Intel® Cluster Toolkit 2.0.1
Release Notes
(Revision 20060424)

Contents

Overview

The Intel Cluster Toolkit 2.0.1 accelerates parallel software development on homogenous Intel® Pentium®, or Intel® Xeon®, or Intel® Itanium® 2, or Intel® EM64T-based systems running Linux*. It supports application development using Intel® MPI Library with optimized parallel libraries, performance analysis, and benchmarks. The Intel Cluster Toolkit 2.0.1 saves software developers time and improves execution performance on distributed computing systems.

Intel Cluster Toolkit 2.0.1 supports crucial parts of the message-passing interface (MPI) application development process including:

- Intel MPI Library 2.0.1 which is approaching conformity with the Message Passing Interface 2 Standard (MPI-2), enables multiple interconnect solutions with a single implementation. Intel seeks to be a software leader in MPI and open standards.

- The Intel® Trace Analyzer and Collector 6.0.1
    - Intel® Trace Collector 6.0.1 provides event-based tracing in cluster applications through an instrumentation library that causes low-overhead in execution. The trace information provides performance data, statistics, multi-threaded events, and automatic instrumentation of user binaries on IA-32 architectures.
    - The Intel® Trace Analyzer 6.0.1 provides visual analysis of application activities gathered by the Intel Trace Collector. This software component has been completely rewritten.
- Application tuning with optimized mathematical library functions from Intel® Math Kernel Library Cluster Edition 8.1 (Intel® MKL Cluster Edition 8.1) that includes ScaLAPACK solvers and Cluster DFTs (Discrete Fourier Transforms).
- Intel® MPI Benchmarks 2.3 make it easy to gather performance information about a cluster system.

Note that when the system administrator or user completes the Intel Cluster Toolkit 2.0.1 installation process, there will be a file called `index.htm` in the `doc` folder on the master node of the cluster. This file can be used as a documentation map to navigate to various information resources pertaining to the Intel Cluster Toolkit. Additional information about the exact location of `index.htm`, and its content are further described in the **Installation** section of this release notes document.

Back to Top

New Features

Intel Cluster Toolkit 2.0.1

- The Intel Cluster Toolkit has an `expect` script called `sshconnectivity.exp` which the user may find useful in establishing secure shell connectivity on a cluster system. For the 2.0.1 release, this `expect` script has been improved to create private and public keys when a login session for each node of the cluster has its own unique `.ssh/` directory. Please see the section titled “Alternative Solution to Establishing Secure Shell Connectivity” below.
- The Intel Cluster Toolkit has an `expect` script called `sshconnectivity.exp` which the user may find useful in establishing secure shell connectivity on a cluster system. For the 2.0.1 release, this `expect` script has been improved to create private and public keys when a login session for each node of the cluster has its own unique `.ssh/` directory. Please see the section titled “Intel Software Downloads and Installation for the Intel Cluster Toolkit”.
- The installer analyzes node configurations and searches for "NFS traps". This is a common error where the user tries to install the Intel Cluster

Toolkit on all cluster nodes into a folder that is mounted on at least some nodes of the cluster via NFS.

- The Intel Cluster Toolkit release notes provide a link for downloading `expect`. This link might be useful, if this software is not already installed on the cluster system.
- The user's guide and the tutorial descriptions for using the Intel MKL Cluster Edition Cluster DFT (Discrete Fourier Transform) software have been improved by consolidating information into tables. The tables are categorized by Intel processor-based architecture.
- The user's guide and the tutorial descriptions for using the Intel MPI Benchmarks have been improved by consolidating information into tables. The tables are categorized by Intel processor-based architecture.
- The user's guide and the tutorial provide a description for altering the number of MPI processes when running the ScaLAPACK test suite.
- The user's guide and the tutorial provide a description for altering the number of MPI processes when running the Cluster DFT (Discrete Fourier Transform) applications.

Intel MKL Cluster Edition 8.1

- Improvements for the Intel® Itanium® 2 processor
  - BLAS
    - Level 3 BLAS improvements
      - DTRMM - 100-150% on small sizes (up to 100-200) and 2-8% on big sizes
    - Level 1 BLAS improvements
      - DSWAP, SSWAP, ZSWAP, CSWAP - up to 1.5-2 times.
      - DASUM - 30-40% in memory (for sizes that are bigger than 500,000)
      - IDAMAX, IDAMIN, ISAMAX, ISAMIN - 30-40% on cache
        - ZAXPY, CAXPY - up to 1.5 times on L2 and memory
        - SAXPY - up to 1.5 times
  - FFTs
    - 1D FFT
      - Double and single complex data N < 500 - from 2% up to 5 times
      - Double and single real data N < 900 - from 2% up to 5 times
    - 2D FFT
      - Double and single complex data M*N < 6000 from 2% up to 2.2 times
      - Double and single complex data M*N, if M or N = 4, 8, 16, 32, 64 up to 1.5 times
      - Double and single real data less M*N < 6000 from 2% up to 1.5 times
      - Double and single real data M*N, if M or N = 4, 8, 16, 32, 64 up to 7%
    - 3D FFT
      - Single complex data M*N*K < 300000 from 2% up to 1.5 times

        - Single complex data M*N*K, if M or N = 4, 8, 16, 32, 64 up to 20%
        - Double complex data M*N*K < 100000 from 2% up to 1.7 times
        - Double complex data M*N*K, if M or N = 4, 8, 16, 32, 64 up to 17%
        - Double and single real data less M*N*K < 60000 from 2% up to 16%
        - Double and single real data M*N*K, if M or N = 4, 8, 16, 32, 64 up to 5%
    - Multiple 1D FFTs with single function call (distance = 1)
        - Single complex data: up to 4.5 times for in-place, up to 3.5 times for out-of-place, average 1.5-2.5 times
        - Double complex data: up to 3 times for in-place and out-of-place, average 1.5-2.5 times
        - Single real data: up to 5 times for out-of-place, average 1.5-2.5 times
        - Double real data: up to 2 times for out-of-place, average 1.5-2.0 times
- Improvements for the Intel® Pentium® 4 processor and Intel® Xeon® processor with SSE3.
    - BLAS
        - Level 3 BLAS
            - Improved DGEMM for processors with Intel EM64T by 1-3%
            - Improved SGEMM with Intel EM64T by 1.5 - 2 times
            - Improved ZTRSM (L side) with Intel EM64T by 50% on small sizes and by 5% on large sizes
            - Improved ZGEMM by 10 - 40%
        - Level 2 BLAS improvements
            - ZGEMV - 10-30%
            - CGEMV - 8-15%
        - Level 1 BLAS improvements for processors with Intel EM64T
            - DDOT - 35% on cache
            - DGEMV - 40% on cache
            - ZDOTU, ZDOTC - 30-90%
            - ZAXPY - 20-50%
            - ZSCAL - 20-40%
    - FFT improvements for processors with Intel EM64T
        - 1D FFT
            - Double and single complex data N < 100 - from 3% up to 5 times
            - Single complex data N = 2**K, K < 16 - up to 30%
            - Double and single real data N < 200 - from 3% up to 3 times
            - Single real data N = 2**K, K < 16 - up to 20%
        - 2D FFT
            - Double and single complex data M*N < 500 from 2% up to 1.6 times
            - Single complex data M*N, if M or N = 2**K up to 18%

        - Double and single real data less M*N < 500 from 2% up to 1.5 times
        - Single real data M*N, if M or N = 2**K up to 14%
      - 3D FFT
        - Single complex data M*N*L < 300 from 2% up to 40%
        - Single complex data M*N*L, if M or N or L = 2**K up to 1.5 times
        - Double complex data M*N*L < 300 from 2% up to 1.5 times
        - Single real data M*N*L < 300 from 2% up to 40%
        - Single real data M*N*L, if M or N or L = 2**K up to 6%
        - Double real data M*N*L < 1000 from 2% up to 11%
      - Multiple 1D FFTs with single function call (distance = 1)
        - Single complex data - in-place: up to 2.5 times for 1 threads 5 times for 2 threads, average 1.5-2 times
        - Single complex data - out-of-place: up to 3.5 times for 1 threads, 6 times for 2 threads, average 1.5-2.5 times
        - Double complex data - in-place: up to 3 times for 1 threads, 3.5 times for 2 threads, average 1.3-2 times
        - Double complex data -out-of-place: up to 2.5 times for 1 threads, 3.5 times for 2 threads, average 1.2-2 times
        - Single real data - out-of-place: up to 4 times for 1 threads, 5 times for 2 threads, average 1.5-2.5 times
        - Double real data - out-of-place: up to 3.4 times for 1 threads, 4 times for 2 threads, average 1.5-2.5 times
    - VML improvements for processors with Intel EM64T
      - Performance of some variants of Inv, Div, Ln, Log10, Tan, Asin, Atan, Atan2, Acosh, Asinh, and Atanh improved by 25% - 250%
- Improvements for the Intel® Core Duo processor
  - VML performance improved by approximately 35%
  - VSL performance improved by approximately 35%

Other Intel MKL Cluster Edition 8.1 Improvements

- Direct sparse solvers
  - PARDISO* update with pivoting for symmetric indefinite matrices
  - Fixed unresolved external symbol __iob referenced in function _PRINT_CI in PARDISO*
- LAPACK
  - Fixed CLAED6/DLAED6/SLAED6/ZLAED6.
  - Fixed CGEESX/DGEESX/SGEESX/ZGEESX, ZGGESX/CGGESX/DGGESX/SGGESX, and DORMQR/SORMQR/CUNMQR/ZUNMQR to correct workspace inquiry issue.
- BLAS
  - Performance improvements for matrix-vector and matrix-matrix sparse BLAS routines in the case of the compressed sparse row and diagonal formats. All these routines are OpenMP* threaded for general matrices.

- Performance improvement for matrix-vector and matrix-matrix multiply routines for the compressed sparse row and compressed sparse column formats in the case of symmetric, skew-symmetric and triangular matrices. Performance gain in the serial mode varies from 20-100% depending on type and structure of a sparse matrix. The matrix-matrix multiply routines are OpenMP* threaded. Fixed diagnostics issue in Sparse BLAS routines.
- Fixed DGEMM, ZGEMM for compatibility mode running 32-bit code on Intel EM64T
- Fixed STRSM/CTRSM/ZTRSM and CTRMM/ZTRMM for parallel mode on 8 threads or more.

- FFTs
  - FFTW 2.x interface support.
  - Fixed the issue for the case when DFTI_USERS_THREADS > 1 and OMP_NUM_THREADS > 1
- VSL
  - The files “mkl_fsl.fi” and “mkl_vsl_subroutine.fi” are compatible with both fixed and free formats of FORTRAN sources.
- Convolution/Correlation
  - Double precision calculations via FFT are supported for 1-dimensional convolution and correlation.
  - Output decimation is supported for all convolution and correlation methods including the calculations "via FFT" mode.
- FORTRAN 90/95 interfaces to LAPACK and BLAS subroutines are now available as source code in the interfaces directory.
- Service
  - Added case sensitive MKL_FreeBuffers() entry point.
- ScaLAPACK (available only in the Cluster Edition)
  - Fixed a few minor, rare bugs
  - Reduced the amount of scratch memory used
  - Improved performance of some of the larger blocking sizes for LU
  - Improved pivoting performance for LU
  - BLACS interface to support SGI* Message Passing Toolkit (MPT)

Intel MPI Library 2.0.1

- Increased application performance
  - Control over the point-to-point protocol threshold through the I_MPI_EAGER_THRESHOLD environment variable
  - Optimized collective operations controlled by the I_MPI_FAST_COLLECTIVES environment variable
  - Extended process pinning for SGI Altix* systems controlled by the I_MPI_PIN_MODE and I_MPI_PIN_PROCS environment variables
  - Optimized memory registration controlled by the I_MPI_RDMA_TRANSLATION_CACHE environment variable
- Improved stability and correctness
  - Control over the MPD* daemon startup connection phase through the I_MPI_MPD_CONNECTION_TIMEOUT environment variable

    - Co-existence of several MPD* rings under one user id with the help of the MPD_CON_EXT and I_MPI_MPD_CONF environment variables
    - Synchronized fallback upon the default device after detection of the DAPL provider issues at job startup
    - Control over the fallback upon the default device through the I_MPI_USE_DEFAULT_DEVICE environment variable
    - Control over the DAPL connection establishment phase through the I_MPI_DAPL_CONNECTION_TIMEOUT environment variable
    - Several application relevant bug fixes
- Increased interoperability
    - Optional non-modification of the /etc/ld.so.conf file during product installation through the `--update-ldsoconf=(yes|no)` installer option
    - Extraction installer capability for non-RPM* based systems through the --extract installer option
    - Selection of alternative TotalView* executable files through the TOTALVIEW environment variable
- Enhanced operating system support
    - SGI* ProPack 4
    - Red Flag* DC Server, version 5.0
    - SuSE*, version 9.3 and 10.0
- Integration with the leading job schedulers
    - LSF*, version 6.1 and higher
    - PBS Pro*, version 7.1 and higher
    - Torque*, version 1.2.0 and higher
- Extended compiler support
    - Intel® Compilers for Linux*, version 9.0 and higher
    - GNU* compilers, version 4.0 and higher
    - Absoft* compilers, version 9.0 and higher
    - PGI* compilers, version 6.0 and higher
- Message queue browsing support in leading parallel debuggers
    - DDT*, version 1.9.2 and higher
    - TotalView*, version 7.1.0 and higher

Intel Trace Analyzer 6.0.1

- Improved presentation of pre-defined and auto generated groups
- New default layout in Collective Operation Profile
- Corrected several Grouping issues
- Corrected problem with Windows* version crashing after using thousands of Functions
- Corrected problem where the colors associated with new “fgroups” and “fgroups” where the structure has been changed are lost
- Corrected wrong time total calculation in Flat Function Profile where recursive calls were involved
- Corrected problem where no timeline or timescale were created by "New View" dialog
- Corrected problem with wrong data shown in call graph

- Corrected problem where a reversed message led to incorrect data in profiles/timelines
- Corrected problem where disabling of refreshing View fails
- Corrected problems with using the Japanese character set
- Added minor extensions and corrections to the documentation
- Intel Trace Analyzer for Windows* installer has been updated
- Added ability to open a new trace file while Intel Trace Analyzer application is still running, given that all views of old trace file are closed
- Intel Trace Analyzer and Collector settings added in ictvars.[c]sh scripts
- Corrected floating point exceptions inside VT_LogMerge() for Itanium systems
- Enabled the Intel Trace Analyzer for Linux* installer to query the user whether the current installation of Intel Trace Analyzer (if one exists) should be overwritten
- Empty Java trace files handled properly by Intel Trace Analyzer
- Fixed inconsistency between number of function calls shown in detailed dialog view and drill-down profile
- Fixed bug relating to the demangling of symbol names “f”, “g”, and “aa” when instrumenting a binary

Intel Trace Collector 6.0.1

- itcinstrument is now working on EM64T
- Corrected floating point exception (SIGFPE) in Intel Trace Collector on Itanium
- Corrected problem where Intel Trace Collector crashes with VT_CONFIG_RANK <> 0
- Corrected problem where invalid send size is logged for MPI_Alltoall
- Corrected a deficiency where Intel Trace Collector documentation was incomplete: trace file options missing
- Intel Trace Collector OS counters: “net IO” counter logs increasing byte counts only
- Intel Trace Collector in conjunction with Bistro SDK (version 1.2.9 or later) works with Red Hat Enterprise Linux 4.0

Back to Top

System Requirements

Processor System Requirements
Intel Pentium 4 processor, or
Intel Xeon processor, or
Intel Itanium 2 processor, or
Intel EM64T-based processor

Note that it is assumed that the processors listed above are configured into homogeneous clusters.

Back to Top

Disk-Space Requirements
10 GBs of disk space (minimum)

Back to Top

Operating System Requirements

Linux Distributions	IA32	EM64T		Itanium 2
		32-Bit Applications	64-Bit Applications	
Red Hat Enterprise Linux* 3.0	S	S	S	S
Red Hat Enterprise Linux 4.0	S	S	S	S
SUSE Linux Enterprise Server* 9	S	S	S	S
SUSE Linux 9.0	S	S	S	N/A
SUSE Linux 9.1	S	S	S	N/A

S = Supported

Back to Top

Memory Requirements
1 GB of RAM (minimum)

Back to Top

Intel® Compilers
For all of the Intel processor architectures the version number on the Intel compilers should be 8.1 or greater.

Back to Top

Setting Up Secure Shell

Installation of the Intel Cluster Toolkit 2.0.1 assumes that the homogenous cluster has `ssh` connectivity.

1. Make sure `ssh` is installed on your computing cluster. To do this type the shell command:

```
which ssh
```

   If it does not exist, one can acquire `ssh` from the URL:

   http://www.openssh.org

2. Create an authentication key as follows:

```
ssh-keygen -t rsa
```

   This will generate a private and public key pair. The private key should be saved in:

```
~/.ssh/identity
```

   and the public key should be saved in:

```
~/.ssh/identity.pub
```

3. Authorize access by placing the contents of the public key into the `~/.ssh/authorized_keys` file. All keys listed in that file are allowed access. One way to do this is to issue the shell command:

```
cat ~/.ssh/identity.pub >> ~/.ssh/authorized_keys
```

   If the computing system that you are connecting to does not share a common file system, then `~/.ssh/identity.pub` should be concatenated to the `~/.ssh/authorized_keys` file of the computing system you will be connecting to. Secure Shell will insist that the file `authorized_keys` has its permissions set so that it is not group writable, so do the following:

```
chmod go-rwx ~/.ssh/authorized_keys
```

   This step avoids the need to enter your password each time you want to run a secure shell command.

4. In order to avoid typing in your pass phrase each time `ssh` is invoked, an `ssh-agent` needs to be created and your pass phrase added. This is done as follows:

```
ssh-agent $SHELL

ssh-add
```

Back to Top

Checklist in Case There Are Problems

1. Make sure that the host names listed in the `machines.LINUX` file are also listed in the `/etc/ssh_known_hosts` file on your network or your `~/.ssh/known_hosts` file in your home directory.
2. It is important that `/tmp` has permissions set to `377`, with root as owner and group 0.
3. `openssh` has a `-v` flag option which is very useful for tracking down handshaking problems.
4. If you encounter the following message:

   "scp: FATAL: Executing ssh1 in compatibility mode failed (Check that scp1 is in your PATH). Lost connection."

   Then there is an incompatibility with the secure shell protocol between the master node and at least one of the other nodes listed in the `machines.LINUX` file. The solution is to install compatible versions of secure shell on all nodes of the cluster.

Back to Top

Alternative Solution to Establishing Secure Shell Connectivity

Within the “tar” package of the Intel Cluster Toolkit, there is an `expect` shell script file called `sshconnectivity.exp`. You can download the `expect` software package from the following URL:

http://expect.nist.gov/

The expect shell script `sshconnectivity.exp` can be used to help the user establish secure shell connectivity on a cluster system. The syntax for the command is:

```
./sshconnectivity.exp machines.LINUX
```

This `expect` shell script will create or update a `~/.ssh` directory on each node of the cluster beginning with the master node which must be the first name listed in the `machines.LINUX` file. This script will prompt the user for a user password twice.

```
Enter your user password:
Re-enter your user password:
```

Each time the user enters the cluster password, asterisks will appear in lieu of the password text so as to provide security. Upon successful completion of the script, the following message fragment will appear:

```
                                        …
Node count = 4
Secure shell connectivity was established on all nodes.
                                        …
```

A log of the transactions for this script will be recorded in:

`/tmp/sshconnectivity.<login-name>.log`

where *<login-name>* is a meta-symbol for the user's actual login.

Back to Top

Installation

The Intel Cluster Toolkit installation process is comprised of six basic steps. The Intel Cluster Toolkit 2.0.1 package consists of the following components:

Software Component	Default Installation Directory
Intel MKL Cluster Edition 8.1	`/opt/intel/ict/2.0.1/cmkl/8.1`
Intel MPI Library 2.0.1	`/opt/intel/ict/2.0.1/mpi/2.0.1`
Intel MPI Benchmarks 2.3	`/opt/intel/ict/2.0.1/imb/2.3`
Intel Trace Analyzer 6.0.1	`/opt/intel/ict/2.0.1/ita/6.0.1`
Intel Trace Collector 6.0.1	`/opt/intel/ict/2.0.1/itc/6.0.1`

Note that the Intel Cluster Toolkit installer will automatically make the appropriate selection of binaries, scripts, and text files, from its installation archive based on the Intel processor architecture of the host system where the installation process is initiated. The user does not have to worry about selecting the correct software component names for the given Intel architecture.

Recall that the user of the Intel Cluster Toolkit may need assistance from their system administrator in installing the associated software packages on their cluster system, if the installation directory requires system administrative write privileges (e.g. `/opt/intel`). This assumes that the user's login account does not have administrative privileges.

Important Note: The 4.2.2 version of RPM on Red Hat Enterprise Linux 3.0 for Itanium 2 has a broken relocation feature. This will be a serious problem for users trying to install on clusters where there are shared devices. A recommended solution is for the user to upgrade to the latest release of RPM. A *possible* URL for retrieving a recent release of RPM that resolves this problem on the Itanium 2 architecture is:

http://www.redhat.com

1. A `machines.LINUX` file will either need to be created, or an existing `machines.LINUX` file can be used by the Intel Cluster Toolkit installer to deploy the appropriate software packages from the toolkit amongst the nodes of the cluster. This `machines.LINUX` file contains a list of the computing nodes (i.e. the hostnames) for the cluster. The format is one hostname per line:

   hostname

   The hostname should be the same as the result from the Linux* command "`hostname`". An example of the content for the file `machines.LINUX`, where a contrived cluster consists of eight nodes might be:

   ```
   clusternode1
   clusternode2
   clusternode3
   clusternode4
   clusternode5
   clusternode6
   clusternode7
   clusternode8
   ```

   It is always assumed that the first node in the list is the master node. The remaining nodes are the compute nodes. The text `clusternode1` and `clusternode2`, for example, represent the names of two of the nodes in a contrived computing cluster. The contents of the `machines.LINUX` file can also be used by users to construct an `mpd.hosts` file for the multi-purpose daemon (MPD) protocol. The MPD protocol is used for running MPI applications that utilize Intel MPI Library.

2. In preparation for doing the installation, the user may want to create a staging area. On the system where the Intel Cluster Toolkit software components are to be installed, it is recommended that a staging area be constructed in a directory such as `/tmp`. An example folder path staging area might be:

   `/tmp/ict_staging_area`

   where `ict_staging_area` is an acronym for Intel Cluster Toolkit staging area.

3. Upon registering for Intel Cluster Toolkit 2.0.1, you will receive a serial number (E.g., C111-12345678) for this product. Your serial number can be found within the email receipt of your product purchase. Go to the Intel Registration Center site and provide the product serial number information.

Once admission has been granted into the registration center, a user will be able to access the Intel® Premier web pages for software support.

4. The license for the Intel Cluster Toolkit license file that is provided to the user should be placed in a location pointed to by the `INTEL_LICENSE_FILE` environment variable. Do not change the file name as the "`.lic`" extension is critical. Common locations for the attached license file are:

   `<installation path>/licenses`

   For example, on the cluster system where the Intel Cluster Toolkit software is to be installed, all licenses for Intel-based software products might be placed in:

   `/opt/intel/licenses`

   It is also imperative that the user and/or the system administrator set the environment variable `INTEL_LICENSE_FILE` to the directory path where the Intel software licenses will reside *prior* to doing an installation of the Intel Cluster Toolkit. For Bourne Shell or Korn Shell the syntax for setting the `INTEL_LICENSE_FILE` environment variable might be:

   ```
   export INTEL_LICENSE_FILE=/opt/intel/licenses
   ```

   For C Shell, the syntax might be:

   ```
   setenv INTEL_LICENSE_FILE /opt/intel/licenses
   ```

5. Patrons can place the Intel Cluster Toolkit software package into the staging area folder.

6. The installer package for the Intel Cluster Toolkit has the following general nomenclature:

   `l_ict_<version>.<release>.tar.gz`

   where `<version>` is a string such as:

   `b_2.0.1`, where `b` is an acronym for beta

   or

   `p_2.0.1`, where `p` is an acronym for production

   The `<release>` meta-symbol is a string such as `007`. This string indicates the package number.

The command:

```
tar –xvzf l_ict_<version>.<release>.tar.gz
```

will create a subdirectory called `l_ict_<version>.<release>`. Change to that directory with the shell command:

```
cd l_ict_<version>.<release>
```

For example, suppose the installation package is called `l_ict_p_2.0.1.007.tar.gz`. In the staging area that has been created, type the command:

```
tar –xvzf l_ict_p_2.0.1.007.tar.gz
```

This will create a subdirectory called `l_ict_p_2.0.1.007`. Change to that directory with the shell command:

```
cd l_ict_p_2.0.1.007
```

As mentioned in the release notes section titled, "**Alternative Solution to Establishing Secure Shell Connectivity**", the `expect` shell script file called "`sshconnectivity.exp`" can be used to help the user establish secure shell connectivity on a cluster system. The syntax for the command is:

```
./sshconnectivity.exp machines.LINUX
```

This `expect` shell script will create or update a `~/.ssh` directory on each node of the cluster beginning with the master node which must be the first name listed in the `machines.LINUX` file. This script will prompt the user for a cluster password twice.

```
Enter your cluster password:
Re-enter your cluster password:
```

Each time the user enters the cluster password, asterisks will appear in lieu of the password text so as to provide security. Upon successful completion of the script, the following message fragment will appear:

```
…
Node count = 4
Secure shell connectivity was established on all nodes.
…
```

A log of the transactions for this script will be recorded in:

/tmp/sshconnectivity.*<login-name>*.log

where *<login-name>* is a meta-symbol for the user's actual login.

Once secure shell connectivity is established, type a variation of the `install` command as demonstrated by the table below and follow the prompts issued by this install script.

Installation command	Is root password required initially?	Installer prompts to be aware of	Default installation area
`./install`	Yes		`/opt/intel/ict/...`

`./install --nonroot`	No	`Installing Intel(R) Math Kernel Library for Linux* version p_... Software packages that are installed using RPMs are available system-wide. We recommend that you install the software using RPM (option 1). This would require root password.`
		
		`If you do not have root password, you can do a local installation in your home folder by choosing option 2 below.`
		
		`Which of the following would you like to do?`
		`1. Install the software using RPM (root password required) - Recommended.`
		`2. Install the software`
			`./intel/ict/...` in user's home directory

		without using RPM database (root password not required). x. Exit Please make a selection: For --nonroot and --nonrpm	
./install --nonrpm	Yes	Super-user or "root" privileges are required in order to continue. Please enter "root" password. Password:	/opt/intel/ict/…
./install --nonroot --nonrpm	No		./intel/ict/… in user's home directory

Note that Intel Trace Analyzer and Intel MPI Benchmarks are only installed on the master node.

By default, the global root directory for the installation of the Intel Cluster Toolkit is:

/opt/intel/ict/*<version#>.<minor-version#>*

where *<version#>* is an integer, and *<minor-version#>* is an integer. An example would be 2.0.1.

Within the folder path /opt/intel/ict/<version#>.*<minor-version#>* one will find the text files:

ictvars.csh

ictvars.sh

and

ictsupport.txt

If one is using Bourne Shell or Korn Shell for the login session, one should type:

. ./ictvars.sh

and for a login session that uses C Shell, one should type:

```
source ./ictvars.csh
```

The file called:

```
ictsupport.txt
```

contains the Package ID and Package Contents information. Please use the information in `ictsupport.txt` when submitting customer support requests.

For the default installation path, an index file, an FAQ file, and the user's guide are located in the directory path:

`/opt/intel/ict/<version#>.<minor-version#>/doc`

where as mentioned above, *<version#>* is an integer, and *<minor-version#>* is an integer. A complete default folder path to the documentation directory might be:

```
/opt/intel/ict/2.0.1/doc
```

The name of the index file is:

```
index.htm
```

The index file can be used to navigate to the FAQ, the release notes, the user's guide, and an internet accessible Intel Cluster Toolkit Tutorial.

The name of the FAQ file is:

```
ICT_FAQ_2.0.1.htm
```

The name of the user's guide file is:

```
ICT_Users_Guide_2.0.1.pdf
```

By default, the local version of the release notes is located in the directory path:

`/opt/intel/ict/<version#>.<minor-version#>/release_notes`

The name of the release notes file is:

Back to Top

Known Problems

1. On Itanium 2-based systems, the following symptom may occur for RPM version 4.2.2 when doing an install with the `--nonroot` option:

```
install --nonroot

the following symptom appears:

Installing Intel(R) MPI Library for Linux* version
p_2.0.1.022...
Extracting
files...    ##############################################
###   [100%]

Would you like to:

    Install:

    i.  Intel(R) MPI Library, Development Kit for Linux*
version 2.0.1-022

    x.  Exit

Your choice? (i/x) [i]: i

Cannot install Intel(R) MPI Library, Development Kit for
Linux* version 2.0.1-022 to
"/home/tooluser/intel/ict/2.0.1/mpi/2.0.1":
RPM version 4.2.2 prevent installation into a non-default
directory.
Installation failed.
WARNING: Intel(R) MPI Library for Linux* installer did
not create "/tmp/install.XX2svw1O/MPI.ini" file; the
product cannot be installed to all cluster nodes.
Press Enter to continue...
```

   The solution is to use an updated version of RPM.

2. For the Intel MPI Benchmarks, when running an instrumented version of `IMB-IO` that has been built with Intel MPI Library and the Intel Trace Collector library, the following runtime error message may occur:

```
                                ...
aborting job:
Fatal error in MPI_Barrier: Other MPI error, error
  stack:
```

```
MPI_Barrier(528): MPI_Barrier(MPI_COMM_WORLD) failed
MPIR_Barrier(75):
MPIC_Sendrecv(161):
MPIC_Wait(321):
MPIDI_CH3_Progress(206): handle_sock_op failed
MPIDI_CH3I_Progress_handle_sock_event(93):
connection_recv_fail(759):
MPIDU_Socki_handle_read(627): connection failure
  (set=0,sock=3,errno=104:Connection reset by peer)
aborting job:
Fatal error in MPI_Barrier: Other MPI error, error
  stack:
MPI_Barrier(528): MPI_Barrier(MPI_COMM_WORLD) failed
MPIR_Barrier(75):
MPIC_Sendrecv(161):
MPIC_Wait(321):
MPIDI_CH3_Progress(206): handle_sock_op failed
MPIDI_CH3I_Progress_handle_sock_event(93):
connection_recv_fail(759):
MPIDU_Socki_handle_read(603): connection closed by peer
  (set=0,sock=3)
rank 0 in job 379  fx-mercury01_34996   caused
  collective abort of all ranks
  exit status of rank 0: killed by signal 9
```

This problem seems to occur only when Intel Trace Collector and Intel® MPI Library are linked to the `IMB-IO` executable. We are presently working on resolving this issue in a future release of the Intel Cluster Toolkit.

Back to Top

Technical Support and Feedback

Submitting Issues

Your feedback is very important to us. To receive technical support and product updates for the software tools provided in this Intel Cluster Toolkit 2.0.1 release you need to register at the Intel Registration Center.

Note: If you are having trouble registering, or are unable to access your Premier Support account, contact technical support.

To submit an issue via the Intel Premier Support website, please perform the following steps:

1. Ensure that Java* and JavaScript* are enabled in your browser.
2. Go to https://premier.intel.com/.
3. Type in your Login and Password. Both are case-sensitive.
4. Click the "Submit Issues" button in the left margin.
5. Read the Confidentiality Statement and click the "I Accept" button.
6. Click on the "Go" button next to the "Product" drop-down list.
7. Click on the "Submit Issue" link in the left navigation bar.

8. Choose "Development Environment (tools, SDV, EAP)" from the "Product Type" drop-down list.
9. If this is a software or license-related issue choose "**Intel(R) Cluster Toolkit**" from the "Product Name" drop-down list.
10. Enter your question and complete the fields in the web-page windows that follow to successfully submit the issue.

Please follow these guidelines when forming your problem report or product suggestion:

1. Describe your difficulty or suggestion. For problem reports, please be as specific as possible (e.g., including compiler and link command line options), so that we may reproduce the problem. Please include a small test case if possible.

2. Describe your system configuration information. Be sure to include specific information that may be applicable to your setup: operating system, name and version number of installed applications, and anything else that may be relevant to helping us address your concern.

Back to Top

Copyright and Legal Information

The software described in this document may contain software defects which may cause the product to deviate from published specifications. Current characterized software defects are available on request.

Intel, the Intel logo, Intel SpeedStep, Intel NetBurst, Intel NetStructure, MMX, i386, i486, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Celeron, Intel Centrino, Intel Xeon, Intel XScale, Itanium, Pentium, Pentium II Xeon, Pentium III Xeon, Pentium M, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

* Other names and brands may be claimed as the property of others. All trademarks and registered trademarks referenced in this Intel Cluster Toolkit document are the property of their respective holders.

Back to Top