Intel(R) MPI Library 2017 Update 2 for Linux* OS
Release Notes

Contents

Overview

The Intel(R) MPI Library for Linux* OS is a multi-fabric message passing library based on ANL* MPICH3* and OSU* MVAPICH2*.

The Intel(R) MPI Library for Linux* OS implements the Message Passing Interface, version 3.1 (MPI-3) specification. The library is thread-safe and provides the MPI standard compliant multi-threading support.

To receive technical support and updates, you need to register your Intel(R) Software Development Product. See the Technical Support section.

Product Contents

The Intel(R) MPI Library Runtime Environment (RTO) contains the tools you need to run programs including scalable process management system (Hydra) and supporting utilities, shared (.so) libraries, and documentation.

The Intel(R) MPI Library Development Kit (SDK) includes all of the Runtime Environment components and compilation tools: compiler commands (mpicc, mpiicc, etc.), include files and modules, static (.a) libraries, debug libraries, and test codes.

Related Products and Services

Information on Intel(R) Software Development Products is available at http://www.intel.com/software/products.

What's New

See http://intel.ly/1OzXhpI for a current list of deprecated features.

Intel(R) MPI Library 2017 Update 2:

- Added environment variables I_MPI_HARD_FINALIZE and I_MPI_MEMORY_SWAP_LOCK. See the Developer Reference for details.

Intel(R) MPI Library 2017 Update 1:

- PMI-2 support for SLURM*, improved SLURM support by default.
- Improved mini help and diagnostic messages, man1 pages for mpiexec.hydra, hydra_persist, and hydra_nameserver.
- Deprecations:
  - Intel(R) Xeon Phi(TM) coprocessor (codenamed Knights Corner) support.
  - Cross-OS launches support.
  - DAPL, TMI, and OFA fabrics support.

Intel(R) MPI Library 2017:

- Support for the MPI-3.1 standard.
- New topology-aware collective communication algorithms (I_MPI_ADJUST family).
- Effective MCDRAM (NUMA memory) support. See the "Memory Placement Policy Control" topic in the Developer Reference for more information.
- Controls for asynchronous progress thread pinning (I_MPI_ASYNC_PROGRESS).
- Direct receive functionality for the OFI* fabric (I_MPI_OFI_DRECV).
- PMI-2 protocol support (I_MPI_PMI2).
- New process startup method (I_MPI_HYDRA_PREFORK).
- Startup improvements for the SLURM* job manager (I_MPI_SLURM_EXT).
- New algorithm for MPI-IO collective read operation on the Lustre* file system (I_MPI_LUSTRE_STRIPE_AWARE).
- Debian Almquist (dash) shell support in compiler scripts and mpitune.
- Performance tuning for processors based on Intel(R) microarchitecture codenamed Broadwell and for Intel(R) Omni-Path Architecture (Intel(R) OPA).
- Performance tuning for Intel(R) Xeon Phi(TM) Processor and Coprocessor (code name Knights Landing) and Intel(R) OPA.
- OFI latency and message rate improvements.
- OFI is now the default fabric for Intel(R) OPA and Intel(R) True Scale Fabric.
- MPD process manager is removed.
- Dedicated pvfs2 ADIO driver is disabled.
- SSHM support is removed.
- Support for the Intel(R) microarchitectures older than the generation codenamed Sandy Bridge is deprecated.
- Documentation improvements.
- The MPI Tuner Tutorial is removed from the installation and is available online at https://software.intel.com/en-us/intel-software-technical-documentation

Intel(R) MPI Library 5.1 Update 3:

- Deprecation list updated. New deprecations:
  - SSHM
  - MPD
  - Epoll
  - JMI
  - PVFS2
- Fixed shared memory problem on Intel(R) Xeon Phi (codename: Knights Landing).
- New algorithms and selection mechanism for nonblocking collectives.

- New psm2 option for Intel(R) Omni-Path.
- Added I_MPI_BCAST_ADJUST_SEGMENT variable to control MPI_Bcast.
- Fixed long count support for some collective messages.
- Binding kit reworked with support for Intel(R) Many Integrated Core Architecture and support for ILP64 on third party compilers.

Intel(R) MPI Library 5.1 Update 2:
- Intel(R) MPI Library supports YARN* cluster manager. Llama* should be installed on the cluster.
- DAPL library UCM settings are automatically adjusted for MPI jobs of more than 1024 ranks, resulting in more stable job start-up.
- ILP64 support enhancements, support for MPI modules in Fortran 90.
- Added the direct receive functionality for the TMI fabric.
- Single copy intra-node communication using Linux supported cross memory attach (CMA) is now default.

Intel(R) MPI Library 5.1 Update 1:
- Change the named-user licensing scheme. See more details in the Installation Instructions section of Intel(R) MPI Library Installation Guide.
- Bug fixes

Intel(R) MPI Library 5.1:
- Supported OpenFabrics Interface* (OFI*) v1.0 API
- Supported Fortran 2008
- Updated the default value for I_MPI_FABRICS_LIST. See the Intel(R) MPI Library Developer Reference for details
- Added the Troubleshooting chapter to the Intel(R) MPI Library Developer Guide.
- Added new application specific features in the Automatic Tuner and Hydra process manager. See the MPI Tuner Tutorial for details.
- Added the MPI_Pcontrol feature for internal statistics.
- Increased the possible space for MPI_TAG.
- Changed the default product installation directories. See the README document for details.
- Bug fixes

Note:
Intel(R) MPI Benchmarks is delivered as part of Intel(R) MPI Library. For the new features of Intel MPI Benchmarks, see the What's New section in Intel(R) MPI Benchmarks README.

Intel(R) MPI Library 5.0 Update 3:
- Support for IBM* General Parallel File System* (GPFS*) through the new environment variable I_MPI_EXTRA_FILESYSTEM
- Support for the new bootstrap server, pbsdsh
- Support for the rename mechanism for the file, stats.txt, to avoid overwriting
- Native statistics collection can be controlled with MPI_Pcontrol
- Bug fixes

Intel(R) MPI Library 5.0 Update 2:
- Single copy intra-node communication using Linux supported cross memory attach (CMA)
- Multi-threaded optimized library is used by default for linking with single and multi-threaded MPI applications

- Added gtool option for running external tools
- Tuning for TMI fabrics on processors using the Haswell microarchitecure
- Enhancements to statistics gathering mode
- Bug fixes

Intel(R) MPI Library 5.0 Update 1:

- Changed settings for non-default installation path. If you choose to install Intel(R) MPI Library to a non-default path, impi/<version>.<package> will be appended to the selected installation path. Use symbolic links to this path if you need a specific pathname.
- Directory structure update. New links have been added to always point to the most recently installed version of the Intel(R) MPI Library.
- Bug fixes, including:
    o Avoid conflict with TMI ADI messages, resolving hangs for LS-Dyna*
- Collective performance improvements
- Documentation update
- Man pages copyright updated
- Added support for -fopenmp in mpiicc, mpiicpc, and mpiifort
- Improved pinning under job schedulers

Intel(R) MPI Library 5.0:

- Improved application binary interface (ABI) compatibility between different MPI implementations
- Improved performance of the OFA fabric on Intel(R) Xeon(R) Processor E5 V2 and E7 V2
- Add the option I_MPI_JOB_RESPECT_PROCESS_PLACEMENT to honour process placement from job schedulers
- Remove all IA-32 architecture support
- Add the debug information into optimized single/multi-threaded libraries
- Implement the MPI-3 standard including but not limited to:
    o Non-blocking collective operations
    o Fast one-sided operations
    o Large counts for messages greater than 2GB
- Support for Parallel Distributed Shell*
- Allow permuted entries in machine file when running a single instance of pmi-proxy
- Support for mixed operating systems in the Hydra process manager (Linux* OS and Windows* OS)
- Make the following changes to the documentation:
    o Change the Intel(R) MPI Library Getting Started Guide to Intel(R) MPI Library Developer Guide
    o Add the Intel(R) MPI Library Getting Started page
    o Add the tutorial: MPI Tuner for Intel(R) MPI Library
- Bug fixes
- Deprecate MPD process managers

Key Features

This release of the Intel(R) MPI Library supports the following major features:

- MPI-1, MPI-2.2 and MPI-3.1 specification conformance
- Support for Intel(R) Xeon Phi(TM) processors (formerly code named Knights Landing).
- Support for any combination of the following interconnection fabrics:

  - o Shared memory
  - o Network fabrics supporting Intel(R) Omni-Path Architecture (Intel(R) OPA) devices, through either Tag Matching Interface (TMI) or OpenFabrics Interface* (OFI*).
  - o Network fabrics with tag matching capabilities through Tag Matching Interface (TMI), such as Intel(R) True Scale Fabric, Infiniband*, Myrinet* and other interconnects
  - o Native InfiniBand* interface through OFED* verbs provided by Open Fabrics Alliance* (OFA*)
  - o OpenFabrics Interface* (OFI*)
  - o RDMA-capable network fabrics through DAPL*, such as InfiniBand* and Myrinet*
  - o Sockets, for example, TCP/IP over Ethernet*, Gigabit Ethernet*, and other interconnects
- Support for the following MPI communication modes related to Intel(R) Xeon Phi(TM) coprocessor:
  - o Communication inside the Intel(R) Xeon Phi(TM) coprocessor
  - o Communication between the Intel(R) Xeon Phi(TM) coprocessor and the host CPU inside one node
  - o Communication between the Intel(R) Xeon Phi(TM) coprocessors inside one node
  - o Communication between the Intel(R) Xeon Phi(TM) coprocessors and host CPU between several nodes
- (SDK only) Support for Intel(R) 64 architecture and Intel(R) MIC Architecture clusters using:
  - o Intel(R) C++ Compiler for Linux* OS version 14.0 through 16.0
  - o Intel(R) Fortran Compiler for Linux* OS version 14.0 through 16.0
  - o GNU* C, C++ and Fortran 95 compilers
- (SDK only) C, C++, Fortran 77, Fortran 90, and Fortran 2008 language bindings
- (SDK only) Dynamic or static linking

System Requirements

The following sections describe supported hardware and software.

Supported Hardware

Systems based on the Intel(R) 64 architecture, in particular:

Intel(R) Core(TM) processor family or higher
Intel(R) Xeon(R) E5 v3 processor families recommended
Intel(R) Xeon(R) E7 v2 processor families recommended
1 GB of RAM per core
2 GB of RAM per core recommended
1 GB of free hard disk space

Intel(R) Xeon Phi(TM) Coprocessor (formerly code named Knights Corner) based on Intel(R) MIC Architecture

Supported Software

Operating Systems: (issues including installation ones are possible for operating systems that are not released at the date of the current Intel MPI Library release)

Systems based on the Intel(R) 64 architecture:
Red Hat* Enterprise Linux* 6, 7
Fedora* 23, 24
CentOS* 6, 7
SUSE* Linux Enterprise Server* 11, 12,
Ubuntu* LTS 14.04, 16.04
Debian* 7, 8

(SDK only) Compilers:

GNU*: C, C++, Fortran 77 3.3 or higher, Fortran 95 4.4.0 or higher

Intel(R) C++ Compiler for Linux* OS 15.0 through 17.0 or higher
Intel(R) Fortran Compiler for Linux* OS 15.0 through 17.0 or higher

(SDK only) Supported Debuggers:

Rogue Wave* Software TotalView* 6.8 or higher
Allinea* DDT* v1.9.2 or higher
GNU* Debuggers

Batch Systems:

Platform* LSF* 6.1 or higher
Altair* PBS Pro* 7.1 or higher
Torque* 1.2.0 or higher
Parallelnavi* NQS* for Linux* OS V2.0L10 or higher
Parallelnavi* for Linux* OS Advanced Edition V1.0L10A or higher
NetBatch* v6.x or higher
SLURM* 1.2.21 or higher
Univa* Grid Engine* 6.1 or higher
IBM* LoadLeveler* 4.1.1.5 or higher
Platform* Lava* 1.0

Recommended InfiniBand Software:

- OpenFabrics* Enterprise Distribution (OFED*) 1.5.4.1 or higher.
- Intel(R) True Scale Fabric Host Channel Adapter Host Drivers & Software (OFED+) v7.2.0 or higher.
- Mellanox* OFED* 1.5.3-4.0.42 or higher.

Additional Software:

- Python* 2.2 or higher, including the python-xml module. Python* distributions are available for download from your OS vendor or at http://www.python.org (for Python* source distributions).
- An XML parser such as expat* or pyxml*.
- If using InfiniBand*, Myrinet*, or other RDMA-capable network fabrics, a DAPL* 1.2 standard-compliant provider library/driver is required. DAPL* providers are typically provided with your network fabric hardware and software.

- DAPL auto provider selection mechanism requires dapl-2.0.37 or higher for Intel(R) Xeon(R) and Intel(R) Manycore Platform Software Stack (Intel(R) MPSS) gold update 3 or higher for Intel (R) Xeon Phi(TM).
- Improved NUMA support requires dapl-2.0.37 or higher
- Intel(R) Many Integrated Core Platform Software Stack (MPSS) 2.1 Beta or higher for the respective host operating system. MPSS may support specific operating system/kernel only. For most recent specific operating system requirements, see http://software.intel.com/en-us/articles/intel-mpi-library-for-linux-kb/all/ or http://software.intel.com/mic-developer .
  Note that the Intel(R) MPI Library is tuned for optimum performance on the latest MPSS versions as of release time.
- If you use the memory policy control function to attach processes memory to specified NUMA* nodes memory, install the libnuma.so* library and numactl* utility 2.0.9 or higher. The following packages are required:
  - numactl
  - numactl-devel
  - numactl-libs

(SDK only) Supported Languages

For GNU* compilers: C, C++, Fortran 77, Fortran 95
For Intel compilers: C, C++, Fortran 77, Fortran 90, Fortran 95, Fortran 2008

Installation Notes

See the Intel(R) MPI Library for Linux* OS Installation Guide for details.

Documentation

Intel(R) MPI Library for Linux* OS Getting Started page, found in get_started.htm, contains information on the following subject:

- Compiling and running your MPI program

Intel(R) MPI Library for Linux* OS Developer Guide, found in Developer_Guide.htm (HTML Uncompressed Help) and Developer_Guide.pdf, contains information on the following subjects:

- First steps using the Intel(R) MPI Library

- First-aid troubleshooting actions

Intel(R) MPI Library for Linux* OS Developer Reference, found in Developer_Reference.htm (HTML Uncompressed Help) and Developer_Reference.pdf, contains information on the following subjects:

- Command Reference describes commands, options, and environment variables

- Tuning Reference describes environment variables that influence library behavior and performance

Intel(R) MPI Library for Linux* OS Installation Guide, found in INSTALL.html, contains information on the following subjects:

- Obtaining, installing, and uninstalling the Intel(R) MPI Library
- Getting technical support

Tutorial: MPI Tuner for Intel(R) MPI Library, found in https://software.intel.com/en-us/intel-software-technical-documentation, contains information on the following subjects:

- How to use the MPI Tuner for Intel(R) MPI Library to get optimized configuration files for the runtime library automatically
- How to troubleshoot the common issues with the MPI tuner

Known Issues and Limitations

Note: The following list includes the information until Intel(R) MPI Library 2017 Update 1 is released. For the most up-to-date list of known issues, as well as latest tips and tricks on using the library, visit the Intel(R) MPI Library for Linux* Knowledge Base at http://software.intel.com/en-us/articles/intel-mpi-library-for-linux-kb/all/

- ILP64 is not supported by MPI modules for Fortran* 2008.

- The I_MPI_HYDRA_PREFORK variable is not supported on Intel(R) Xeon Phi(TM) coprocessors codenamed Knights Corner.

- In case of program termination(like signal), remove the trash in the /dev/shm/ directory manually with: rm -r /dev/shm/shm-col-space-*

- In case of large number (per node) of simultaneously used communicators (more than 10.000), it is recommended to increase maximum numbers of memory mappings with one of the following methods:
  o echo 1048576 > /proc/sys/vm/max_map_count
  o sysctl -w vm.max_map_count=1048576
  o switch off shared memory collectives with next variable:
    I_MPI_COLL_INTRANODE=pt2pt.

- Intel(R) MPI Library does not support the new processes startup method (I_MPI_HYDRA_PREFORK=1) on Intel(R) Xeon Phi(TM) processors (formerly code named Knights Corner).

- On some Linux* distributions, the Intel(R) MPI Library will fail for non-root users due to security limitations.
  This has been seen on Ubuntu* 12.04, and could impact other distributions and versions as well. Two workarounds have been identified for this issue.
  o Enable ptrace for non-root users with:
    echo 0 | sudo tee /proc/sys/kernel/yama/ptrace_scope

  o Or, revert the Intel(R) MPI Library to an earlier shared memory
    mechanism which is not impacted by setting:
    I_MPI_SHM_LMT=shm

- Ubuntu* does not allow attaching a debugger to a non-child process. In order to use -gdb, this must be disabled by setting the sysctl value /proc/sys/kernel/yama/ptrace_scope to 0.

- Cross-OS runs using ssh from a Windows* host fail. Two known workarounds exist:
  - o Create a symlink on the Linux* host that looks identical to the Windows* path to pmi_proxy.
  - o Start hydra_persist on the Linux* host in the background (hydra_persist &) and use -bootstrap service from the Windows* host. This requires that the Hydra service also be installed and started on the Windows* host.

- The OFA fabric and certain DAPL providers may not work or provide worthwhile performance with the Intel(R) MPI Library for Linux* OS, for example:
  - o Intel(R) True Scale Fabric and Intel(R) Omni-Path Fabric. Use the OFI/TMI libraries included with the Intel(R) MPI Library when running over the Intel(R) True Scale Fabric PSM* interconnect interface for best performance.

- Switching on statistics gathering could result in increased time in MPI_Finalize.

- In systems where some nodes have only Intel(R) True Scale Fabric or Intel(R) Omni-Path Fabric available, while others have both Intel(R) True Scale and e.g. Mellanox* HCAs, automatic fabric detection will lead to a hang or failure, as the first type of nodes will select ofi/tmi, and the second type will select dapl as the internode fabric. To avoid this, explicitly specify a fabric that is available on all nodes as the internode fabric.

- Intel(R) MPI Library 5.x for Linux* OS is binary compatible with the majority of Intel MPI Library 4.1.x-based applications. Recompile your application only if you use:
  - o MPI_Dist_graph_create, MPI_Dist_graph_create_adjacent, MPI_Dist_graph_neighbors, MPI_Dist_graph_neighbors_count, (C, C++, Fortran)
    MPI::Get_address (C++ only)

- Hydra hangs when used under versions 7.2 or 7.3 of the GNU* Debugger gdb.

- In order to run a mixed operating system job (Linux* and Windows*), all binaries must link to the same single or multithreaded MPI library. The single and multithreaded libraries are incompatible with each other and should not be mixed. Note that the pre-compiled binaries for the Intel® MPI Benchmarks are inconsistent (Linux* version links to multithreaded, Windows* version links to single threaded) and as such, at least one must be rebuilt to match the other.

- Intel(R) MPI Library does not support using the OFA fabric over an Intel(R) Symmetric Communications Interface (Intel(R) SCI) adapter. If you are using an Intel SCI adapter, such as with Intel(R) Many Integrated Core Architecture, you will need to select a different fabric.

- The TMI and OFI fabrics over PSM do not support messages larger than 2^32 - 1 bytes. If you have messages larger than this limit, select a different fabric.

- Intel(R) Software Manager will always install to either /opt or $HOME on Linux* OS even if a custom installation path is chosen. This can slow installation when the destination folder is a slow NFS shared folder, even if locally hosted.

- If communication between two existing MPI applications is established using the process attachment mechanism, the library does not control whether the same fabric has been selected for each application. This situation may cause unexpected applications behavior. Set the I_MPI_FABRICS variable to the same values for each application to avoid this issue.

- Do not load thread-safe Intel(R) MPI libraries through dlopen(3).

- Certain DAPL* providers may not function properly if your application uses system(3), fork(2), vfork(2), or clone(2) system calls. Do not use these system calls or functions based upon them. For example, system(3), with:
  - OFED* DAPL* provider with Linux* kernel version earlier than official version 2.6.16. Set the RDMAV_FORK_SAFE environment variable to enable the OFED workaround with compatible kernel version.

- MPI_Mprobe, MPI_Improbe, and MPI_Cancel are not supported by the TMI and OFI fabrics.

- You may get an error message at the end of a Checkpoint-Restart enabled application, if some of the application processes exit in the middle of taking a checkpoint image. Such error does not impact the application, thus you can ignore it. To avoid this error, set a larger number than your previous configured value for the <sec> argument of the -checkpoint-interval option. The following error message details are for your reference:

  ```
  [proxy:0:0@hostname] HYDT_ckpoint_blcr_checkpoint (./tools/ckpoint/blcr/
  ckpoint_blcr.c:313): cr_poll_checkpoint failed: No such process
  [proxy:0:0@hostname] ckpoint_thread (./tools/ckpoint/ckpoint.c:559):
  blcr checkpoint returned error
  [proxy:0:0@hostname] HYDT_ckpoint_finalize (./tools/ckpoint/ckpoint.c:878)
  : Error in checkpoint thread 0x7
  ```

- The Intel(R) MPI Library requires the presence of the /dev/shm device in the system. To avoid failures related to the inability to create a shared memory segment, make sure the /dev/shm device is set up correctly.

- The Intel(R) MPI Library uses TCP sockets to pass stdin stream to the application. If you redirect a large file, the transfer can take long and cause the communication to hang on the remote side. To avoid this issue, pass large files to the application as command line options.

- DAPL auto provider selection mechanism and improved NUMA support require dapl-2.0.37 or newer.

- If you set I_MPI_SHM_LMT=direct, the setting has no effect if the Linux* kernel version is lower than 3.2.

- When using the Linux boot parameter isolcpus with an Intel(R) Xeon Phi(TM) processor using default MPI settings, MPI based applications may fail. If possible, change or remove the isolcpus Linux boot parameter. If this is not possible, you can try setting I_MPI_PIN to off.

- In some cases, collective calls over the OFA fabric may provide incorrect results. Try setting I_MPI_ADJUST_ALLGATHER to a value between 1 and 4 to resolve the issue.

Technical Support

Your feedback is very important to us. To receive technical support for the tools provided in this product and technical information including FAQ's and product updates, you need to register for an Intel(R) Premier Support account at the Registration Center.

This package is supported by Intel(R) Premier Support. Direct customer support requests at:
    https://premier.intel.com

General information on Intel(R) Software Development Products support offerings may be obtained at:
    http://www.intel.com/software/products/support

The Intel(R) MPI Library home page can be found at:
    http://www.intel.com/go/mpi

The Intel(R) MPI Library support web site,
http://software.intel.com/en-us/articles/intel-mpi-library-for-linux-kb/all/
provides the latest top technical issues, frequently asked questions, product documentation, and product errata.

Requests for licenses can be directed to the Registration Center at:
    http://www.intel.com/software/products/registrationcenter

Before submitting a support issue, see the Intel(R) MPI Library for Linux* OS Developer Guide for details on post-install testing to ensure that basic facilities are working.

When submitting a support issue to Intel(R) Premier Support, please provide specific details of your problem, including:

- The Intel(R) MPI Library package name and version information

  - Host architecture (for example, Intel(R) 64 architecture)
  - Compiler(s) and versions
  - Operating system(s) and versions
  - Specifics on how to reproduce the problem. Include makefiles,
    command lines, small test cases, and build instructions.
    Use <installdir>/test sources as test cases, when possible.

You can obtain version information for the Intel(R) MPI Library package in the file mpisupport.txt.

Submitting Issues

- Go to https://premier.intel.com
- Log in to the site. Note that your username and password are case-sensitive.
- Click on the "Submit Issue" button in the top right.
- Search for and select "Intel(R) MPI Library, Linux*" from the product list,
  then click "Next".
- Complete the fields describing your issue and attach any supporting files.
- Review the issue and complete submission.

Note: Notify your support representative prior to submitting source code where access needs to be restricted to certain countries to determine if this request can be accommodated.

Disclaimer and Legal Information

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Intel technologies features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at Intel.com, or from the OEM or retailer.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, Xeon, and Xeon Phi are trademarks of Intel Corporation

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804