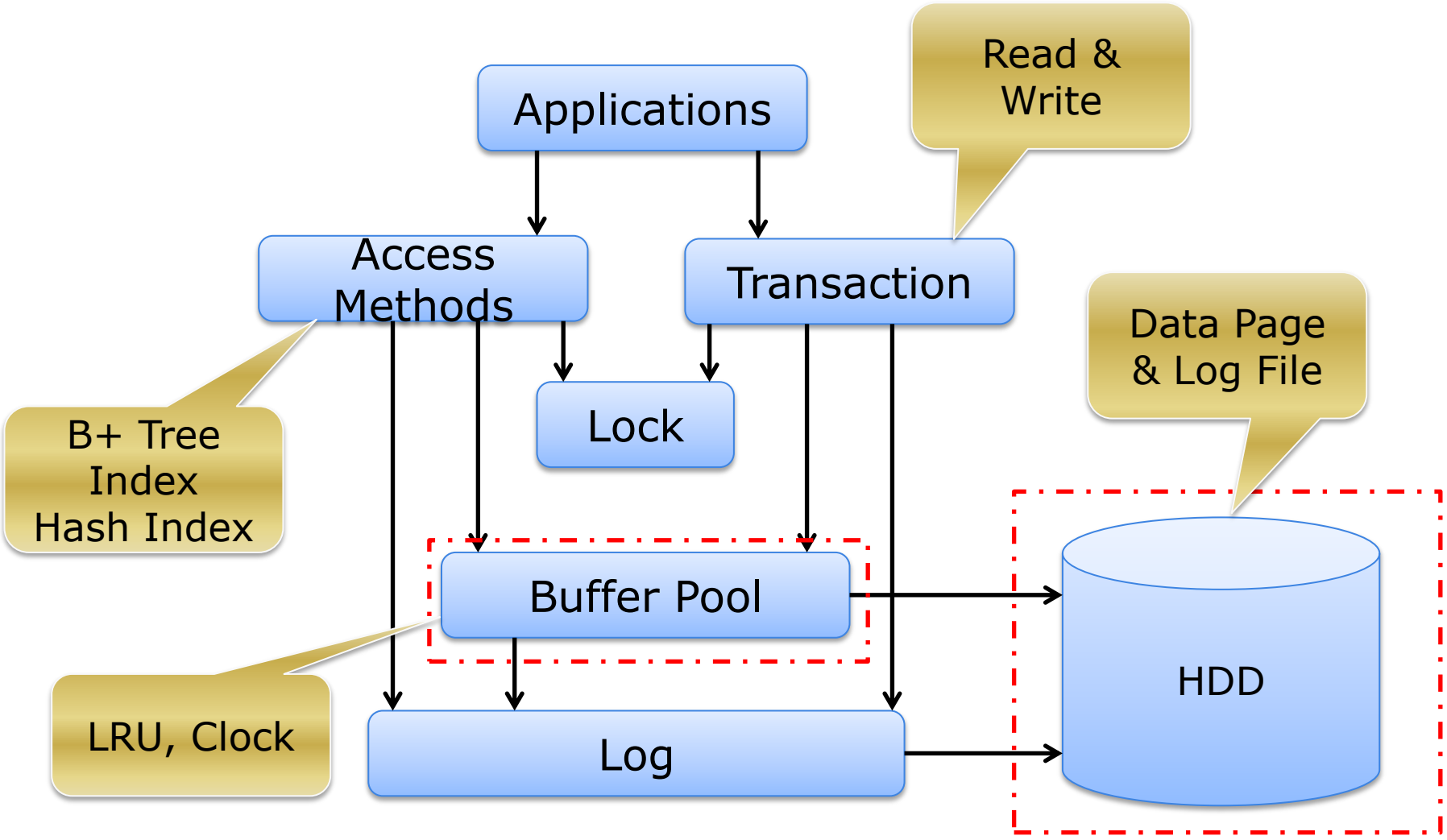
ISSCC: Samsung preps 8-Gbit phase-change memory

Peter Clarke

11/29/2011 6:28 AM EST

LONDON – Samsung Electronics Co. Ltd. is set to re-ignite debate about whether phase-change memory is commercially viable with the presentation of an 8-Gbit, 20-nm device at the 2012 International Solid-State Circuits Conference.

The Impacts of PCM on DBMSs



The Impacts of PCM on DBMSs

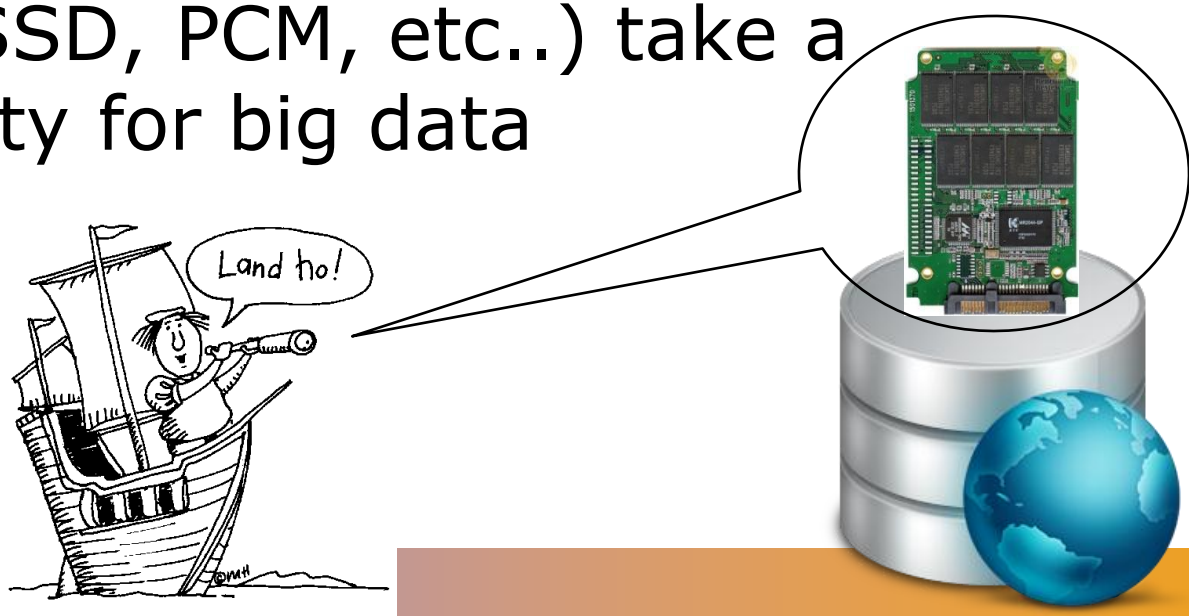
- In-memory buffer pool can be obviated, or at least read buffer can be obviated?
- What about logging? Logging is still necessary?
- Opportunity to rethink data structures for implementing database system, such as B+ Tree, record organization, etc.
- Even Opportunity to rethink the Database Machines……

"Bring data to computation"

➔ **"Bring computation to data"**

Conclusion

- Flash devices open the new world for DBMSs
  - Buffer(ACR), Join(DigestJoin), ShadowPage,….
  - And, there are a lot of research topics still ahead (at least for the coming 5 years) and thus you can jump on the flash-based database researches.
- New storage(SSD, PCM, etc..) take a new opportunity for big data management



SELECT thanks FROM me
SELECT questions FROM you

Tape is Dead
Disk is Tape
Flash is Disk

--Jim Gray, 1998

致 谢

本报告的工作得到了国家自然科学基金重点项目“闪存数据库技术研究”(60833005)的资助

About our Lab

- Innovative Data Management Research
- Http://idke.ruc.edu.cn
- Google wamdm