

Using Non-Volatile Memory

Bill Bridge
Software Architect
Oracle

What is Non-Volatile Memory

- DIMMs on memory bus that behave like DRAM except contents are preserved through system reset or power down
- Intel announced terabytes of NVM per processor socket available in 2017
- Load/store instructions access NVM at byte granularity without calling OS
- Stores go into processor cache becoming visible before being persistent
- Dirty cache lines must be flushed from processor and memory controller
 - Software must track dirty cache lines and issue flush instructions
 - Cache lines may be flushed by hardware at anytime without a flush instruction
 - Flushed cache lines may become persistent in any order
 - Commit barrier waits for memory controller to make all flushed data persistent

Facilities for using NVM in an application

- File system built on NVM is used to mmap NVM directly into an application
- NVM transactions are needed to maintain consistent data structures
 - Multiple stores to NVM committed atomically to DIMM even if there is a crash
 - NVM locks provide isolation for multi-threaded applications
- NVM heap provides malloc/free functionality for NVM in a mapped file
- Language extensions reduce bugs and simplify code
 - The compiler needs syntax extensions to distinguish NVM vs. DRAM stores
 - Compiler inserts appropriate flushes. A missing flush is a very difficult bug to find
 - Transaction syntax ensures every transaction commits or aborts
 - Transactional store operators automatically generate undo.
 - Compiler adds checking code to detect/prevent NVM corruption sooner

For More Information

- SNIA NVM Programming Model
 - <http://www.snia.org/forums/ssi/nvmp>
- Intel Architecture Instruction Set Extensions Programming Reference
 - <https://software.intel.com/en-us/intel-isa-extensions>
- Open source NVM library and C Language extensions from Oracle
 - <https://github.com/oracle/NVM-Direct>
 - Precompiler based on clang will be open sourced when complete
- Open Source NVM Library work from Intel
 - <http://pmem.io>
- Linux kernel support & instructions
 - <https://github.com/01org/prd>