LEVERAGING SYSTEM DIAGNOSTICS IN AN APPLICATION WITH INTEL® CLUSTER CHECKER

Becca Paren
Cluster Systems Engineer
Software and Services Group

November 2017
Agenda

- Intel® Cluster Checker Introduction
- Basic Commands Hands-on
- API Overview
- Using the API Hands-on
- Extensibility and Personalization
INTEL® CLUSTER CHECKER INTRODUCTION
Clusters are complex systems!

Challenge is to reduce this complexity barrier for:

- Application developers
- Cluster architects
- Cluster users
- System administrators
Clusters are complex systems!

Challenge is to reduce this complexity barrier for:

- Application developers
- Cluster architects
- Cluster users
- System administrators
Intel® Cluster Checker
for Linux* clusters

An **expert system** approach that provides cluster systems expertise in a tool

- Verifies system health
- Offers suggested actions
- Provides extensible framework
- API for integrated support
Overview

Delivers pre-packed cluster systems expertise as a diagnostic tool

✔ Suitable for HPC experts and those new to HPC

**Intel® Cluster Checker 2018 Features**

✔ Checks cluster functionality, uniformity, and performance
  ✔ Standard performance tests (e.g. DGEMM, HPL, Intel® MPI Benchmarks, IOzone, STREAM)
  ✔ Hardware and software uniformity
  ✔ Consistency for certain kernel and BIOS settings

✔ Ability to embed, extend, and customize the checking capability (API, SDK)

✔ Supports:
  ✔ Intel® Xeon® and Intel® Xeon Phi™ processor families
  ✔ Intel® Omni-Path Fabrics, Intel® True Scale Fabric, Ethernet*, InfiniBand*
  ✔ Red Hat* Enterprise Linux* 6 and 7, SUSE* Linux Enterprise Server* 11 and 12, Ubuntu* 16.04 and 17.04
  ✔ Lustre* filesystem
Expert System Based Design
Modelled on Clinical Decision Support Systems

Key Concepts:

- **Symptoms** are subjective indications of health
- **Signs** are objective indications of health detected by direct observation
- **Diagnoses** are the identification of the root cause of an issue
- **Remedies** are methods to resolve an issue
Key Concepts:

- **Symptoms** are subjective indications of health
- **Signs** are objective indications of health detected by direct observation
- **Diagnoses** are the identification of the root cause of an issue
- **Remedies** are methods to resolve an issue

<table>
<thead>
<tr>
<th>Concept</th>
<th>Human</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symptom</td>
<td>Difficulty walking; ankle hurts</td>
<td>Job is running slow</td>
</tr>
</tbody>
</table>
| Signs   | • Range of ankle motion limited to 50% of normal  
          • Ankle severely inflamed compared to non-injured leg  
          • X-ray negative for fracture | • DGEMM performance is 50% of peak  
          • Background process is using 100% of the processor |
| Diagnosis| Sprained ankle | Background process is stealing cycles |
| Remedy  | Ice ankle & keep it elevated, take 500 mg of ibuprofen every 4-6 hours | Kill the background process to reduce the load of the system |
Basic Use Case

% clck-collect -f nodefile -F health...

Collects Diagnostic Data

% clck-analyze -f nodefile -F health ...

Analyzes and Applies Rules

Suggests Remedies

Intel® Cluster Checker 2018
Mode: health
Database: SQLite file /home/clck/.clck/2018.0/clck.db.

38 checks requested: all_to_all, cpu, datconf, dgemm, environment, ethernet, heartbeat, hpl, icr_cluster, icr_version, imb_pingpong, infiniba nd, intel_c

Reading the database for the following checks:
  all_to_all... done (0.0264 seconds)
  cpu... done (0.0176 seconds)
Framework Definitions (FWD)

Easy grouping of collection and analyze components for, consisting of

- Data providers (used to collect data on the system)
- Extension modules (used to transform data from database to objects)
- Knowledge base modules (rules that reason about the data from the database)
- Optional details on path for all of the above, and to include other framework definitions

Allows execution of Intel® Cluster Checker with defined scope (‘-F’)

- Default analysis executes all components listed in ‘health.xml’ framework definition
- Pre-defined framework definitions (e.g. ‘benchmarks.xml’)
- Custom framework definitions
Pre-defined Framework Definitions

All framework definitions located at: opt/intel/clck/<version>/etc/fwd/

Some General Definitions for HPC Clusters (more located in above path):

- basic_internode_connectivity.xml
- benchmarks.xml
- hardware.xml
- health.xml
- imb_pingpong.xml
- infiniband.xml
- local_disk_storage.xml
- network_time_uniformity.xml
- opa.xml

Definitions for Intel® Scalable System Framework (SSF) Compliance:

- compat-base.xml
- compat-hpc.xml
- core.xml
- hpc-cluster.xml
Diagnoses may include:

- **Remedy** – suggested resolution for a diagnosis

Diagnoses and Signs each have:

- **Confidence** – level of certainty (0-100%)
- **Severity** – level of seriousness (0-100%)
Intel® Cluster Checker 2018 provides diagnoses

1 diagnosis:

1. The hardware and/or firmware are not fully uniform for nodes in the same grouping.
   [ Id: non-uniform-hardware ]
   [ Severity: 20%; Confidence: 75% ]
   [ Node: node7 ]

1 diagnosed sign:

1. The amount of physical memory, 41060648 KB, is not within the range of 32803372 KB +/- 262144 KB defined by nodes in the same grouping.
   [ Id: memory-not-uniform ]
   [ Severity: 20%; Confidence: 75% ]
   [ Node: node7 ]

0 undiagnosed signs
BASIC COMMANDS HANDS-ON
Basic Use Case

We will work with the following files and commands interactively:

nodefile
basic_internode_connectivity.xml
click-collect
click-analyze
clickdb

*commands in readme-click-basic.txt in home directory*
Intel® Cluster Checker 2018 – API

*The Cluster Checker API is in C++

API allows integration into other software

- An application can check its cluster environment, health, functionality
- A monitoring system can display status information of cluster checking
- A deployment system can configure or trigger data collection/analysis
- A resource manager can control background execution of cluster checking
- A job scheduler can validate node groups before applications run
Intel® Cluster Checker 2018 – API

*The Cluster Checker API is in C++

API allows integration into other software

- An application can ... check its cluster environment, health, functionality
- A monitoring system can ... display status information of cluster checking
- A deployment system can ... configure or trigger data collection/analysis
- A resource manager can ... control background execution of cluster checking
- A job scheduler can ... validate node groups before applications run

Easy packaging with other software

- When compiled and linked using the API, Intel Cluster Checker functionality can be simply included and redistributed (no per copy royalty fee; not for evaluation version; terms in EULA and ‘redist.txt’ file)
Intel® Cluster Checker API Use Case - Application

- Integrate Intel® Cluster Checker into application(s) and distribute to customers
  - Installation of application: Check the HW-/SW-environment
  - Support: Remote check of customer system
  - Start of application: Check system health
  - Maintenance: Allow customer to maintain cluster health
Cluster Checker allows personalization by

- Using existing information or creation of a custom data provider
- Creating custom analysis

Examples can be found here:

API example: Embedded analysis in MyApp

Intel® Cluster Checker 2018 includes an API to embed analysis capabilities

MyApp

Hello World

Intel® Cluster Checker 2018

API

collect()
analyze()
get_faults()

Nodes can’t all talk to eachother!

MyApp output:

“Dear User, not all nodes in the system are accessible from all other nodes, so MyApp will not function correctly.”
## API Basic Use

* API is in C++

<table>
<thead>
<tr>
<th>Use</th>
<th>Command Line</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>clck.xml</td>
<td>config</td>
</tr>
<tr>
<td>Collection</td>
<td>clck-collect</td>
<td>collect()</td>
</tr>
<tr>
<td>Analysis Run</td>
<td>clck-analyze</td>
<td>analyze()</td>
</tr>
<tr>
<td>Analysis Output</td>
<td>clck-analyze</td>
<td>get_faults()</td>
</tr>
</tbody>
</table>
USING THE API HANDS-ON
myApp Example

myApp embeds Intel® Cluster Checker into a Message Passing Interface (MPI) program in C++

Files looking at today:

- clck.cpp
  - Uses Cluster Checker API to collect and analyze data
  - Looks at Process, CPU, Ethernet framework definitions

- mpi.cpp
  - Runs helloworld through mpi if the cluster is healthy, calls the cluster checker function created in clck.cpp
myApp Example

myApp embeds Intel® Cluster Checker into a Message Passing Interface (MPI) program in C++

Make the file:
- make

Run myApp:
- mpirun -f /home/test/nodefile -ppn 1 /home/test/shared_dir/mpi_sample/out/mpi-cpp
EXTENSIBILITY AND PERSONALIZATION
Cluster Checker allows personalization by

- Using existing information or creation of a custom data provider
- Creating custom analysis

Examples can be found here:

How Do We Adapt Cluster Checker to Specific Environments?

Want:

- **Customized analysis**: Users can specify any set of collection or analyze components to run, and can also create new ones

- **Central location for modifications**: Users should not have to edit multiple configuration files, they can put all personalization in one location

Don’t Want:

- **To modify root installation of the tool**: A user should not need root access to extend the tool
Solution - Framework Definitions (FWD)

Easy grouping of collection and analyze components for, consisting of:

- Data providers (used to collect data on the system)
- Extension modules (used to transform data from database to objects)
- Knowledge base modules (rules that reason about the data from the database)
- Optional details on path for all of the above, and to include other framework definitions

Allows execution of Intel® Cluster Checker with defined scope ('-F')

- Default analysis executes all components listed in 'health.xml' framework definition
- Pre-defined framework definitions (e.g. 'benchmarks.xml')
- Custom framework definitions
Framework Definition - Basic Example
‘benchmarks.xml’ – includes other framework definitions

```xml
<configuration>
  <framework_definition name="benchmarks">
    <includes>
      <include>fwd/dgemm_cpu_performance.xml</include>
      <include>fwd/ethernet.xml</include>
      <include>fwd/hpl_cluster_performance.xml</include>
      <include>fwd/imb_pingpong_fabric_performance.xml</include>
      <include>fwd/iozone_disk_bandwidth_performance.xml</include>
      <include>fwd/sgemm_cpu_performance.xml</include>
      <include>fwd/stream_memory_bandwidth_performance.xml</include>
    </includes>
  </framework_definition>
</configuration>

% clck-analyze –f nodefile –F benchmarks
```
Basic Use Case – Command Line

2017 old way: Specify checks.


2018 new way: Specify a path for the Framework Definition (not just a name of the file)

% clck-collect –f nodefile –F benchmarks
% clck-analyze –f nodefile –F benchmarks
Pre-defined Framework Definitions

All framework definitions located at: opt/intel/clck/<version>/etc/fwd/

Some General Definitions for HPC Clusters (more located in above path):

- basic_internode_connectivity.xml
- benchmarks.xml
- hardware.xml
- health.xml
- imb_pingpong.xml
- infiniband.xml
- local_disk_storage.xml
- network_time_uniformity.xml
- opa.xml

Definitions for Intel® SSF Compliance:

- compat-base.xml
- compat-hpc.xml
- core.xml
- hpc-cluster.xml
Framework Definition - myApp

Customized Framework Definition ‘myApp.xml’

<configuration>
  <framework_definition name="myApp">
    <includes>
      <mod>fwd/cpu.xml</mod>
      <mod>fwd/tcl.xml</mod>
      <mod>fwd/mpi_multimode_functionality.xml</mod>
      <mod>fwd/node_process_status.xml</mod>
    </includes>
  </framework_definition>
</configuration>
Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS”. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Copyright © 2017, Intel Corporation. All rights reserved. Intel, Pentium, Xeon, Xeon Phi, Core, VTune, Cilk, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

**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
User Testimonials at a Glance - Intel® Cluster Checker with Intel® Parallel Studio XE Cluster Edition

“Altair Engineering provides with HyperWorks and PBS Works best-in-class solutions for CAE modeling and simulation as well as workload management. For the development of our leading solvers like RADIOSS, AcuSolve, OptiStruct, FEKO and PBS Works we rely on the Intel® Parallel Studio XE Cluster Edition components for compilation, build and optimization of these high-performing solutions. With the latest Intel® Cluster Checker we are able to enhance our software products with additional functionality for validation and diagnosis of cluster configurations. Running our applications on a validated cluster system will help our customer to save time and costs during the solution of their engineering challenges.” said Eric Lequiniou, Director High Performance Computing, Altair Engineering Inc.

“Here at COMSOL we develop COMSOL Multiphysics - the multiphysics simulation platform that can model and simulate any physics-based system. To provide our users and customers with a high quality tool we use the Intel® Parallel Studio XE Cluster Edition components to compile, build and optimize our high performing application. With the new Intel® Cluster Checker we are able to provide new functionality to our customers which will enable them to validate their cluster configuration or to diagnose potential issues. Through Intel® Cluster Checker our customers are able to save time by taking full advantage of COMSOL Multiphysics performance on their cluster systems.” said Winfried Geis, Branch Manager, COMSOL Multiphysics GmbH.

“Installing Intel® Cluster Checker with the Intel® Parallel Studio XE Cluster Edition, I found the setup, configuration, ease of use, and the depth of the checks to be truly excellent. Its first-rate diagnostics facilitated understanding complex cluster problems, to make clusters at STFC working and performing to its optimum. A great asset for any big cluster.” said Ian McKenna, System Admin, Science and Technology Facilities Council (STFC, UK).
Framework Definition Components

```xml
<configuration>
  <framework_definition name="myFramework">

  <!-- includes -->
  <includes></includes> -->

  <providers>
    <provider>myProvider.clp</provider>
  </providers>

  <extension_mods>
    <mod>myCheck1</mod>
  </extension_mods>

  <kb.mods>
    <mod>myKB_Module.clp</mod>
  </kb.mods>

  <!-- provider_path -->
  <provider_path></provider_path> -->

  <!-- extension_path -->
  <extension_path></extension_path> -->

  <!-- kb_path -->
  <kb_path></kb_path> -->

  <provider_path></provider_path>
  <extension_path></extension_path>
  <kb_path></kb_path>

  Can include other FWDs
  Data Providers
  Connector Extension Modules
  Knowledge Base Modules
  Data Providers Path
  Connector Extensions Path
  Knowledge Base and Message Catalog Path (.xmc file)

</framework_definition>
</configuration>
```
Intel® Cluster Checker 2018 – what’s new

Ensure Your HPC Cluster Components Work Together

New features continue to improve usability and checking capabilities:

- Adding support for new Intel silicon and platform elements (processors, fabric, memory, storage, cluster provisioning, HPC platforms)
- Introducing simplified grouping of checks for extensibility
- Improved diagnostic output
- Validation of Intel® Scalable System Framework Classic HPC Cluster Reference Architectures
- In-depth Intel® Omni-Path Fabric checking
- Analysis from multiple database sources
- General improvements, bug fixes
Where to get Intel® Cluster Checker 2018

Software
- Install with Intel® Parallel Studio XE 2018 Cluster Edition for Linux*, free 30-day evaluation available

Getting started

Support
- http://supporttickets.intel.com– Online Service Center, a software support portal

Further information
Intel® Cluster Checker 2018

Encounter cluster performance issues and know its results

10 undiagnosed signs:

1. The STREAM benchmark result of 26.707 GB/s is outside the statistical range of 128.305 GB/s +/- 21.078 GB/s defined by nodes in the same grouping.
   [ Id: stream-outlier ]
   [ Severity: 60%; Confidence: 75% ]
   [ Node: node8 ]

2. The STREAM benchmark result of 26.298 GB/s is outside the statistical range of 128.305 GB/s +/- 21.078 GB/s defined by nodes in the same grouping.
   [ Id: stream-outlier ]
   [ Severity: 60%; Confidence: 75% ]
   [ Node: node7 ]
Intel® Cluster Checker 2018

Encounter cluster performance issues and know its results

1. The STREAM benchmark result of 26.707 GB/s is outside the statistical range of 128.305 GB/s +/- 21.078 GB/s defined by nodes in the same grouping.
   [ Id: stream-outlier ]
   [ Severity: 60%; Confidence: 75% ]
   [ Node: node8 ]

2. The STREAM benchmark result of 26.298 GB/s is outside the statistical range of 128.305 GB/s +/- 21.078 GB/s defined by nodes in the same grouping.
   [ Id: stream-outlier ]
   [ Severity: 60%; Confidence: 75% ]
   [ Node: node7 ]

3-8. The memory mode, 'Cache', is not uniform. 75% of nodes in the same grouping have the same mode.
   [ Id: xp-memory-mode-not-uniform:Cache ]
   [ Severity: 40%; Confidence: 25% ]
   [ 6 Nodes: node[1-6] ]
   [ Remedy: Please change the desired "Memory Mode" in the BIOS. ]

9-10. The memory mode, 'Flat', is not uniform. 25% of nodes in the same grouping have the same mode.
   [ Id: xp-memory-mode-not-uniform:Flat ]
   [ Severity: 40%; Confidence: 75% ]
   [ 2 Nodes: node[7-8] ]
   [ Remedy: Please change the desired "Memory Mode" in the BIOS. ]
Intel® Cluster Checker 2018

Encounter cluster performance issues and know its results

1. The STREAM benchmark result of 26.707 GB/s is outside the statistical range of 128.305 GB/s +/- 21.078 GB/s defined by nodes in the same grouping.
   [ Id: stream-outlier ]
   [ Severity: 60%; Confidence: 75% ]
   **Node: node8**

2. The STREAM benchmark result of 26.298 GB/s is outside the statistical range of 128.305 GB/s +/- 21.078 GB/s defined by nodes in the same grouping.
   [ Id: stream-outlier ]
   [ Severity: 60%; Confidence: 75% ]
   **Node: node7**

3-8. The memory mode, 'Cache', is not uniform. 75% of nodes in the same grouping have the same mode.
   [ Id: xp-memory-mode-not-uniform:Cache ]
   [ Severity: 40%; Confidence: 25% ]
   [ 6 Nodes: node[1-6] ]
   [ Remedy: Please change the desired "Memory Mode" in the BIOS. ]

9-10. The memory mode, 'Flat', is not uniform. 25% of nodes in the same grouping have the same mode.
   [ Id: xp-memory-mode-not-uniform:Flat ]
   [ Severity: 40%; Confidence: 75% ]
   [ 2 Nodes: node[7-8] ]
   [ Remedy: Please change the desired "Memory Mode" in the BIOS. ]
Everything fine - group of nodes is validated OK, ready to run applications

Python... done (0.00293 seconds)
RPM... done (1.86 seconds)
Shells... done (0.00434 seconds)
Storage... done (0.0249 seconds) Stream... done (0.0106 seconds)
TCL... done (0.0043 seconds)
x11_tools... done (0.00225 seconds)

Analyzing: 100%, rules completed/remaining: 7/0

Nodes being tested:
bdwa[1-4].imu.intel.com, imu-portal.imu.intel.com

0 diagnoses

0 diagnosed signs

0 undiagnosed signs

This analysis took 4.66548 seconds.

PASS: All checks passed.
Intel® Cluster Checker 2018

Cluster systems expertise in a box:

- On-demand and background monitoring modes for distributed cluster tests
- Built-in rule based expert system technology to analyze multifaceted issues
- Built-in knowledge base facilitates remedies for common issues
- Built-in database with threshold data for major range of components
- Automated checking throughout cluster life cycle
- API to integrate in other software, SDK to extend tests

‘2018’ version supports:

- Latest Intel® Xeon® and Intel® Xeon Phi™ processors
- Intel® Omni-Path Fabric, Intel® True Scale Fabric, Ethernet*, InfiniBand*
- Lustre* filesystem
- Installs with Intel® Parallel Studio XE 2018 Cluster Edition for Linux*