LIBMSR Documentation

Welcome to the LIBMSR Documentation.

LIBMSR has undergone a significant overhaul this summer. We attempted to minimize changes to the user interface, but some were inevitable. Hopefully this document will clear up any confusion about how to use the new LIBMSR.

We highly recommend that you use the latest version of LIBMSR. We have fixed many bugs that give corrupted or unreliable data.

If you find any bugs please report them to Scott Walker and Marty McFadden (see contacts info below).

Contacts

Project Lead

Barry Rountree, rountree4@llnl.gov

MSR-SAFE Lead Developer

Marty McFadden, mcfadden8@llnl.gov

LIBMSR Lead Developer

Scott Walker, walker91@llnl.gov

Developer

Kathleen Shoga, shoga1@llnl.gov

Build System Developer

Lauren Morita, morita4@llnl.gov

New Features

- Batch optimizations greatly reduce overhead for large numbers of reads/writes
- Can now set DRAM limits (see bug below)
- Added PMC performance counters
- No longer have to change #defines in msr_core.h to compile

Known Bugs

- Watts reading and watts limit for DRAM do not correspond as expected

Fixed Bugs

- Some core thread MSRs will load the wrong core's data if hyper threading is turned on
- Restoring MSRs was broken after batch update (deprecated)
- There are some asserts that need to be removed
- Setting a single PKG RAPL limit (there are 2 limits in 1 MSR) no longer overwrites the existing limit
- Limits and time windows can no longer overflow
- Fixed the time window conversion for most domains
- Architecture and core topology now detected at runtime
- Fixed rounding errors in RAPL
- Fixed the order of the file descriptors opened when hyper threading is enabled
- All memory is now allocated dynamically, and only when needed
- Reduced wasteful computation and memory in most files

Search this documentation

Featured Pages

Recently Updated Pages