Using Memory-Tier NAND Flash to Accelerate In-Memory Database Transaction Logging

Presented by Steve Graves, McObject

In-Memory Database System (IMDS)

- Definition: database management system that stores all records in main memory
- Contrast to traditional DBMSs premised on disk storage for all data
- IMDSs eliminate
  - Disk and file I/O
  - Cache processing
  - Data transfer
- Result: IMDSs perform orders of magnitude faster
- Increasingly popular for business analytics, telecom, capital markets, industrial control and more

IMDSs, Volatility & Transaction Logging

- DRAM's volatility is viewed as an IMDS disadvantage – how can an IMDS gain data durability?
- Transaction Logging provided in most IMDS products
  - Logging enables recovery of committed transactions in the event of system failure
- Objection: this re-introduces writes to persistent storage

McObject's benchmark tests addressed the question, "Will an IMDS with transaction logging still outperform a traditional "on-disk" DBMS?" and measured the performance impact of different storage technologies

Benchmark Tests

- Performance measured for database inserts, updates, deletes, index searches & table traversals
- Across different storage technologies:
  - Hard disk (Western Digital VelociRaptor, 600 GB)
  - SSD (SanDisk Extreme Solid State Drive, 240 GB)
  - Memory-tier NAND flash (Fusion-io ioDrive2)
- Database systems compared:
  - *eXtreme*DB In-Memory Database System (w/ transaction logging feature enabled)
  - *eXtreme*DB Fusion on-disk DBMS
- Dell PowerEdge T110 Tower Server with 4GB of 1333 mhz memory

Test Results

- Database index searches & table traversals
  - Minimal impact on performance when moving from on-disk DBMS to IMDS w/ transaction logging (IMDS+TL), or when changing storage device
  - These database “reads” are typically much less costly, in performance terms, than writes (inserts, updates & deletes)
  - Ample system memory ensured that most read requests resulted in DBMS cache hits
- Database inserts, updates and deletes
  - A completely different story: Database system and storage type resulted in dramatic differences in performance, ranging as high as 2,300%

Test Results: DBMS & IMDS+TL Writes

Insert	Loops/ms	Perf. Multiple
HDD – On-disk	1.60	1.00
HDD – IMDS+TL	5.11	3.20
SSD – IMDS+TL	15.49	9.69
ioDrive2 – IMDS+TL	32.05	20.05

Update	Loops/ms	Perf. Multiple
HDD – On-Disk	3.00	1.00
HDD – IMDS+TL	5.32	1.77
SSD – IMDS+TL	17.30	5.77
ioDrive2 – IMDS+TL	38.25	12.75

Delete	Loops/ms	Perf. Multiple
HDD – On-Disk	1.50	1.00
HDD – IMDS+TL	5.31	3.55
SSD – IMDS+TL	17.77	11.87
ioDrive2 – IMDS+TL	34.72	23.19

Test Results, Cont.

Red = *Entire* Database Stored on SSD or ioDrive2

Insert	Loops/ms	Perf. Multiple
SSD – On-Disk	3.00	1.88
ioDrive2 – On-Disk	6.12	3.83
HDD – IMDS+TL	5.11	3.20
SSD – IMDS+TL	15.49	9.69
ioDrive2 – IMDS+TL	32.05	20.04

Update	Loops/ms	Perf. Multiple
SSD – On-Disk	7.23	2.41
ioDrive2 – On-Disk	15.99	5.33
HDD – IMDS+TL	5.32	1.77
SSD – IMDS+TL	17.30	5.77
ioDrive2 – IMDS+TL	38.25	12.75

Delete	Loops/ms	Perf. Multiple
SSD – On-Disk	2.98	1.99
ioDrive2 – On-disk	6.17	4.12
HDD – IMDS+TL	5.31	3.55
SSD – IMDS+TL	17.77	11.87
ioDrive2 – IMDS+TL	34.72	23.19

Why Is IMDS w/ Transaction Logging Faster Than On-Disk DBMS?

- On-disk DBMSs’ caching sub-system imposes performance overhead; an IMDS (with or without transaction logging) eliminates caching
- Widely used B-tree indexes accelerate some reads but are expensive for an on-disk DBMS to maintain during inserts/updates/deletes
  - B-tree overhead grows as database size increases
  - B-tree lookups are less costly with an IMDS: they impose no cache processing, happen at in-memory speed, and trees are shallower because they contain no duplicate index data
- Sequential writes (logging) vs. writes to random disk locations (on-disk DBMS writing through cache)

Discussion

- In-memory database system (IMDS) use is growing; some applications will require maximum speed *and* data durability
- Memory channel NAND flash greatly enhances performance of the most common IMDS data durability mechanism, transaction logging
  - Accelerated database writes by approximately 600% - 700% compared to hard disk transaction log storage
- Could store on-disk DBMS on memory channel NAND flash or SSD, but IMDS+TL is faster (4.42x on average for writes w/ memory channel NAND flash)