



Database Persistence, Without The Performance Penalty

Benchmarking McObject's In-Memory Database System
With AgigA Tech's Non-Volatile DIMM Technology

McObject LLC
22525 SE 64th Place
Suite 302
Issaquah, WA 98027

Phone: 425-888-8505
E-mail: info@mcobject.com
www.mcobject.com

Copyright 2014, McObject LLC

In the past decade, in-memory database systems (IMDSs) have emerged as the database solution for many real-time applications. An IMDS stores data entirely in main memory (DRAM), leveraging memory's superior read/write speed while eliminating the file I/O, cache management, data transfer and other overhead inherent in traditional database management systems (DBMSs) that are designed to store records on persistent media. As a result, in-memory database systems perform an order of magnitude faster than their disk-based counterparts.

A disadvantage to DRAM-based storage is volatility. What happens to records stored in main memory if power disappears? Without some kind of backup or log, they're gone – and some applications require a higher level of data durability and recoverability. As a solution, IMDS vendors offer transaction logging, which keeps a record of changes to the database. But this logging requires writing to persistent storage. Will re-introducing these “database writes” eliminate IMDSs' performance advantage over disk-based DBMSs?

Another solution is to deploy an IMDS using DRAM that is backed up by battery power. However, disadvantages of this “battery-backed RAM” or non-volatile RAM (NVRAM) include batteries' restrictive temperature requirements and the risk of leaking corrosive and toxic fluids. Battery-backed RAM also suffers from limited length of storage (data typically must be transferred from the chip within 72 hours of failover to battery), long re-charge cycles, finite battery shelf life, and overall high cost-of-ownership.

To get around these limitations, AgigA Tech, a Cypress Semiconductor subsidiary, has introduced its AGIGARAM® Non-Volatile DIMM (NVDIMM) solution, which removes the battery from battery-backed RAM by combining DRAM with NAND flash and an ultracapacitor power source. When a system is operating normally, AGIGARAM NVDIMMs function as normal, high speed DRAM. But in the event of unexpected power loss, the ultracapacitor provides a burst of electricity that is used to write the contents of main memory to the NAND flash chip. This data can be held there virtually indefinitely, with AGIGARAM restoring the FLASH contents to DRAM upon system recovery.

Managing data with the speed of DRAM, but with instant, reliable persistence in the event of system failure, and none of the drawbacks of batteries – it sounds like IMDS nirvana, at least for that sub-set of real-time applications that require data durability. And happily, McObject can confirm that it works exactly as described, having tested its *eXtremeDB* IMDS with AGIGARAM in a self-induced power-off, power-on scenario. The next question likely to come from real-time system designers: is using AGIGARAM with an IMDS as *fast* as working with standard DRAM? An IMDS may gain persistence through AGIGARAM, but does it give up performance?

To answer that question, McObject and AgigA Tech collaborated on benchmark tests comparing IMDS performance using AGIGARAM for main memory storage compared to using conventional DRAM. The benchmark tests also addressed the other question raised above, namely, to what extent will an IMDS with transaction logging retain its performance advantage over a disk-based DBMS? For these latter tests involving persistent storage (of the IMDS's transaction log, and the

entire database in the case of the on-disk DBMS) the storage “device” consisted of a RAM-disk configured using the AGIGARAM NVDIMM. The reasons for using a RAM-disk instead of a conventional hard disk drive or solid state drive are described below.

Test hardware and software

The test platform consisted of the following elements:

- **Hardware platform.** Intel Oak Creek Canyon reference motherboard with Intel Pentium Dual Core CPU 1407 @ 2.8 GHz processor, 8 GB Kingston conventional DDR3-1333 DRAM and Agiga Tech’s 4 GB AGIGARAM DDR3-1600 NVDIMM. The AGIGARAM NVDIMM modules are designed to plug into printed circuit boards just like conventional DRAM DIMMs. Note that although the AGIGARAM NVDIM is rated for DDR3-1600 speeds, the benchmarks were run at DDR3-1333 due to system configuration limitations.
- **Operating system.** Debian Linux 2.6.32.5.
- **In-memory database system.** McObject *eXtremeDB* IMDS.
- **Disk-based DBMS.** McObject *eXtremeDB* Fusion. This hybrid edition of *eXtremeDB* adds optional, highly configurable persistent storage to the core IMDS, enabling the developer to combine in-memory and on-disk database storage, or specify entirely on-disk storage. In the 100% “on-disk” (with the disk being a RAM-disk) database scenario used in the tests, below, *eXtremeDB* implements caching, file I/O and other aspects of a disk-based DBMS relevant to performance.

Methodology

The test application performed five database operations: inserts, updates, deletes, index searches and table traversals. It measured performance on each operation while looping, with each loop constituting a database transaction and each loop containing at least two instances of the operation, as follows:

Insert test – 2 record inserts/loop

Update test – 2 record updates/loop

Delete test – 2 record deletes/loop

Index search test – 8 searches/loop

Table traversal test – 8 traversals/loop

Total loops executed for each test ranged from 20,000 to 400,000. The benchmark application recorded the number of loops accomplished per millisecond for each of the two database types

(on-disk DBMS and in-memory database system with transaction logging, or “IMDS+TL”) and both types of memory (AGIGARAM NVDIMM and conventional DRAM DIMM).

The *eXtremeDB* database system supports multiple application programming interfaces (APIs) including SQL/ODBC/JDBC and native interfaces for the C/C++, C# (.NET) and Java languages. Code for the operations described above was implemented using the native C/C++ API.

DBMS Recovery from Failure

The AGIGARAM NVDIMM’s response to system failure was tested using a simple technique: rebooting the test system mid-execution. When the system came back up, the test application re-started automatically, accessed the database in its pre-failure state (upon recovery, the AGIGARAM NVDIMM had loaded it from its flash into its DRAM), checked for database consistency and resumed operation.

Test application code enabling database recovery leveraged an *eXtremeDB* capability McObject originally added to enable customers in the aerospace industry to use the IMDS with battery-backed RAM as storage. This feature enables a process to re-connect to an NVRAM-hosted *eXtremeDB* database, initiate any needed cleanup, and resume normal operation after a system reboots. An application’s recovery algorithm assumes that the memory block of the database memory device assigned as MCO_MEMORY_ASSIGN_DATABASE can be re-used after an application crash or power failure by re-opening it with the additional flag MCO_DB_OPEN_EXISTING. In this way, after re-boot, *eXtremeDB* was able to access the database from the same AGIGARAM NVDIMM memory space that was used prior to the system restart.

Test 1 – On-Disk DBMS vs. In-Memory Database System + Transaction Logging

In-memory database system vendors offer transaction logging to mitigate the volatility of “pure” in-memory data storage. Changes to the database are recorded in a log, which can be used to restore a database to its pre-failure state. However, transaction logging requires persistent storage (for the log) which could impact IMDS performance. For this reason, IMDS vendors are sometimes asked whether their products, when deployed with transaction logging, still outperform on-disk DBMSs.

The first benchmark test sought to answer this question. In a twist, the “hard disk” used for persistent storage was actually a RAM disk (that is, a memory-based analog of disk storage) configured using the AGIGARAM NVDIMM as memory. This approach was chosen partly to further test the AGIGARAM product (i.e. to confirm if it would work to create a RAM-disk and have a database system interact with it), and partly to shed light on the reason *why* an IMDS

with transaction logging should significantly outperform an on-disk DBMS. As any in-memory database vendor will explain, IMDSs differ from on-disk DBMSs in important ways beyond the storage devices they're usually paired with (hard disk/solid state drive for on-disk DBMSs vs. DRAM for IMDSs). An IMDS that is designed from scratch as an IMDS eliminates caching, file I/O and other sources of overhead inherent in traditional DBMS architecture. Eliminating the hard disk – replacing it with a RAM disk – eliminates overhead stemming from physical operation of the storage device, in order to shine a spotlight on any latency effect of the IMDS's streamlined design vs. the on-disk DBMS's higher complexity.

Results

Inserts, Updates & Deletes

The test showed that for insert, update and delete operations, the IMDS with transaction logging maintained a significant speed advantage over the traditional on-disk database system (again, with both of these using a RAM-disk for their "persistent" storage). Figure 1 shows results in loops/ms for each of the configurations, as well as the performance multiple exhibited by the IMDS+TL over the on-disk DBMS. For example, in the test of database deletes, the IMDS+TL was 12.77 times faster than the on-disk DBMS. For database inserts, the IMDS+TL was 11.79 times faster than the on-disk DBMS – again, using a RAM-disk on AgigA Tech's AGIGARAM NVDIMM as the "persistent" storage device (to hold the entire database, in the case of the on-disk DBMS; and to hold the transaction log, in the case of IMDS+TL).

| <u>Insert</u> | <u>Loops/ms</u> | <u>Perf. Multiple</u> |
|--------------------------------|-----------------|-----------------------|
| On-Disk DBMS - NVDIMM RAM-disk | 4.08 | 1.00 |
| IMDS+TL - NVDIMM RAM-disk | 48.13 | 11.79 |
| <u>Update</u> | <u>Loops/ms</u> | <u>Perf. Multiple</u> |
| On-Disk DBMS - NVDIMM RAM-disk | 14.843 | 1.00 |
| IMDS+TL - NVDIMM RAM-disk | 83.143 | 5.60 |
| <u>Delete</u> | <u>Loops/ms</u> | <u>Perf. Multiple</u> |
| On-Disk DBMS - NVDIMM RAM-disk | 4.082 | 1.00 |
| IMDS+TL - NVDIMM RAM-disk | 52.151 | 12.77 |

Figure 1.

Keep in mind that all these operations are occurring in main memory (the RAM-disk), which is very fast. Still, the in-memory database system with transaction logging performs many times faster than the disk-based DBMS for inserts, updates and deletes. Why is that?

First, on-disk DBMSs cache large amounts of data in memory to avoid disk writes. The algorithms required to manage this cache are a drain on speed. An IMDS (with or without transaction logging) eliminates the caching sub-system and its attendant overhead.

Second, widely-used b-tree indexes can greatly improve random lookups, and retrieval of database content in sorted order, but they are also expensive for an on-disk DBMS to maintain during insert/update/delete operations, and become more burdensome as database size increases. (B-tree lookups are less costly with an IMDS because they impose no cache processing, accesses happen at in-memory speed, and trees are shallower because they contain no duplicated index data.)

Index Searches & Table Traversals

Database index searches and table traversals showed little to no performance change when moving from on-disk DBMS to IMDS+TL. This result was expected, because such database “reads” are typically much less costly, in performance terms, than insert, update and delete operations. Because they do not change database contents, reads impose little overhead, and hence have less to gain from moving from disk to main memory for data storage, when the disk is a RAM drive, and/or when the database is fully cached.

Test 2 – In-Memory Database System Performance with Conventional DRAM vs. AgigA Tech AGIGARAM

The next series of tests focused on the *eXtremeDB* database system’s performance as a “pure” in-memory database system storing records in main memory (i.e. with transaction logging disabled, no RAM-disk and no explicitly persistent writes – i.e. persistence was a “free” by-product of using AGIGARAM NVDIMM). These tests contrasted IMDS speed using AgigA Tech’s AGIGARAM NVDIMM as main memory against the same operations using conventional DRAM. Results are shown in Figure 2 and Figure 3, below, with results of the earlier test involving RAM-disks included for comparison. Again, the performance multiple refers to the speed advantage lent by a particular configuration (e.g. “pure” IMDS on NVDIMM) compared to the on-disk DBMS using the NVDIMM RAM-disk for storage.

Inserts

On-Disk DBMS - NVDIMM RAM-disk
IMDS+TL - NVDIMM RAM-disk
"Pure" IMDS - NVDIMM
"Pure" IMDS - Conv. Memory

Loops/ms

4.08
48.13
108.99
109.29

Perf. Multiple

1.00
11.79
26.69
26.76

Updates

On-Disk DBMS - NVDIMM RAM-disk
IMDS+TL - NVDIMM RAM-disk
"Pure" IMDS - NVDIMM
"Pure" IMDS - Conv. Memory

Loops/ms

14.843
83.143
415.369
424.628

Perf. Multiple

1.00
5.60
27.98
28.61

Deletes

On-Disk DBMS - NVDIMM RAM-disk
IMDS+TL - NVDIMM RAM-disk
"Pure" IMDS - NVDIMM
"Pure" IMDS - Conv. Memory

Loops/ms

4.082
52.151
115.274
115.274

Perf. Multiple

1.00
12.77
28.24
28.24

Figure 2.

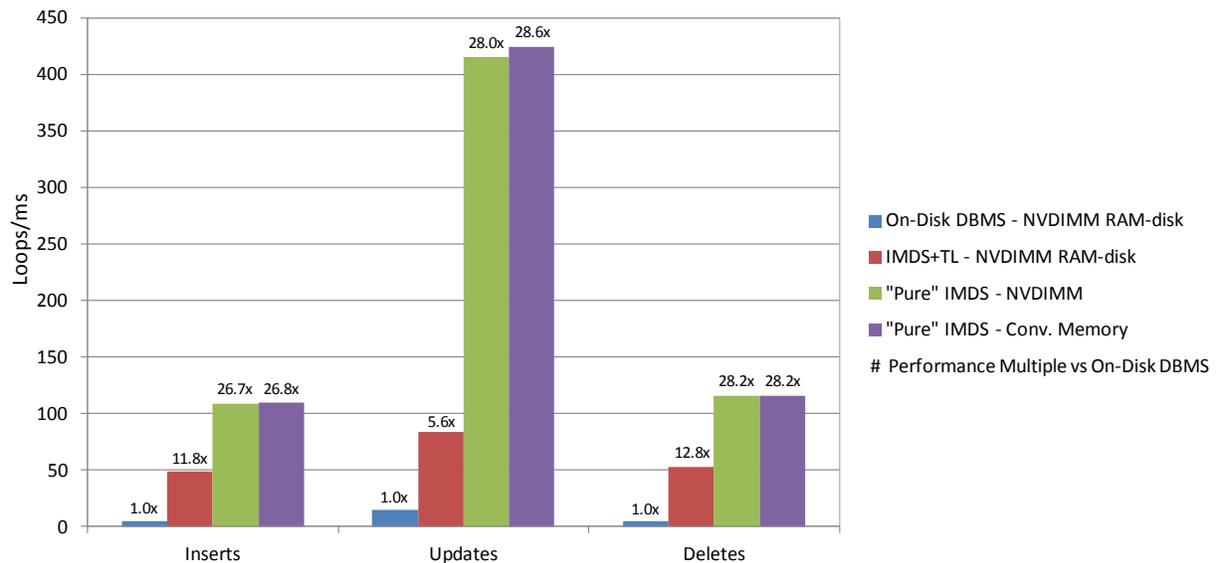


Figure 3.

Inserts, Updates & Deletes

The test showed comparable performance between AgigA Tech's AGIGARAM NVDIMM and conventional DRAM when both are used as an IMDS's main memory storage. The greatest difference occurred in the test of database updates, but even there, conventional memory outperformed the NVDIMM by a negligible 2.2% (possibly the result of a difference in timing between the larger 8 GB conventional memory and the 4 GB NVDIMM). The tests of inserts and deletes were essentially tied. The results for all three operations also confirm the speed advantage of using "pure" IMDS vs. an IMDS with transaction logging or an on-disk DBMS. While the IMDS +TL combination outperformed the on-disk DBMS by a factor ranging from 5.6x to 12.77x (560% to 1277% faster), moving to the true IMDS with main memory for storage boosted this advantage into the 26x to 28x range (2600% to 2800% faster).

Index Searches & Table Traversals

Tests of IMDS index searches and table traversals showed no speed difference between using AgigA Tech's AGIGARAM and conventional DRAM – further confirming the two memory types' interchangeability in terms of performance.

Conclusions

It is hard to overstate the significance of these test results involving a new technology combination – in-memory database system (IMDS) software and battery-free, DRAM-/flash-/ultracapacitor-based NVRAM – for systems that demand maximum data management speed as well as data durability and recoverability. That challenge is now typically met using IMDS technology with transaction logging. But AgigA Tech's AGIGARAM NVDIMM opens the door to a new and much faster approach.

In the tests described above, *eXtremeDB's* inserts and updates with AGIGARAM NVDIMM for main memory storage were twice as fast as using the same IMDS with transaction logging, and approximately five times faster for database updates (and this with the transaction log stored on RAM-disk, a solution that is faster than storing the log on a hard disk or SSD). The performance was as good as using conventional DRAM, with the key distinction that *AGIGARAM provided the IMDS with the same level of data durability as using transaction logging*. The possibility of gaining so much speed while giving up nothing in terms of data durability or recoverability makes the IMDS with AGIGARAM NVDIMM combination impossible to ignore in many application categories, including capital markets, telecom/networking, aerospace, industrial systems and more.

Certainly, factors other than performance matter to real-time system designers. Potential users will also consider price as well as hardware compatibility (AgigA Tech's product line requires newer generation Intel hardware), among other considerations. In contemplating an IMDS + AGIGARAM NVDIMM solution for data management, potential users should also investigate the

features provided by their preferred database system to support the kind of failover and recovery displayed in these benchmark tests. As mentioned above, McObject's *eXtremeDB* IMDS includes special features – added early in the product's development to support its interaction with battery-backed RAM – that enabled database recovery to work seamlessly with Agiga Tech's AGIGARAM. Using an IMDS without such features may entail more complexity, with significant development and testing required before reaching a workable solution.