ClearSpeed

Base Package for Release 3.0
Release Notes

Release Notes for the Base Package

This document describes the most important changes to the base package since the previous release. In addition, it lists the known open issues and limitations in this release.

For more information regarding the status and workarounds related to any of these issues, please contact ClearSpeed support quoting the relevant CTS number.

You should check the ClearSpeed customer support website (http://support.clearspeed.com) for updates to these release notes.

1 What's new in Release 3.0

Release 3.0 contains many improvements to the development tools which have been made since Release 2.51. The major changes are summarized in this section.

1.1 Operating systems

This release supports use of ClearSpeed's products on a wider range of operating systems. The supported operating systems are:

- 32- and 64-bit Red Hat Enterprise Linux 4
- 64-bit Red Hat Enterprise Linux 5
- 64-bit SUSE Linux Enterprise Server 9
- 64-bit SUSE Linux Enterprise Server 10
- 32-bit Microsoft Windows XP and Windows Server 2003
- 64-bit Microsoft Windows Server 2003 and Compute Cluster Server 2003

Note: *Please note that there is a single Windows installer for both 32- and 64-bit Windows. It automatically installs the 64-bit driver on 64-bit operating systems. See CTS 4591 for more information about using the software on 64-bit Windows operating systems.*

1.2 CSXL library

The CSXL library file has been replaced with two versions; one is ACML specific and the other is for MKL. The CSXL MKL version is compatible with most other host BLAS libraries. The appropriate CSXL library should be called by the program linker and/or referenced in any application environment variables. See the *CSXL User Guide* for more details.

The CSXL library supports a greater number of BLAS and LAPACK functions as well as a version of the DGEMM function that can be called directly by code running on the Advance card. The following Level 3 BLAS and LAPACK functions are directly accelerated by this version of the CSXL library:

- `DGEMM`, the BLAS level-3 routine that multiplies a real double precision matrix by a real double precision matrix.
- `ZGEMM`, the BLAS level-3 routine that multiplies a complex double precision matrix by a complex double precision matrix.
- `ZGEMM3M`, an implementation of `ZGEMM` that requires three real matrix multiplications and five real matrix additions to compute the complex matrix product; `ZGEMM` uses four real matrix multiplications and two real matrix additions. See the *CSXL User Guide* for additional information. This may be faster than the standard implementation under certain circumstance.
- `DTRSM`, the BLAS level-3 routine that solves a real double precision triangular system of equations with multiple right-hand-sides. This routine constitutes a large percentage of the computation done in the LAPACK routines that factor and solve a general system of linear equations, respectively `DGETRF` and `DGETRS`.
- `DGETRF` and `DGETRS`, the LAPACK routines that factor and solve a real double precision general system of linear equations using the LU method.

- `DPOTRF` and `DPOTRS`, the LAPACK routines that factor and solve a real double precision symmetric positive definite system of linear equations using the Cholesky method.
- `DORGQR`, the LAPACK routine that generates all or part of the orthogonal matrix Q from a QR factorization computed by `DGEQRF`.
- `DORMQR`, the LAPACK routine that multiplies a matrix by the orthogonal matrix Q from a QR factorization computed by `DGEQRF`.

1.3 Runtime software

There have been some significant changes to the host interface library, CSAPI. See the *CSAPI Migration Guide* for detailed information on changes you may have to make to your code for this release.

The board diagnostics package including the test program, `run_tests.pl`, has been removed in this release. Further diagnostic tools will be made available in future releases.

The `csrun` program has been enhanced to allow command line parameters to be passed to the `main` function of a C^n program running on the Advance card. See the *Runtime User Guide* for more information.

In previous releases there was an (undocumented) utility called `temp_mon` which could be used to measure the temperature of an Advance accelerator cards. This is no longer supported.

Over temperature handling (X620 cards only)[1]

Caution: Please consider the information that follows as failure to do so can damage the card or produce erroneous results.

If the temperature threshold is breached, a warning message is printed to inform the user. For example:

> Warning: Processor 0 board temperature alert threshold reached on instance 0.
> Current core temperature is 90 degC, board temperature is 70 degC.
> Further temperature alerts from Processor 0 will be ignored until reset.

This means that either the card or core is running at a temperature which is above its environmental limits. This may cause errors in the results and risk permanent damage to the card[2].

The warnings do not stop the use of the card. However it is a significant risk to continue without improving cooling to the card.

If the card or core temperatures continue to rise and exceed 90^{o}C (card) or 110^{o}C (core) then the card will immediately shutdown to prevent damage. This is likely to cause the host application using the card to hang or crash.

1. Please refer the *X620 Card User Guide* for more information.
2. A log file on the card stores information regarding running over temperature.

The warning message can be stopped by setting the environment variable CS_DISABLE_TEMPERATURE_ALERT. However we would strongly advise that users do NOT use this option and that the temperature warnings are heeded.

2 Issues fixed in Release 3.0

The following issues have been fixed in this release:

CTS 1820: The functions:

```
CSAPI_read_mono_memory_async_wait
CSAPI_read_mono_memory_async_poll
```

and their `CSAPI_write` counterparts did not return an error code if the asynchronous transfer failed.

CTS 4128: The diagram showing the memory map for regions allocated by the CSAPI functions of the **Runtime User Guide** did not appear in some versions of the document.

CTS 5014: The CSAPI function `CSAPI_semaphore_wait` was not thread safe if the semaphore being waited on had already been signalled on entry.

CTS 5073: The CSXL `ZGEMM` function did not work correctly for all matrix sizes. In the case where `LDB` < `LDC`, that is, the leading dimension of the matrix `B` is less than the leading dimension of the matrix `C`, the matrix `B` will be overwritten. This can cause the program to terminate abnormally.

CTS 5204: It was possible for address calculations in CSXL to overflow when using very large matrix sizes (larger than about 7 GB). This could only happen on 64 bit host systems.

3 Known Issues in Release 3.0

3.1 Runtime

CTS 239: `csrun` or host client applications cannot check whether the CSX processor has been reset. Running code on a processor that has not been reset should not be attempted. It is the responsibility of the user to reset the processor before running code (using `csreset -A`).

CTS 2004: This release includes a script for resetting the Advance accelerator when `csreset` fails to do so. This is less likely to occur in this release than previously.

This script does a 'hard' reset of the processors. Before using the reset script, please gather any diagnostic or debugging information as all state will be lost by the hard reset. For example, the output from `csreset -v.`

Before running the script, first setup your environment if you have not already done so. Under Linux, source the `bashrc` file (usually present in `/opt/clearspeed/csx600_m512_le/bin`). For Windows, start a command prompt using the shortcut from the ClearSpeed Start menu item. If you have more than one board, set the environment variable `LLDINST` to the instance number of the board to be recovered. For example, to reset only the first board under Linux enter `export LLDINST=0`

To run the script, type the command `recover_board`. You should then see some output like this:

```
Board recovery utility

This should only be used:
- when csreset fails to reset your board
- after any useful diagnostic information has been gathered (e.g. the
output from csreset -v).

If you wish to continue, press the return key. Otherwise, press
control-c to exit.

If you are happy to run, then press the return key. You will then see
output as follows:

Starting...
25%
50%
75%
DONE.

Board recovery attempted - you can now re-run csreset.
```

CTS 3161: The Release 3.0 runtime software will not work with obsolete versions of the firmware on the Advance cards.

See the firmware upgrade release notes on the customer support site (http://support.clearspeed.com) for details.

CTS 5702: If a semaphore handle is passed to `CSAPI_allocate_duplicate_shared_semaphore`, for allocation on a different card, it cannot be released using `CSAPI_free_semaphore`. The semaphore allocations can only be released by deleting the card state and creating a new one.

CTS 5703: The hot pragma described in the SDK Reference Manual is not fully implemented in this release. The toolchain generates a text section (`.text.onchip`) containing the code marked as 'hot'. This section should be loaded into on-chip memory (ESRAM). However, the `CSAPI_load()` function does not do this by default. Therefore, when using the `csrun` command to run a program on an Advance card, the functions marked as hot will not be loaded into ESRAM.

If the host application loads and runs the CSX code, it can then instruct `CSAPI_load()` to put these functions in ESRAM by setting the `section_names` parameter to `".text.onchip"`.

3.2 CSXL

CTS 1108: If a host application program using the CSXL library is terminated abnormally (for example, by using [Ctrl]+[C]), the Advance Accelerator board may be left in an undefined state. It may be necessary to reset the board (using the `csreset` command) before restarting the application.

CTS 3003: CSXL does not work with the currently released version of the Goto BLAS host library. Please contact ClearSpeed support (via the support web site http://support.clearspeed.com) for more information on updates and workarounds for this issue.

CTS 3891: This release of CSXL does not support the C pass-by-value interface for ACML. The C pass-by-reference and Fortran interfaces are supported.

CTS 4591: ClearSpeed provides a driver for the supported 64-bit Microsoft Windows operating systems (Windows Server 2003 and Compute Cluster Server) but currently we only have 32-bit libraries and tools. It is not possible to mix 32 and 64-bit application software and so only 32-bit applications can be used with the CSXL library, for example.

If you are using a 64-bit Windows operating system, you can install the ClearSpeed software for Microsoft Windows. The runtime package will install a 64-bit driver on 64-bit operating systems. You may now use these components as you would on a 32-bit operating system with the caveat that the application and all the libraries that you run against the installed 32-bit ClearSpeed software stack must also be 32-bit.

So for example if you have a 64-bit native version of MATLAB installed this will not work correctly and you will have to install the 32-bit version of the application software. Similarly if you build (ATLAS or GotoBLAS) or install (Intel MKL or AMD ACML) math libraries from third party sources then you will need to make sure they are also 32-bit versions.

CTS 4633: The current implementations of ZGEMM and ZGEMM3M are based on calls to DGEMM. The code for ZGEMM and ZGEMM3M requires temporary arrays which are roughly the size of A and B for ZGEMM and the size of A, B and C for ZGEMM3M. For large matrices this can exceed available memory on the host system. When this occurs the matrices are partitioned creating smaller ZGEMMs that have temporary arrays which can be allocated. Each smaller ZGEMM is performed sequentially.

Note: *For optimum performance the dimensions m and n provided to ZGEMM / ZGEMM3M should be a multiple of 192 and k should be a multiple of 288, where possible.*

CTS 5717: Each of the DGEMM matrices (A, B and C) are limited to 4e+9 elements. For square matrices the would limit the dimensions to m=n=k=65535. Incorrect results are expected beyond this limit.

CTS 5584: MKL has changed the way linkage with the different libraries work in version 10. Currently MKL version 10 will work with CSXL on Linux but not on Windows.

On Linux, when using MKL version 10 with CSXL, the environment variable CS_HOST_BLAS is specified differently than when using older versions of MKL with CSXL.

For MKL 10, CS_HOST_BLAS must include the interface library, threading library and computational library. When calling BLAS/LAPACK routines, you must include libmkl_intel_lp64.so, libguide.so, libmkl_intel_thread.so and libmkl_core.so. If for example MKL is installed in /opt/intel/mkl/10.0.3.020 and you are using Intel or GNU compilers, then

```
% export CS_HOST_BLAS = \
/opt/intel/mkl/10.0.3.020/lib/em64t/libmkl_intel_lp64.so: \
/opt/intel/mkl/10.0.3.020/lib/em64t/libguide.so: \
/opt/intel/mkl/10.0.3.020/lib/em64t/libmkl_intel_thread.so: \
/opt/intel/mkl/10.0.3.020/lib/em64t/libmkl_core.so
```

We have tested CSXL with MKL 10. If you encounter failures when using MKL 10, we recommend using MKL 10.1 Beta or later (if applicable), as test failures we have encountered are resolved in this Beta. If you continue to see issues, please report these via a support request to ClearSpeed at http://support.clearspeed.com/support/case/.

ClearSpeed Technology, Inc.
3031 Tisch Way, Suite 200
San Jose, CA 95128
United States of America

Tel: +1 408 557 2067
Fax: +1 408 557 9054

Email: info@clearspeed.com

Web: http://www.clearspeed.com

Support: http://support.clearspeed.com

ClearSpeed Technology plc
3110 Great Western Court
Hunts Ground Road
Bristol BS34 8HP
United Kingdom

Tel: +44 (0)117 317 2000
Fax: +44 (0)117 317 2002