Banish the I/O: Together, SSD and Main Memory Storage Accelerate Database Performance

Today's Presentation

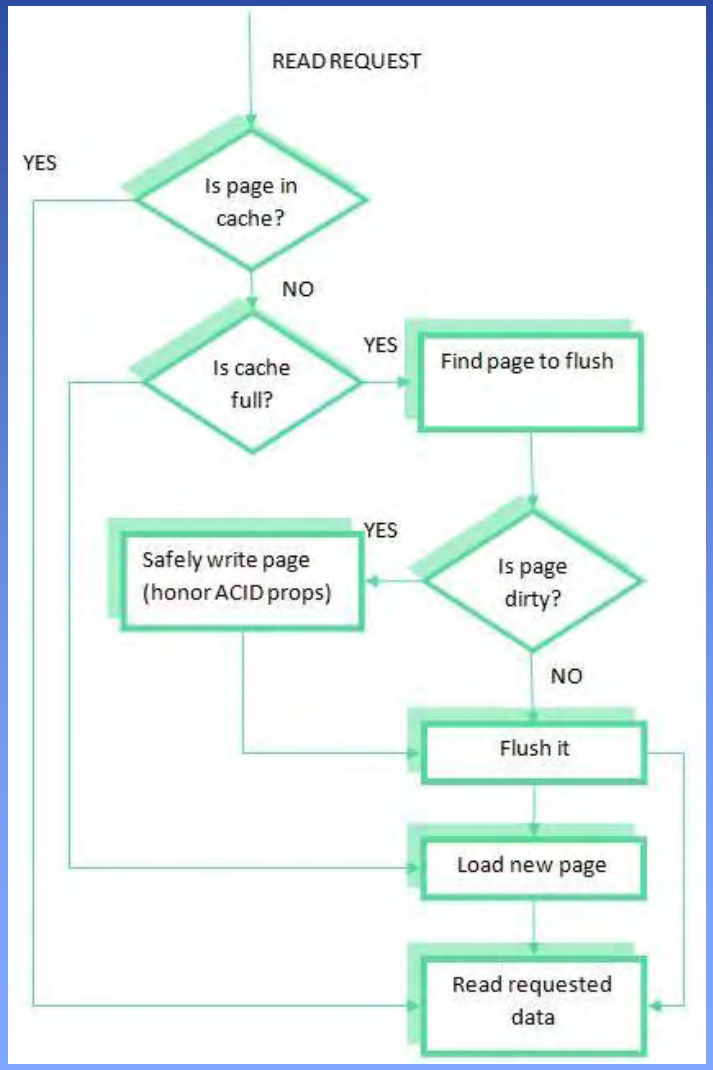
- Conventional Database Performance Optimization Goal:
  Minimize I/O
- Legacy Approach: Cache
- 21st Century Approaches:
  A. Solid State Disk (SSD)
  B. In-Memory Database System (IMDS)
- Combine A & B For The Best Of Both Worlds

Cache – A Quick Review

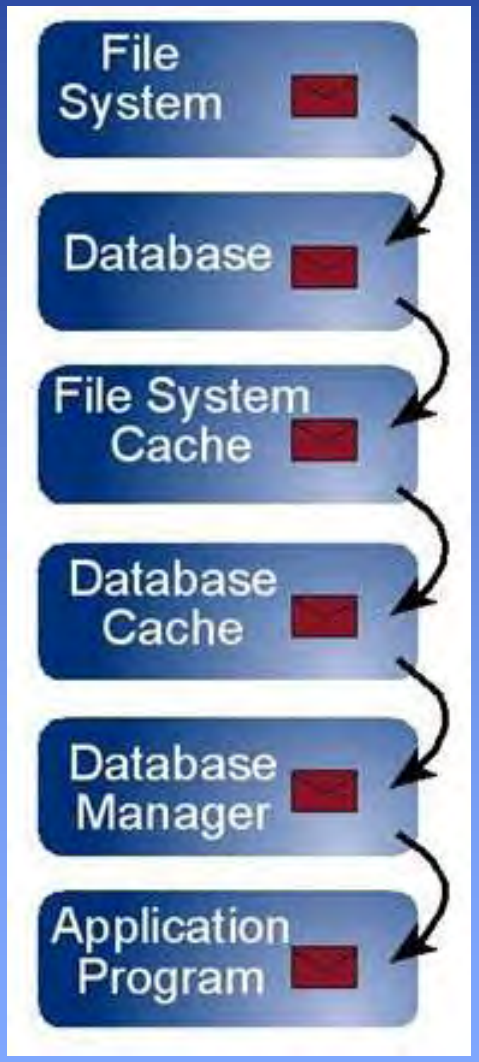
- A database cache is nothing more than an optimization strategy to minimize (not eliminate) file I/O
- Assumptions:
  - Trading memory consumption for file I/O is reasonable
  - Trading CPU cycles for file I/O is reasonable
  - Trading “storage space” for file I/O is reasonable
- Net Effect: Conventional DBMS use memory (and storage) and CPU inefficiently
- Cache only improves read performance, not write performance

Cache – A Quick Review

Cache Logic



Copies and Transfers



Precision data management

Databases and SSD

- NAND Flash prices are dropping, but still relatively costly
- Could be cost-competitive with some HDD in 2010
- But NAND Flash memory quality is dropping, too
  - 90nm-generation technology in 2004-05 offered 100,000 rewrites, data retention of ~10 years
  - 30nm 2-bit/cell chips have no more than 3,000 cycles, data retention ~1 year
  - 3-bit/cell chips have only a few hundred rewrites
- Facts above somewhat offset by greater capacity

Source:
SSDs Challenge HDDs, but Quality a Problem
Nikkei Electronics Asia, June 2009

Databases and SSD

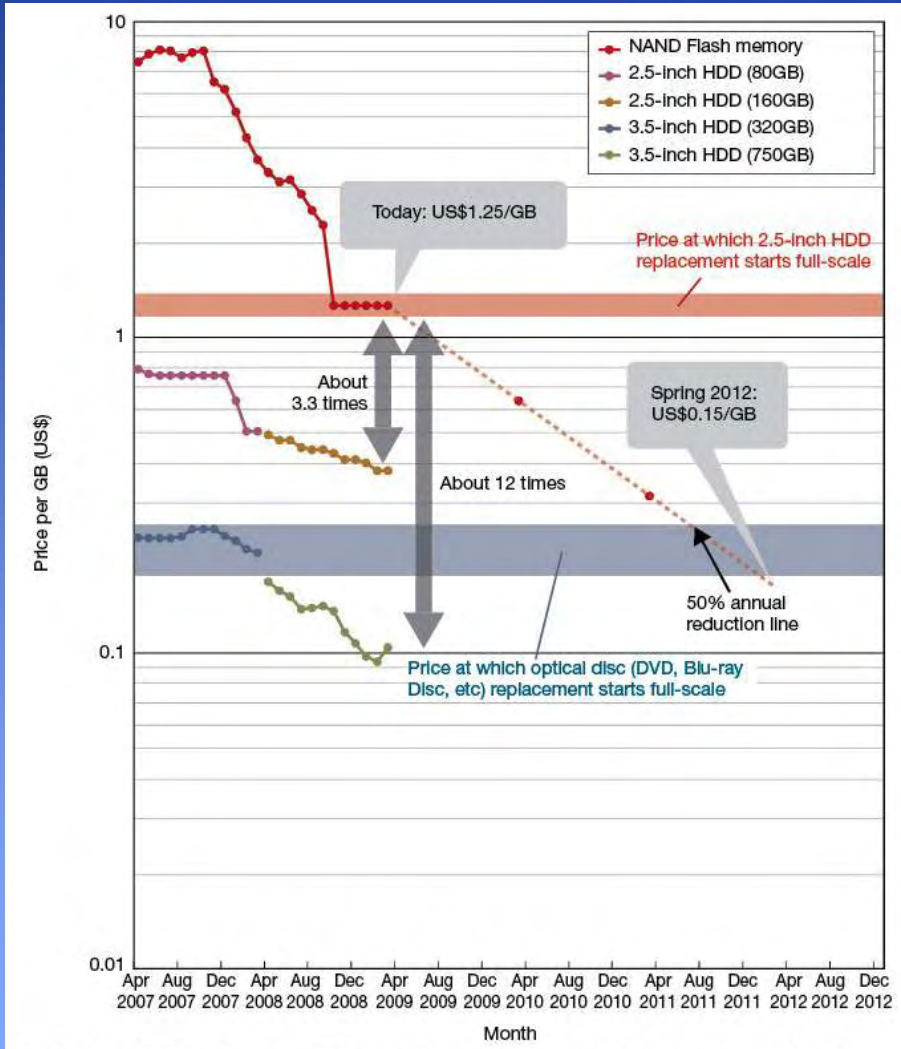


Fig 6 Per-GB Price Down to US$1.25 Diagram plots the price-per-GB for NAND Flash memory and HDDs. As of Mar 2009, NAND Flash memory is about US$1.25 per GB. Prices through Mar 2009 from *Nihon Keizei Shimbun*, beyond that forecasts by *Nikkei Electronics*.

Source:
SSDs Challenge HDDs, but Quality a Problem
Nikkei Electronics Asia, June 2009

Databases, SSD, and HDD

- SSD access time is .2 - .3 milliseconds
- SSD Transfer Rate up to ~150 MB/s
- HDD access time is ~9 milliseconds
- HDD Transfer Rate ~64 MB/s

Source: Transcend Information, Inc.

Databases and SSD

Pros

- SSD is transparent to the DBMS
- Improves query and insert/update/delete performance
- Low power

Cons

- NAND Flash has short life expectancy
- DBMS cause a lot of writes
  - To keep indexes balanced
  - For transaction rollback/rollforward
  - For isolation
- Faster than HDD, but slower than RAM

In-Memory Database Systems

- In-memory optimization:
  - No file I/O to minimize, no cache needed
  - Storage space (memory) is not abundant and cheap
    - Minimize memory consumption
  - Speed is the ultimate objective
    - Minimize CPU cycles

Copies and Transfers

In-Memory Database Systems

Pros

- Best query/insert/update/delete performance possible
- Store more data in less “storage space”

Cons

- Data is in volatile RAM
- Must be an IMDS (i.e. not Oracle 11g, SQL Server, etc)

IMDS and SSD For Transaction Logging

- Keep IMDS for maximum performance
- To offset RAM volatility, journal transactions to SSD for durability
- Lose “some” performance on insert/update/delete
- Still faster than conventional DBMS on SSD
- Maximize life expectancy of SSD (fewer write/erase cycles)

Write Performance Comparison

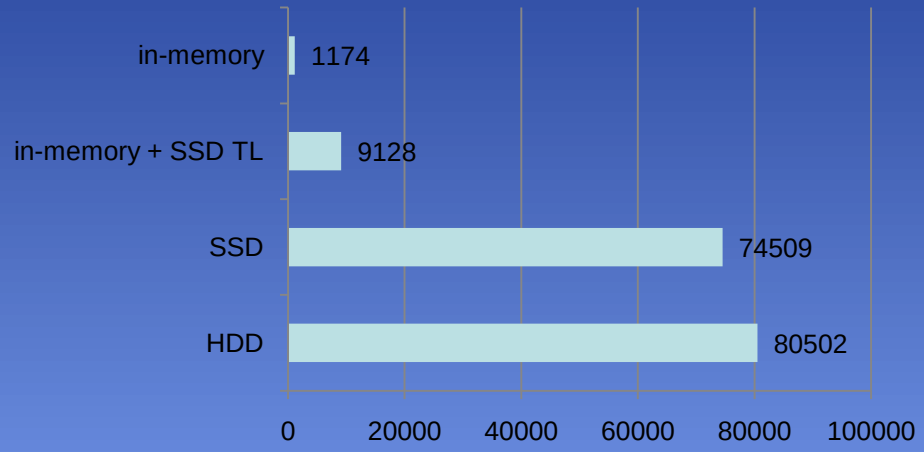
*eXtreme*DB-TL Transaction:
Exactly 1 File I/O regardless of the size of the transaction

Disk-based Database:
Assume a transaction involving one row in each of two tables, each with one b-tree index 5 levels deep:

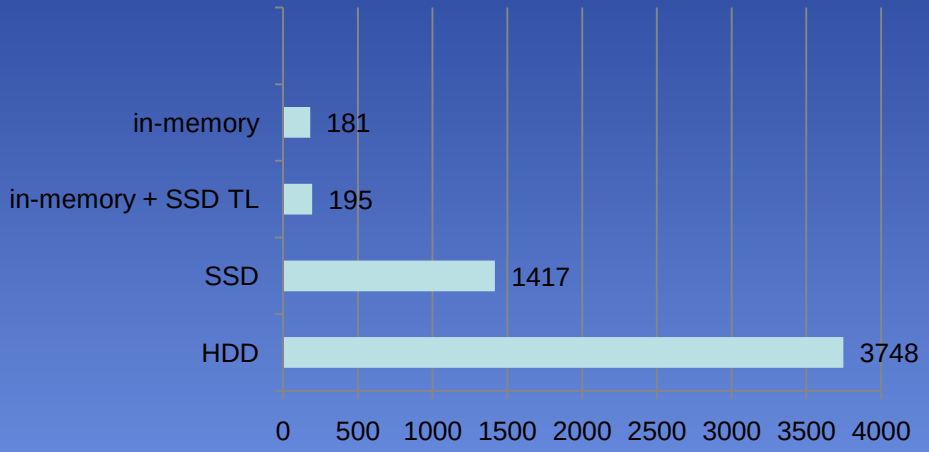
- 1 file write for Table A row
- 1 file write for Table B row
- 3 file reads + 1 to 3 file writes for Table A index
- 3 file reads + 1 to 3 file writes for Table B index
- ? file writes for transaction log
- 4 to 8+ total write operations.

Database Performance - 200,000 objects
Transcend 32 GB SATA SSD,
Seagate Barracuda 160GB HDD

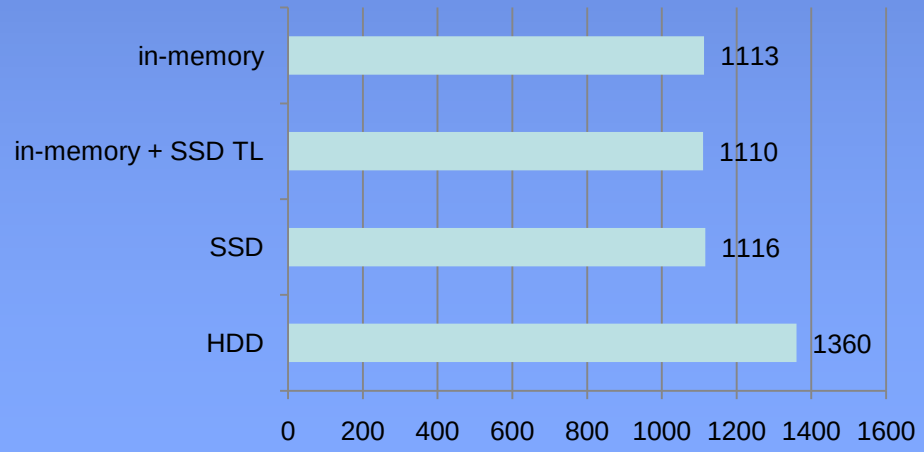
Insert



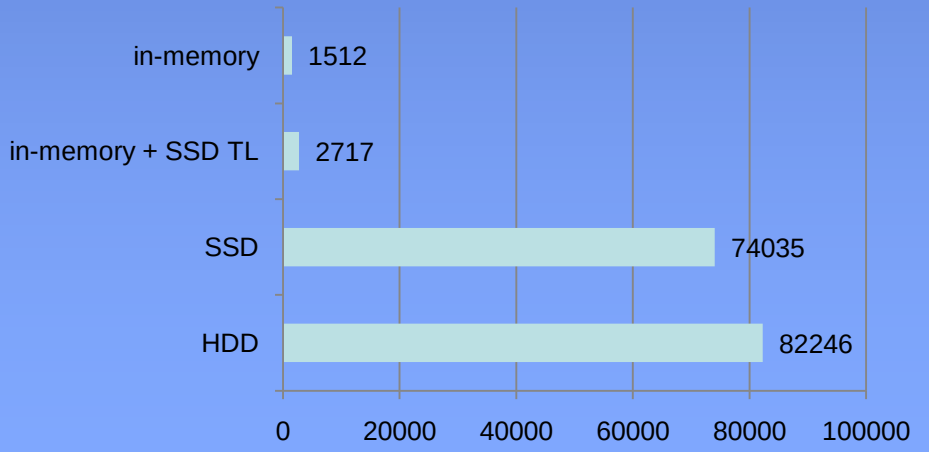
Hash Search



Tree Search



Delete



Summary

- SSD vastly outperform HDD on read, modestly on write
- Are becoming cost-competitive
  - May already be, factoring in heat & power considerations
- In-memory database system performance cannot be matched
- In-memory databases are subject to RAM volatility
- Combining IMDS and transaction logging to SSD yields performance better than conventional DBMS on SSD, without sacrificing durability