Multicore Software Development Kit for High-Performance Computing

Release Notes

Applies to Release: 03.00.01
Publication Date:
May 27, 2015

TEXAS INSTRUMENTS

Document License

Contents

Release Notes

MCSDK-HPC Release 3.0.1

1 Overview

This document is the Release Notes for Release 3.0.x of the Multicore Software Development Kit for High-Performance Computing (MCSDK-HPC). MCSDK-HPC is built as an add-on on top of the foundational Multicore Software Development Kit (MCSDK). MCSDK-HPC, along with MCSDK, provides the complete environment to develop HPC applications on a cluster of TI Keystone II devices.

2 Licensing

Please refer to the software manifest which outlines the licensing status for all packages included in this release.

3 Documentation

- **README**: Provides information on setting up the environment to compile the provided components and build out-of-box demos.
- **MCSDK-HPC Software Manifest**: Provides license information on software included in the MCSDK-HPC release. This document can be found in the release at [INSTALL-DIR]/mcsdk-hpc_<ver>/docs.

4 Platform and Device Support

The device and platforms supported with this release include:

Platform	Supported Devices	Supported EVM
[K2H]	66AK2H12	XTCIEVMK2X

5 Technical Support and Product Updates

For technical discussions and issues, please visit:

- KeyStone Multicore forum
- BIOS Embedded Software forum
- Linux Embedded Software forum
- Code Composer Studio forum

- **TI C/C++ Compiler forum**
- **Embedded Processors wiki**

For local support in China, please visit

- **China Support forum**

6 Releases

6.1 Components/Versions packaged in this release

Component	Version
framework_components	03.31.00.02
openmp_dsp	02.01.16.03
ti-openmpi	01.00.00.22
xdais	07.24.00.04
fftlib	03.00.00.02
ti-opencl	1.1.1.1
blas	03.11.00.03
ti-openmpacc	01.02.00
gdbc6x	1.0.0
gdbserver-c6x	1.0.0
cmem, cmem-mod-dkms	4.10.00.01
debugss-mod-dkms	1.0.1
dsptop	1.1.5
gdb-proxy-mod-dkms	1.0.1
mpm	2.0.1.6
mpm-transport	1.0.5.2
temperature-mod-dkms	1.0.1
ti-llvm-3.3	3.3
xdc-headers	3.25.05.94
ti-cgt-c6x	8.0.3
uio-module-drv-dkms	01.00.00.00
edma3-lld	02.11.11.15
ti-fftw	3.0.0.3
ti-linalg	1.0.0.1
ti-ml	01.00.02.00
ti-gl	1.00.00

6.2 Release Features

6.2.1 DEV.MCSDK_HPC.03.00.01.12

Changes from previous release:

- **OpenMPI**
  New optional MCA parameters added:
  - HLINK: "--mca btl_hlink_rx_thread <0|1>" disables RX side thread (value of 0), but in that case only non-blocking messages up to 2MB can be served. Default setting is '1' allowing larger non-blocking messages to be used.
  - SRIO: "--mca btl_srio_pdsp_credit_period <1|2|3|4>" modifies credit message interval, exchanged between neighboring nodes. Default setting is '1' . Bigger credit size may allow higher BW for multi-hop message transfers.
  - Handling of 3 nodes in HLINK.
    - It is possible now to use HLINK interface for clusters with 3 nodes as well, but only for direct connections. It is necessary to specify TCP and/or SRIO as alternative interface that will be used for communication between non-adjacent nodes.
  - Use EDMA for diagonal HLINK links
    - This feature is enabled ONLY with "--mca btl_hlink_rx_thread 1" setting.
    - Communication over HLINK between non-adjacent nodes is carried over staging area in common adjacent node. Data are copied from node A to node B. Node C does polling of memory region in node B to detect incoming message.
    - If pending message is detected, node C has used CPU memcpy() which has low BW. Now, EDMA is used to do read from node B to node C memory (increasing BW 20x).

- **OpenCL**
  - Give larger alignment to buffers allocated in extended memory to help MPAX allocation.
  - Provide better error reporting when error occurs in OpenCL runtime.

- **FFTW**
  - DSP acceleration was added for the following
    - fftw_plan_dft_2d
    - fftw_plan_dft_3d
    - fftw_plan_dft_r2c_1d
    - fftw_plan_dft_r2c_2d
    - fftw_plan_dft_r2c_3d
    - fftw_plan_dft_c2r_1d
    - fftw_plan_dft_c2r_2d
    - fftw_plan_dft_c2r_3d
    - fftwf_plan_dft_1d
    - fftwf_plan_dft_2d
    - fftwf_plan_dft_3d
    - fftwf_plan_dft_r2c_1d
    - fftwf_plan_dft_r2c_2d
    - fftwf_plan_dft_r2c_3d
    - fftwf_plan_dft_c2r_1d
    - fftwf_plan_dft_c2r_2d
    - fftwf_plan_dft_c2r_3d
    - fftw_plan_many_dft_c2r
    - fftw_plan_many_dft_r2c

  - fftwf_plan_many_dft
  - fftwf_plan_many_dft_c2r
  - fftwf_plan_many_dft_r2c

- Added cleanup of opencl artifacts in /var/tmp/ after execution of fftw.

- **Machine Libraries (ML)**
  - Enable CNN states write/read
  - Enable minimum memory mode for CNN testing
  - Enable batch mode for CNN testing/training
  - Add new layer types, accuracy layer, softmaxCost layer
  - Remove softmax layer from nonlinear layer to be a new layer type
  - Fixes for cross compilation of libraries on x86

- **Code Gen Tools (ti-cgt-c6x)**
  - Code Gen Tools have been updated to version 8.0.3

- **Linear Algebra (Linalg)**

  - Fixes for cross compilation of examples from x86

- **HPC**
  - Fixes for IPK installation of FFTW and libraries (TI-ML and TI-GL) on EVM.
  - Fix for automated test to run on EVMs

- **Known Issues**
  - LINALG: CLAPACK Test fails.
  - TI-ML: On K2H EVM, Machine learning libraries, many examples require opencv and protobuf packages to compile and run which are not a part of the HPC installer on EVM. Hence these examples do not compile/run on K2H EVM.

6.2.2 DEV.MCSDK_HPC.03.00.01.11

Changes from previous release:

- **OpenMPI**
  - Increased MSMC footprint to 1MB to avoid the PDSP lockup issue.
  - Reduce SRIO latency by reducing min pkt size from 2048 to 264

- **OpenCL**
  - Make __malloc/free_ddr/msmc() thread safe, use Lock to guard the bookkeeping map accesses
  - Update to always flush buffers after kernel execution, even when kernel has only WRITE_ONLY buffers
  - Update MPX examples to query available memory and allocate buffers accordingly
  - Add OSAL critical section support in RM client code that allocates QMSS resources

- **Linear Algebra (Linalg)**

  - Added multiple thread support
  - Removed dependency on libatlas-base-dev package.
  - Added native compilation for DSP code

- **Machine Libraries (ML)**
  - Changed dependency from libaltas-base-dev package to ti-linalg package
  - Added pre-downloaded image as a part of examples (/examples/timl/app/cnn/class/cifar10/) so that they could be run out of the box.

- **HPC**
  - Updates to the build process facilitating the use of latest DSP LLDs by OpenCL and OpenMP while building the DSP monitor.
  - Reduce the MSMC available for applications to 4.5MB (from 5MB)

- **Known Issues**
  - OpenMPI: MPI Intel Benchmark Tests over SRIO while running concurrently with dgemm example of OpenCL can cause Intel Benchmark Tests to fail.
  - TI-FFTW: FFTW1D Batch demo (fft_1d_dp_batch) fails.

6.2.3 DEV.MCSDK_HPC.03.00.01.10

Changes from previous release:

- **New Components Added:**
  The following new components have been added to MCSDK-HPC

TI Linear Algebra (ti-linalg): This includes a CLAPACK library (http://www.netlib.org/clapack/) for high level dense linear algebra and a BLIS based BLAS library (https://code.google.com/p/blis/) for low level dense linear algebra primitives. The CLAPACkKlibrary runs on the ARM. The BLAS library includes an ARM interface, ARM compute library and DSP compute library.

TI Graph Libraries(ti-gl):
This is a template graph library that supports the Pregel API http://dl.acm.org/citation.cfm?id=1807184). Pregel is a vertex centric graph library based on a barrier synchronous protocol with nice scaling properties to multiple devices using Open MPI.

TI Machine Learning Libraries(ti-ml): The Texas Instruments Machine Learning (TIML) library is C implementation of common machine learning functions optimized for TI embedded devices. Initially, the library supports convolutional neural networks (CNNs), but will grow with time in scope. The library has a minimal set of dependencies upon other libraries to simplify the installation and use. More detailed documentation located at /usr/share/doc/timl/timl.pdf.gz on the file system after the installation.

NOTE: Please note that HPC automated testing does not test the machine learning libraries as of now. The tests have to be carried out manually as per the steps mentioned in the documentation.

FFTW(ti-fftw):FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data (as well as of even/odd data, i.e. the discrete cosine/sine transforms or DCT/DST). TI-FFTW is FFTW version 3.3.4 adapted for TI 66AK2Hxx family. Some FFTW functions are accelerated with C66x DSP for faster performance. Please visit FFTW website for documentation http://www.fftw.org/

- **OpenMPI**
  - Reduce MSMC memory footprint of MPI framework, from 1.5MB to 0.5MB
  - MPI BTL SRIO receive thread introduced to handle sequence of non-blocking messages (OSU benchmarks)

- **HPC**
  - New modules integrated – LINALG,FFTW,TI-GL,TI-ML
  - Increased MSMC area for applications to 5 MB
  - Fix for non-root user being unable to run RM based applications, including MPI over SRIO
  - Automated test modifications to
    - Custom ignore some test directories as needed
    - Skip single node MPI tests in the automated test sequence

  NOTE: Please note that the multimode fft examples (openmpi+opencl, and openmpi+opencl+openmp) which were present in drop 9, are not present in this release. These tests were removed as the FFTLIB interfaces changed to support FFTW and are not compatible anymore with the above demos.

- **PDK-Keystone**
  - Fix for PDSP firmware involving read-write-after-fence operations.

- **OpenCL**
  - Use of RM to manage QMSS resources between OpenMP and other applications.

- **OpenMP**
  - Updates to documentation.
  - Added support for long long loop counters in OpenMP work-sharing loops (#omp for).(Visible only with 8.1 CGTOOLS though)
- **FFTLIB**
  - Modifications to interface with FFTW. Please note that the older FFTLIB APIs for interfacing with DSP are not supported anymore.

- **Known Issues**

  - MPI Intel Benchmark Tests over SRIO while running concurrently with dgemm example of OpenCL can cause Intel Benchmark Tests to fail.

6.2.4 DEV.MCSDK_HPC.03.00.01.09

Changes from previous release:

- **OpenMPI**
  - Fix for the SRIO Type11 discontinuity issue.
  - Support for restarting of MPI over SRIO, after a termination by Ctrl-C.
    - [known issue] Termination initiated with Ctrl-C needs 20-30 seconds. Avoid starting another MPI test before this time.
  - Support for 4MB payload for Intel benchmark tests
  - Fix to trigger regeneration of /dev/rio_mports
- **HPC**
  - Increased MSMC area for applications to 4 MB
- **OpenCL**
  - Change use of QMSS resources to avoid conflicts with MPI.
  - Fix clocl issue where incorrect path was specified for the intermediate .bc file when clocl invoked the C6x compiler.
  - Do not copy a buffer created with USE_HOST_PTR when the pointer points to the middle of an object allocated via __malloc_ddr or __malloc_msmc
  - Fixed a C++ global objects destruction ordering problem that can cause a segmentation fault at OpenCL application exit.
  - Update version scheme for debian packaging to ww.xx.yy.zz format.

Where:

ww.xx - Maps to supported OpenCL specification version. Currently 1.1.
yy - TI release
zz - TI patch release

- **Known Issues**
  - MPI Intel Benchmark Tests over SRIO while running concurrently with dgemm example of OpenCL can cause Intel Benchmark Tests to fail.

6.2.5 DEV.MCSDK_HPC.03.00.01.08

Changes from previous release:

- **OpenMPI**
  - Added support for 180 node cluster
    - NOTE: The current version supports message size up to 128Kbytes for 180 node clusters. Higher message sizes (up to 1MB) works well with up to 72 node clusters. 4MB message size not supported)
  - Fix for memory spillover due to Type11 message size of 2044 (not 2048) for the last message of 64K MPI fragment
- **HPC**
  - Fixes for native compilation of examples on EVM.
- **CMEM**
  - Updated to release version 4.1.0.1

6.2.6 DEV.MCSDK_HPC.03.00.01.07

Changes from previous release:

- **OpenMPI**
  - Routing Algorithm enhancements to avoid loops in a cluster.
    NOTE: srio_topology.bin has been changed and needs to be re-created using the newer topologyJson2bin utility from the release.
  - Buffer descriptors and PDSP forwarded packets are placed in MSMC memory.
  - SRIO Type11 priorities fixed
  - RoutingTableGenTest Utility has been updated to specify the nodes themselves for simulation. Also routing connections in the cluster are displayed graphically as well.
  - Add SRIO link re-discovery in OpenMPI over SRIO
  - Fix for concurrent execution of dgmemm example with IMB -1MB
- **OpenCL**
  - Faster workgroup handling on the dsp. The file /var/lock/opencl will now be locked just in time within a process at the point when OpenCL is first used. Previously, the lock was created at startup of the process, even if OpenCL was not invoked for some time. The OpenCL library creates a host cpu thread to monitor traffic to and from the DSP device. Previously that thread would consume a host CPU. It has been improved to not consume any CPU resources unless there is an OpenCL active kernel on the DSP device. When there is an active kernel on the DSP device, it will consume resources at a nice level of 4. This can reduce the CPU impact of the monitoring thread. Additionally an environment variable TI_OCL_WORKER_SLEEP can be set to a non-negative number that represents the number of micro seconds that the monitoring thread should sleep after each device check. This will reduce the cpu overhead of the monitor thread but will increase the round trip time for any given kernel by the sleep value specified plus approximately an additional 70 micro seconds. A value of 100 microseconds is sufficient to lower the thread overhead to near 0%.

- Caching of kernel compiles by the OpenCL library are now off by default and enabled by the environment variable TI_OCL_KERNEL_CACHE. Previously, caching was enabled by default and was disabled by the environment variable TI_OCL_CACHE_KERNELS_OFF.
- Double-precision floating point support is no longer enabled by default in the OpenCL library and OpenCL C compiler. It can be re-enabled with the environment variable TI_OCL_ENABLE_FP64. Without double-precision support, the TI OpenCL implementation passes all Khronos OpenCL conformance tests and will be submitted for acceptance. When double-precision is enabled withTI_OCL_ENABLE_FP64, the double-precision built in functions are not conformant to the OpenCL spec. This is primarily due to the spec requirement that double-precision operations handle sub-normal value sproperly and the DSP hardware does not support sub-normals. The DSP hardware implements flush to zero behavior for sub-normals.

- Fix sub-buffer alignment check.

- **Openmpacc**
  - Update to CLACC to simplify cross-compiling on x86. $TARGET_ROOTDIR/usr/include is now added to the search paths for locating headers. This enables the source-to-source compiler locate ti_omp_device.hti-
- **CMEM**
  - Cache-coherency fixes involving the following
    - Change vm_page_prot WRITETHROUGH to WRITEALLOC for CMEM_CACHED case
    - Fix for cache handling of physical address.

- **HPC**
  - Automated tests included as a part of the release. Please refer to the below for usage.
  - Increased MSMC memory allocation to account for increased MSMC requirement of MPI.

6.2.7 DEV.MCSDK_HPC.03.00.01.06

Changes from previous release:

- OpenMPI
  - Fix for ORTE intermittent failure in MPMD mode of operation.
- OpenCL
  - Reduced the CPU load for the OpenCL thread that is created to monitor communication traffic to and from the DSP device. Previously that thread would consume 100% of a CPU during the existence of the t read. With this modification, the thread will only consume 100% of a CPU while the DSP is currently executing kernels. Otherwise, the thread will consume minimal resources.

  - Fix subbuffer alignment check..

6.2.8 DEV.MCSDK_HPC.03.00.01.05

Changes from previous release:

- OpenMPI
  - Use PDSP based packet forwarding, instead of SRIO hardware IP based packet forwarding
  - This brings in significant infrastructure under MPI
    - Navigator/Queue Management Sub System
    - Resource Manager: RM server daemon started as part of Upstart Job

        - PDSP firmware
        - SRIO type-11 wrapper for interfacing with SRIO BTL in MPI
- MPM-transport
    - Robustness in Hyperlink SERDES start-up sequence
    - Hyperlink @ 6.25 Gbps per lane support
    - Eliminate /dev/mem accesses to restore non-root user access for OpenMPI
- OpenCL
    - Fix an MPAX allocation issue where subbuffers >= 1Gb could not be used
    - Fix issue compiling opencl kernels with barriers
    - Fix a dsptop issue by adding a ulm termination call
    - Bug fix to work with PyOpenCL (Python OpenCL)
- MPM
    - Fix incorrect assumptions on installation order of uio_module_drv and the dsps. This was causing OpenCL to hang on certain reboots
- Update udev rule with permission 0660 and group keystone-hpc
- Additional 256 KB is reserved in MSMC for PDSP based SRIO routing
- Bug Fix in EDMAMgr (FC) in handling large strides in 3D transfers
- OpenMP accelerator: Install examples to EVM file system
- Debian packaging updates

6.2.9 DEV.MCSDK_HPC.03.00.01.04

Changes from previous release:

- MPM-transport
    - Reliable SERDES start-up sequence for Hyperlink
    - Implement timeout for Hyperlink port open, so that Hyperlink can be used only when 2 nodes are powered ON
- OpenMPI
    - Use CPU initiated transfers, instead of EDMA, for short messages over Hyperlink. This is for performance improvements
    - Extend support for 60 nodes interconnected via SRIO
    - Enhancement to do SRIO enumeration as part of MPI start-up
    - Bump up priority on NREAD requests (so they don't get timed out competing with packet forwarding data traffic)
    - Topology/Routing changes to avoid loops in a cluster (because, there'll be a LSU lock up with an SRIO loop). **NOTE**: After HPC installation is done, the file */etc/cluster/srio_topology.bin* needs to be regenerated, as per the instructions here
    - Packet forwarding fixes to exclude a node's own id, and to have no packet forwarding entries for end point nodes
    - Temporary workaround for LSU errors: Software throttling of SRIO traffic. Will be removed in next drop
- EDMAMgr (FC) update to handle large strides in 3D transfers
- Debian packaging updates
- **NOTE:** Only Root users can run MPI (MPM Transport users /dev/mem to poke hyperlink PSC registers). This restriction will be removed in future release

6.2.10 DEV.MCSDK_HPC.03.00.01.03

Changes from previous release:

- Addressed Intel MPI Benchmarking Test issues with Hyperlink

    - Random hangs in 30min to 3 hours with 2 nodes
    - Fails in all_gatherv w/ 128K buffer size with 4 nodes
- Hyperlink SERDES initialization moved from HPC upstart job to OpenMPI
- OpenMPI: Fix for Cartridge ID = 45 with BTL SRIO
- DSP Compiler (ti-cgt-c6x) upgraded to updated to 8.0.0 GA
- MPM Transport: Hyperlink SERDES initialization sequence changes and boost workaround
- DSPTOP: Documentation updates and changes to eliminate dsptop_sync file extension and source permissions
- OpenMP Accelerator:
    - Addition of host versions of DSP functions for cache and dynamic memory Management and a dspheap example
    - Bug Fix for memory leak in implementation for target update construct
- OpenCL:
    - Performance improvements of Improve kernels with barriers and those which uses 'reqd_work_group_size' attribute.
    - Bug fixes for clFlush() semantics and deadlock in event callback functions.
- FFTLIB: Fix for multicore overheads with OpenMP

6.2.11 DEV.MCSDK_HPC.03.00.01.02

Changes from previous release:

- New components included in MCSDK-HPC.
    - dsptop: This provides functionality similar to Linux top utility, i.e. visibility into usage data for TI DSP + ARM KeyStone II system-on-chips such as the 66AK2Hxx family. For more details, please see online wiki documentation
    - gdb6x: This allows developers to utilize the standard features of GDB (GNU Debugger) to gain visibility to and debug the C66x DSP cores in the heterogeneous DSP + ARM KeyStone II system-on-chips such as the 66AK2Hxx family. For more details, please see online wiki documentation
    - Supporting Kernel and user space modules for dsptop and gdb6x (debugss-mod, temperature-mod, gdbproxy-mod, gdbserver)
- OpenCL: Integration with dsptop, gdbc6x components
- OpenCL: Dangling pointer fix on the sgemm example
- OpenCL: Various bug fixes exposed by conformance testing
- OpenMP accelerator: GA release with OpenMP 4.0 target constructs
- Debian packaging updates in preparation for Partner Migration
- DSP Compiler (ti-cgt-c6x) upgraded to Beta-4 release.
- OpenMPI: Fixed the segmentation fault during SRIO routing simulation
- MPM: Updates for compatibility with Linux kernel 3.13
- MPM Transport: Serdes sequence updates for Hyperlink
- Framework Components: Updates to add support for 3D-edma transfers

6.2.12 DEV.MCSDK_HPC.03.00.01.01

Changes from previous release:

- Publicly available/ freely distributable C66x DSP Code Generation Tools (CGT) are provided with HPC installation (Users no longer need to download CGT separately)
- MPI: 6.25 Gbps over Hyperlink (needs new DTS, and updated MPM, MPM-transport components); Known Issues from 3.0.1.0 with OpenMPI+OpenCL and OpenMPI+OpenMPacc are addressed.
- OpenCL: Added opencl/openmp examples: sgemm, dgemm, fftlib_offload
- OpenCL: Separate LLVM into a separate package

- FFTLIB C66x Library updates to support OpenCL dispatch
- Several building blocks to support DSPTOP and SocTune are now available with MCSDK-HPC. Integration of debug features into OpenCL and collateral updates will happen as part of next release

6.2.13 DEV.MCSDK_HPC.03.00.01.00

Changes from previous release:

<u>OpenCL</u>

- Support for all clauses of target construct in OpenMPacc (SDSCM00050116)
- Added host memory allocation functions __malloc_ddr(), __free_ddr(), __malloc_msmc(), and __free_msmc(). These functions have the same interface as standard malloc/free but manages memory from host+dsp shareable memory. These can be used within the host application
- Added dsp memory allocation functions __malloc_ddr(), __free_ddr(), __malloc_msmc(), __free_msmc(), __malloc_l2(), and _free_l2(). These functions are available from dsp kernel code to manage memory from the dsp heaps. The OpenCL example 'dspheap' has been added to illustrate how the dsp heap memory can be allocated and managed between kernel invocations.
- Significantly improved OpenCL performance on cache coherency operations for buffer sizes of 4MB and above.
- The async_workgroup_copy and async_workgroup_strided_copy functions are now implemented to use EDMA channels when available. (SDSCM00050115)
- The number/size of Kernel arguments are now fully compliant with the OpenCL 1.1 spec. Up to 1024 bytes of arguments are supported and vector types can be passed as standalone arguments.
- On the dsp, the standard malloc() function will now allocate from an 8M pool of DDR.
- The OpenCL package version number can now be obtained by either invoking 'clocl --version' or making a standard OpenCL platform query. The OpenCL 'platform' example illustrates how to make an OpenCL platform query.
- The opencl runtime provides built in functions that would allow a user to explicitly change the L2 cache size. Since this call would occur in the DSP code, the user would need to ensure that the app did not allocate l2 SRAM that would conflict with their l2 cache size choice. (SDSCM00050114)
- Setting the environment variable TI_OCL_CGT_INSTALL is not required if the C6000 compiler runtime library and header files are located in the ppa package default install path. TI_OCL_CGT_INSTALL can still be used to specify an alternate search path.
- The environment variable used to enable OpenCL kernel debugging has been changed from TI_OCL_DEBUG_KERNEL to TI_OCL_DEBUG.
-

<u>MPI</u>

- Latency and Bandwidth Improvements with SRIO (SDOCM00108189)
  - Latency: 4uS, BW: up to 240MB/s (@5gbps)
    - SRIO user space module for RMA
    - CMEM cached allocated message buffers
- Latency and Bandwidth Improvements with Hyperlink
  - Latency: 5.5uS, BW, up to 420MB/s (@4x3.125gbps)
    - CMEM cached allocated message buffers
    - EDMA linked transfers for messages >4Kbytes
    - Reduced EDMA setup time w/ preallocated EDMA resources
- Sudo privilege was needed for using MPI over SRIO. Now, it is non-root users can also use MPI over SRIO (SDOCM00107864)
- MPI using SRIO was broken for cartridges enumerated above 7 (c7) (SDOCM00107865)
- Reliable SRIO mports don't come up (sometimes) when all the nodes are not booted at once (SDOCM00107669)

- MPI latency measurement triggering memory leakage (BTL SRIO & Hyperlink) (SDOCM00107945)
- **KNOWN ISSUE**:
  OPENMPI runtime by itself works but when used with OpenCL or OpenMPacc doesn't work. However, there is a work around for examples involving Hyperlink interchip connection. The work around is as below.
  - Vecadd_mpi example (OpenMPI+OpenMPacc), should use additional mca parameter as temporary workaround:

    ```
    /opt/ti-openmpi/bin/mpirun -x LD_LIBRARY_PATH -x TI_OCL_INSTALL -x TI_OCL_CGT_INSTALL -x
    TI_OMPACC_INSTALL --mca btl_hlink_use_edma 0 --mca btl self,hlink -np 2 -host c7n1,c7n2
    vecadd_mpi
    ```
  - Multimode_batch_fftdemo example (OpenMPI+OpenCL),should use additional mca parameter as temporary workaround:

    ```
    /opt/ti-openmpi/bin/mpirun --mca btl_hlink_use_edma 0  --mca btl self,hlink -np 2 -x
    LD_LIBRARY_PATH -x PATH -x OPAL_PREFIX -host c7n1,c7n2 ../host/demo_multinode_batch_fft
    ../testfiles/fftw_testfile.inp ../testfiles/fftOut.bin
    ```

  These issues will be addressed in upcoming release.

FFTLIB

- Provide pre-built Library, along with Makefiles for Library and Unit-tests, in addition to CCS projects (SDOCM00107667)

OpenMP-DSP

- Added dynamic heap management APIs on the DSP

Debian Security Violations

- Change CMEM auto-insertion to an upstart job, instead of modifying /etc/modules (modifying(/etc/modules is a debian security violation) (SDOCM00107905)

6.2.14 DEV.MCSDK_HPC.03.00.00.19

Changes from previous release:

- Requires MCSDK 3.0.4.18 (GA). Incompatible with previous MCSDK releases.
- All bundled components promoted to GA status (except for OpenMPacc)
- OpenMPI over SRIO with packet forwarding for switch-less topologies
- Default Hyperlink bandwidth increased from 3.125 to 6.25 Gbps
- Support to natively compile all out of box examples on EVM
- All out of box examples (opencl, openmp, openmpi, hpc) have been consolidated and installed to /usr/share/ti/examples
- mpi_examples present as separate package in previous release have been removed
- Dropbear patch has been removed
- Additional documentation and collateral on wiki pages
- HPC additions to the file system are delivered as IPKs and via apt-get

6.2.15 DEV.MCSDK_HPC.03.00.00.18

- First public release (Beta)
- Release bundles the following components:
  - OpenMP DSP runtime.
  - OpenCL
  - OpenMP Accelerator Model

- OpenMPI w/ SRIO and hyperlink support
- FFTLIB
- BLAS
- Framework Components with EdmaMgr support
- Examples exercising combinations of each of the components.

- More details available at the MCSDK HPC Getting Started Guide (http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Getting_Started_Guide)