

CONTENTS

1. OVERVIEW
 2. NEW FEATURES
 3. SYSTEM REQUIREMENTS
 4. WHERE TO FIND THE RELEASE
 5. INSTALLATION NOTES
 6. DOCUMENTATION
 7. KNOWN LIMITATIONS AND TROUBLESHOOTING
 8. TECHNICAL SUPPORT
 9. DISCLAIMER AND LEGAL INFORMATION
-

1. OVERVIEW

Intel® Cluster Checker verifies the configuration and performance of Linux*-based clusters and checks the cluster's compliance with the Intel® Select Solutions for Simulation and Modeling.

1.1. RELATED PRODUCTS AND SERVICES

Information about Intel® oneAPI beta software development products is available at <http://www.intel.com/software/en-us/oneapi>

These are some of the products related to Intel® Cluster Checker:

- o The Intel® oneAPI Data Parallel C++ and Fortran Compilers include advanced optimization and multithreading capabilities, highly optimized performance libraries, and analysis tools for creating fast reliable multithreaded applications.

<http://www.intel.com/software/en-us/oneapi>

- o The Intel® oneAPI HPC Toolkit contains the MPI Library for Linux*, the Intel® Trace Analyzer and Collector for Linux*, and the Intel® Math Kernel Library Cluster Edition for Linux* are the most awarded development tools. They create, analyze, and optimize high-performance applications on clusters of Intel® processor-based systems.

<http://www.intel.com/en-us/oneapi/hpc-kit>

2. NEW FEATURES

2.1 WHAT'S NEW IN VERSION 2021.2 (2021.2.1)

- Improved execution of HPL and HPCG with mpirun launch command on workload

- managed systems (e.g. Slurm)
- Intel(R) Ethernet 800 series support (intel_ethernet800_* frameworks)
 - driver checks and version uniformity
 - required libraries
 - memlock unlimited check
 - PCIe checks
- setvars scripts compatible with mksh environments

2.2 WHAT'S NEW IN VERSION 2021.1 (2021.1.1)

Intel(R) Cluster Checker Version 2021.1 represents several significant improvements from the last major revision of Intel(R) Cluster Checker Version 2019. Many of the modifications and improvements listed below have been introduced in subsequent updates to Intel(R) Cluster Checker Version 2019 over the past year, and included in beta releases for Version 2021, the following is a compilation of all the significant changes, organized by category, not by update or beta release.

What's New Section Contents

- 2.2.1 Improvements to Output Formatting, Error Reports and JSON Support
- 2.2.2 New Support for OSU, MPICH, and MPI Workloads and Analysis
- 2.2.3 New and Modified Framework Definitions
- 2.2.4 New Feature for Node Group Analysis and Reporting
- 2.2.5 Support for Linux Environment Modules
- 2.2.6 Improved Support for ksh and zsh shells
- 2.2.7 Propagation of Environment Variables
- 2.2.8 Changes to Default Temporary File Permissions
- 2.2.9 Update to checks on \$TMPDIR
- 2.2.10 Updates of 3rd Party Components to Correct CVEs

2.2.1 Improvements to Output Formatting, Error Reports and JSON Support

Intel(R) Cluster Checker has greatly improved its output formatting and error reporting. These changes have been introduced over the past year in the Update packages, and during the beta release for oneAPI. Intel(R) Cluster Checker has adopted a new summarized report upon completion, with details from the analysis available in the `clck_report.log` (or designated `-o <filename>`), and details about the execution environment for Intel(R) Cluster Checker in the `clck_execution_warnings.log` file.

2.2.2 New Support for OSU and Intel(R) MPI workloads and analysis

Intel(R) Cluster Checker now provides a wide array of workloads and test suites available for the evaluation and continued performance analysis of the compute environment. Support is now extended to the execution of Ohio State University (OSU) workloads. While the specific OSU workloads are not provided as part of the standard Intel(R) Cluster Checker installation, the OSU workloads can be installed separately and used by Intel(R) Cluster Checker for the evaluation and analysis of the environment. Many of the OSU workloads support the MPICH

messaging protocols.

2.2.3 New and modified framework definitions

Several new framework definitions have been added to Intel(R) Cluster Checker to aid in the validation and analysis of clustered environments. While others have been cleaned up or modified to enable a cleaner and better understanding of the frameworks.

These new and modified framework definitions include the following:

```
health_base
health_user
health_exteded_user
health_admin
health_extended_admin
osu_*
```

To get a full listing of the available framework definitions execute the following on the command line:

```
click -X list
```

For details about a specific framework definition either refer to the reference section of the User's Guide or perform the following on the command line:

```
click -X <framework definition name>
example: click -X health_base
```

2.2.4 New Feature for Node Group Analysis and Reporting

Intel(R) Cluster Checker has introduced the ability to analyze discrete groups of compute nodes. The new feature utilizes a command line option, `-g`, and an xml file designating the grouping of the compute nodes. The grouping could be based upon physical characteristics of the nodes, location, or other features. A compute node can belong in multiple node groups at the same time. The new feature only applies to the analysis of the nodes. Collection of the data for the specific framework definitions is across all designated nodes. Please see the User's Guide for more details on implementing this feature, the xml file setup and example, and interpretation of the analysis results.

2.2.5 Support for Linux Environment Modules

Intel(R) Cluster Checker now supports the use of Linux environment modules for setting up the proper runtime environment. The environment modules support provide an alternative to the current setup requirements found in the `clickvars.[sh|csh]` script.

Please see the User's Guide for additional details and setup requirements.

2.2.6 Improved Support for ksh and zsh shells

Intel(R) Cluster Checker has expanded the number of shell environments that can be utilized for the collection and analysis of clustered systems by supporting ksh (korn shell) and zsh (z-shell). Specific details, setup, and exemptions are included in the User's Guide for Intel(R) Cluster Checker.

2.2.7 Propagation of Environment Variables

Intel(R) Cluster Checker has now added environment variable propagation with its default pdsh-based remote execution on compute nodes. This enables proper operation of data collection on compute nodes within consistent environment settings. A new command line option '-e' was added to turn off the environment variable propagation for backward compatibility.

2.2.8 Changes to Default Temporary File Permissions

Intel(R) Cluster Checker has modified its default temporary file permissions to remove the "world-writable" flag on all files and the temporary directory used by Intel(R) Cluster Checker to store temporary files before analysis. This change is to provide better security to users and ensure only those authorized to access Intel(R) Cluster Checker results are allowed to do so. Intel(R) Cluster Checker still allows users and groups authorized to execute the tools to read and write files, but limits access by others.

2.2.9 Update to checks on \$TMPDIR

Intel(R) Cluster Checker has modified the check for file permissions to not require any 'world' permissions for \$TMPDIR.

2.2.10 Updates of 3rd Party Components to Correct CVEs

Intel(R) Cluster Checker leverages several third-party components to assist in the collection, analysis, and generation of reports. To maintain security commitments to Intel(R) and our customers, Intel(R) Cluster Checker is regularly scanned using industry standard tools for exposure of Common Vulnerable Exploits in the third-party components. It is important for customers using Intel(R) Cluster Checker to maintain their implementations at the latest level to not only receive the latest improvements to the collection and analysis, but also to receive the most recent updates to these third-party components.

2.2 OLDER VERSIONS

- The release notes for older, versions of Cluster Checker can be found at:
<https://software.intel.com/en-us/articles/intel-cluster-checker-release-notes-and-new-features>

3. SYSTEM REQUIREMENTS

The following sections describe hardware and software requirements.

3.1. HARDWARE

- Intel® Xeon® processor (Intel® 64 architecture)
- 1 GB of RAM recommended
- 160 MB of free hard disk space required for installation

3.2. SOFTWARE

Operating Systems:

- CentOS 7
- CentOS 8
- Red Hat* Enterprise Linux* 7
- Red Hat* Enterprise Linux* 8
- SUSE* Linux* Enterprise Server 12
- SUSE* Linux* Enterprise Server 15

Runtimes:

- Intel® MPI Library

Note: While the full SDK versions of these components fulfill the requirement, only the runtime library is required.

4. WHERE TO FIND THE RELEASE

Intel® Cluster Checker can be installed with Intel® oneAPI HPC Toolkit.

See the Installation section of the User Guide or the Installation documentation for the Intel® oneAPI HPC Toolkit for more information.

5. INSTALLATION NOTES

The default Intel® Cluster Checker install path is:
/opt/intel/oneapi/clck/2021.2.1

Notes:

- Intel® Cluster Checker needs to be installed on all nodes. This can either be accomplished either by installing into a shared directory or by installing a local copy on each node.
 - To install a local copy on each node, repeat the package installation for each node.
-

6. DOCUMENTATION

This release of Intel® Cluster Checker includes the following documentation:

The Getting Started Guide walks through using Intel® Cluster Checker for the first time.

The Intel® Cluster Checker User's Guide contains information about how to use, configure, and extend Intel® Cluster Checker. The User's Guide describes the basic usage models, contains information about specific configuration options, explains how to embed Intel® Cluster Checker functionality into other applications, shows how to add new checks to the tool, and demonstrates how to modify existing checks.

The Intel® Cluster Checker API reference describes the API that may be used to embed Intel® Cluster Checker functionality into other software programs.

The documentation can be found at:

<https://software.intel.com/en-us/intel-cluster-checker-support/documentation>.

7. KNOWN LIMITATIONS AND TROUBLESHOOTING

The following is a list of known issues in this release.

- Data collection behavior and functionality
 - o Intel® MPI Benchmarks (imb_*) providers will output false positives when run with MPICH.
 - o When executing data collection as root, the following framework definitions can just hang with no message, but maybe terminated by a single ctrl-c or terminate by a provider timeout:
 - mpi_multinode_functionality
 - hpl_cluster_performance
 - imb_* (Intel® MPI Benchmarks)
 - osu_* (OSU Micro-Benchmarks)
 - o Currently Cluster Checker will not collect data for the HPCG benchmarks correctly if Intel® MPI and MPICH (www.mpich.org) are both installed on the environment being tested. The test environment will look for and execute the Intel® MPI optimized binary for HPCG and thus reset the environment variables for MPICH. Discovery and handling of this limitation will be corrected in future versions of Intel® Cluster Checker.
 - o imb_pingpong_fabric_performance framework definition when launched with an odd number of nodes through Slurm, with MPI as the collector mechanism (mpi.so), will report no-data for the last server assigned to the slurm job. Workaround involves using a nodefile to specifically test the last server where 'no-data' was reported with another server in the infrastructure.

- o The compute node hostname identified in the nodefile must match the hostname reported by either the `uname` or `hostname` utility on the compute node itself. Deviations in the hostnames, or use of fully qualified domain names in either the nodefile or the compute node, may impact or produce inaccurate uniformity percentages and counts and be reported as a failure or warning by Cluster Checker.
- o Please note that for execution of HPCG benchmarks (such as in the checks `hpcg_single` and `hpcg_cluster`) on non-standard install path for the Intel® MPI Library and Intel® Math Kernel Library (Intel® MKL) runtime, libraries must be installed and be exported in the `LD_LIBRARY_PATH` on the system.
- o Use of the latest runtime libraries for Intel® MPI Library and Intel® Math Kernel Library is required to ensure compatibility with Intel® Cluster Checker.
- o If the temporary directory used during collection is located on a shared file system, the directory will not be deleted.
- o Databases located on NFS file systems mounted with the "nolock" option are not supported. Not all data from concurrent data collection instances per database will be written to the database and the database may become corrupted. A single data collector instance per database can usually be used successfully in this case.
- o The error "Error: disk I/O error" may be generated when accessing a database located on a Lustre file system. The Lustre file system must be mounted with the "-o flock" option.
- o If collecting data as root, the value of the `CLCK_SHARED_TEMP_DIR` environment variable must be set to the fully-qualified path of a directory accessible on all nodes.
- o Intel® Cluster Checker uses the command `ldconfig -p` as well as the environment variable `LD_LIBRARY_PATH` to detect the presence of required libraries. In order for Intel® Cluster Checker to detect required libraries, they must be present in the `LD_LIBRARY_PATH` or the result of `ldconfig -p`. (Applies to the Framework Definitions `second-gen-xeon-sp_user`, `second-gen-xeon-sp_priv`, `intel_hpc_platform_compat-hpc-2018.0`, `intel_hpc_platform_sdvis-core-2018.0`, and `intel_hpc_platform_second-gen-xeon-sp-2019.0`)
- o In order for Intel® Cluster Checker to detect the Intel® Distribution for Python*, it must be in the user's `PATH`. (Applies to the Framework Definitions `second-gen-xeon-sp_user`, `second-gen-xeon-sp_priv`, `intel_hpc_platform_compat-hpc-2018.0`, and `intel_hpc_platform_second-gen-xeon-sp-2019.0`)
- o If Intel® Parallel Studio is sourced before the Intel® Distribution for Python* in the user's environment, Intel® Cluster Checker is unable to detect all the required libraries for Intel® MPI Library. (Applies

to the Framework Definitions `second-gen-xeon-sp_user`,
`second-gen-xeon-sp_priv`, `intel_hpc_platform_compat-hpc-2018.0`, and
`intel_hpc_platform_second-gen-xeon-sp-2019.0`)

- o The detected version of Intel® MPI Library is used to determine whether Intel® Cluster Checker checks for Intel® Parallel Studio 2018 or 2019. If the Intel® MPI Library version does not match the version of the rest of Intel® Parallel Studio, the wrong set of libraries will be checked. (Applies to the Framework Definition `intel_hpc_platform_compat-hpc-2018.0`)
- o Intel® Cluster Checker can only detect the version of the Intel® Fortran Compiler version with Intel® Parallel Studio 2017 or later. (Applies to the Framework Definitions `second-gen-xeon-sp_user`, `second-gen-xeon-sp_priv`, `intel_hpc_platform_compat-hpc-2018.0`, and `intel_hpc_platform_second-gen-xeon-sp-2019.0`)
- o Intel(R) TrueScale InfiniBand(TM) does not have driver support for Operating Systems supported by oneAPI. As a result collection of TrueScale data is not supported with Intel(R) Cluster Checker.
- o In addition, there are limitations to validating Intel® Select Solutions compliance when running on Ubuntu. It is not recommended to use Intel® Cluster Checker for Intel® Select Solutions compliance when running on Ubuntu.

- Analysis behavior and functionality

- o Clusters containing dual port InfiniBand* adapters where the second port is unused should suppress the 'infiniband-port-physical-state-not-linkup' and 'infiniband-port-state-not-active' signs. See Chapter 4 of the User's Guide for more information on how to suppress signs.
- o When using the Linux* boot parameter `isolcpus` with an Intel® 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 and you are using the Intel® MPI Library, you can try setting `I_MPI_PIN` to off. Refer to the Intel® Cluster Checker reference manual for details on specifying environment variables for tests.
- o When run with `dgemm/dgemm_cpu_performance` or `stream/stream_memory_bandwidth_performance` framework, "stream-outlier" or "dgemm-data-is-substandard" may be observed as the corresponding provider scripts may not yield the expected performance with SNC-2/SNC-4 cluster mode and Flat memory mode configurations for Intel® Xeon Phi(TM) processor. There may be an issue with the kernel itself (BZ#1479763), documented at <https://access.redhat.com/errata/RHBA-2017:2581>. If there are no corresponding diagnoses, the signs may be suppressed.
- o The sign `paraview-missing` fires despite ParaView* being present on the system. (Applies to the Framework Definition `intel_hpc_platform_sdvis-cluster-2018.0`)

- Installation Limitation

- o 2021.2 release is conflicted with 2021.1 release if it was installed by YUM/DNF/Zypper package manager. Please remove 2021.1 release before installing 2021.2 release. To install both releases side-by-side, please use web installer tool.

8. TECHNICAL SUPPORT

Intel® Cluster Checker is part of the Intel® oneAPI HPC toolkit. Information on priority support for Intel® oneAPI Base and HPC Toolkits is available here: <https://software.intel.com/en-us/oneapi/support>

Community forum based support is available through the forums: <https://community.intel.com/t5/Intel-oneAPI-HPC-Toolkit/bd-p/oneapi-hpc-toolkit>

For information about how to find Technical Support, Product Updates, User Forums, FAQs, tips and tricks, and other support information, please visit: <https://software.intel.com/en-us/oneapi/support>

9. 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 in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

© 2021 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