IBM HPC Cluster Health Check

IBM Redbooks Solution Guide

The IBM® Cluster Health Check (CHC) toolset is an extensible framework and collection of tools to check and verify the health of an IBM Cluster. The CHC framework provides an environment for integrating and running a subset of individual checks that can be performed on a single node or a group of nodes. The tests are categorized into different tool types and groups so that only checks for a specific part of the cluster are performed. Because the framework and toolset is extensible, users can also create their own groups and tools for checking different components of the cluster. Figure 1 shows the different cluster components and the verification pyramid. Each level of verification depends on the correctness of the underlying stages. The higher in the pyramid you are before a problem is found, the more difficult it might be to attribute it to a more foundational problem.

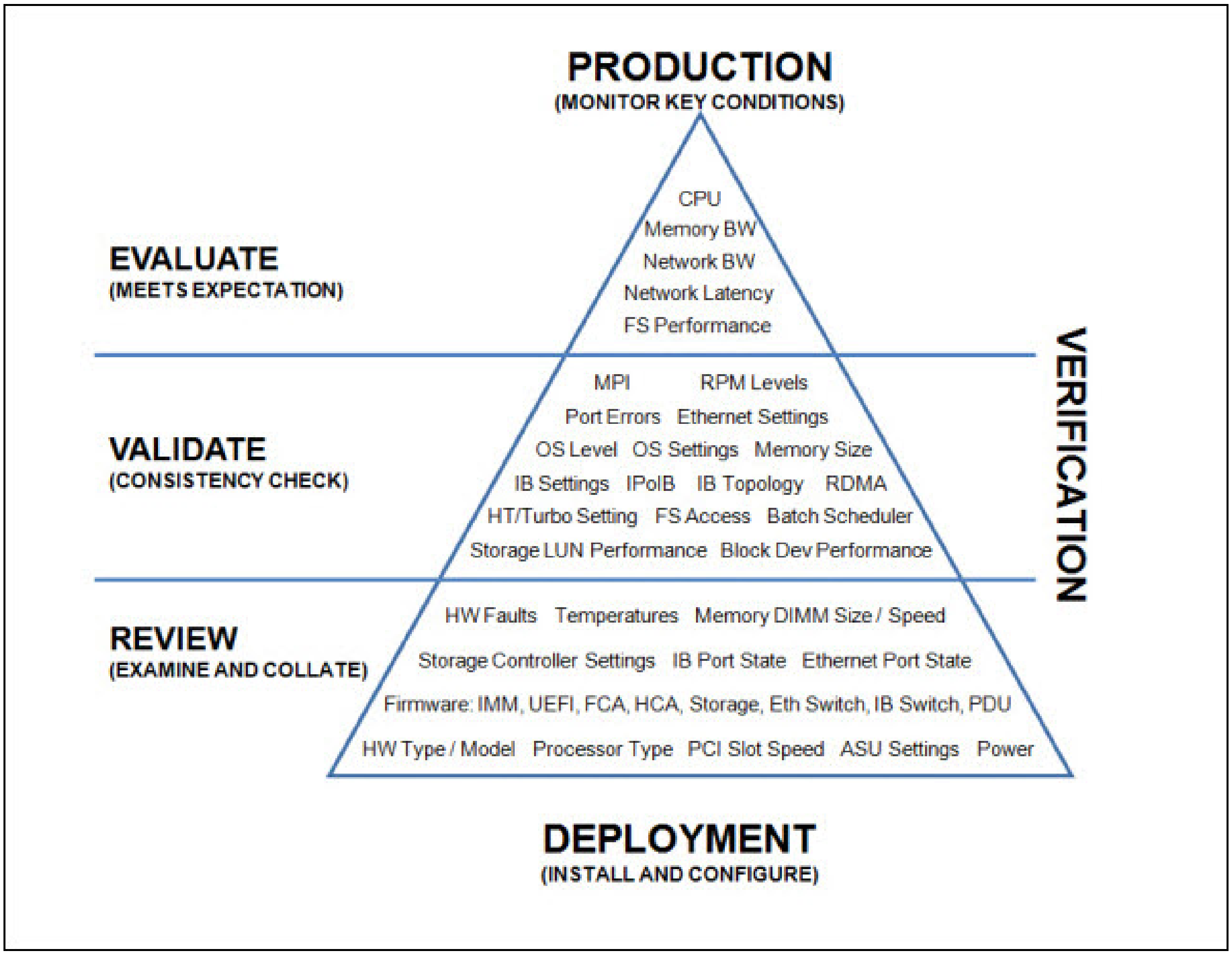

Figure 1. Cluster components verification pyramid

Did you know?

The IBM Cluster Health Check toolset (CHC) is an extensible framework and collection of tools that is used to check and verify the health of an IBM Cluster. The main wrapper tool is hcrun, which provides access to all tools and passes them the environment. Currently, the toolset is available on an *as-is* support basis with the intention of receiving input from users regarding its usefulness. Although many of the tools work in both the x86 and IBM POWER® solutions, the initial toolset concentrates on x86 solutions. There is nothing in the initial framework that precludes integrating tools that are geared toward a POWER solution.

Business value

Health checking involves a series of checking and verification steps. Basically, you must have uniformity among the different individual computers within a cluster system. Usually, thousands of individual computers are involved in one HPC cluster system. To check them one by one manually is not a good approach because manual operations are needlessly time-consuming and error prone. Thus, some kind of automated checking and verification facilities are necessary.

A single, fully automated checking and verification facility for an HPC cluster system do not exist because it is too difficult to develop such a comprehensive facility. There are too many different computer architectures, different types of operating systems, different types of input/output adapters, and different types of storage systems. To implement such a fully automated checking and verification facility, the verification tool itself must be tested and verified in an environment that includes various combinations of all these types of components. This is a difficult goal to achieve because there are too many combinations. Based on these reasons, the administrators of an HPC cluster system should implement their own checking and verification tools to leverage and adapt existing tools to their unique cluster environment.

The IBM Cluster Health Check toolset provides a set of tools that helps check and ensure consistency and stability of the cluster. The focus for the tools is to help reduce initial environment deployment time, minimize the efforts that are required for health state checking, and helps decrease performance variation.

Solution overview

There are significant challenges about how to obtain pertinent health data that is in many layers of clustered systems. Reporting and correlating critical system data and transforming it into usable information is not trivial. Today’s cluster health check tools mostly test components and some point-to-point network performance. They cannot comprehensively monitor and assess overall system health, truly aggregate performance, and reconcile conflicting usage, user expectations, and demands.

In this first release, the health check toolkit attempts to establish a few reference baselines of performance and verify the prerequisite services at the component level, which allows productive testing and detection of anomalous conditions and can even extend to periodic consistency checking at the component level when the tests are not invasive. This approach is particularly important when you try to isolate the cause of performance problems as initial deployment system testing uncovers problems.

These tools do not intend to devalue benchmarking and performance-based modeling as an alternative approach. Some researchers had success with performance modeling results that used strict cluster start and acceptance criteria. This performance-modeling approach does extend the time that it takes to start a cluster and adds considerable expense. Even those who are successful with performance-based modeling admit that there are significant challenges in setting reasonable performance expectations for broadly diversified workloads on new HPC systems to realistically set accurate expectations for performance at all stages of system integration.

Solution architecture

The CHC framework has the following characteristics:

- Display available tools in an organized fashion.
- Organize and display results of the tools with as much or as little information as the user requests.
- Configuration file driven to allow for customization and extension.
- Customize tool execution order through the usage of configuration files.
- Extensible. You can add your own tools and make them available under CHC by updating the configuration files and using key environment variables from CHC.

The CHC tools cover the following broad areas:

- Node health and configuration consistency, and test tools.
- InfiniBand fabric health and configuration consistency, and test tools to verify health of the fabric.
- Utilities.
- Basic configuration checking for consistency across a group of nodes and devices, as well as against a baseline.

Usage scenarios

From build-up to final decommissioning, a cluster passes through different phases during its lifetime in iterative cycles. Updates, changes, maintenance, or outages require methods to run through the stages consistently. Each phase has its own processes, characteristics, and needs. The goal of the methodology is to reduce nonproductive time to a minimum.

The lifecycle starts with the deployment stage, where all the different components are assembled, connected, powered, installed, and configured. While the cluster grows, it must be carefully and continuously examined. Basic functions are verified and uniformity between similar devices must be ensured. Discrepancies and abnormalities must be analyzed and eliminated or they increase with the size of a cluster and ultimately result in negative impacts to the ability to meet expectations. IBM provides preferred practices and recommendations to help prevent you from running into known problems and traps.

More components are collated and tested until the whole cluster is ready for verification. During that phase, a cluster must prove that it is able to perform estimated workloads and meet expectations. This phase is when you attest functionality and performance through extensive testing and evaluation. Verification starts with the smallest entity in the cluster and then scales to the whole cluster size. For example, it is a preferred practice to prove the performance of each single compute node before running a test for the whole cluster. Reducing the size and complexity to a minimum is one of the basic rules about health. If a test fails, the source of the failure must return to the deployment stage until it can meet the requirements.

Finally, after all inspections, a cluster reaches the production phase and must carry out its duty: Run user jobs and return results. To keep the cluster at that state, it is mandatory to monitor all key characteristics continuously. Failures or discrepancies from standards must be quickly identified, the cause investigated, and the source isolated. If it is not possible to separate the cause of a problem, then the cluster on the whole must be verified again, and if a system-wide change is required to solve a problem, then the lifecycle starts again at the beginning of the deployment phase. Insufficient results in one of the stages causes a start from the beginning at the previous phase.

The lifecycle that is shown in Figure 2 does not describe only the phases of the cluster; it also applies to each single component. There is no difference between a large cluster or a single device when it comes to changes. So, even if a single cable must be replaced, the old one must be isolated from the remaining cluster. Afterward, a new cable must be deployed according to preferred practices and carefully verified before you resume production.

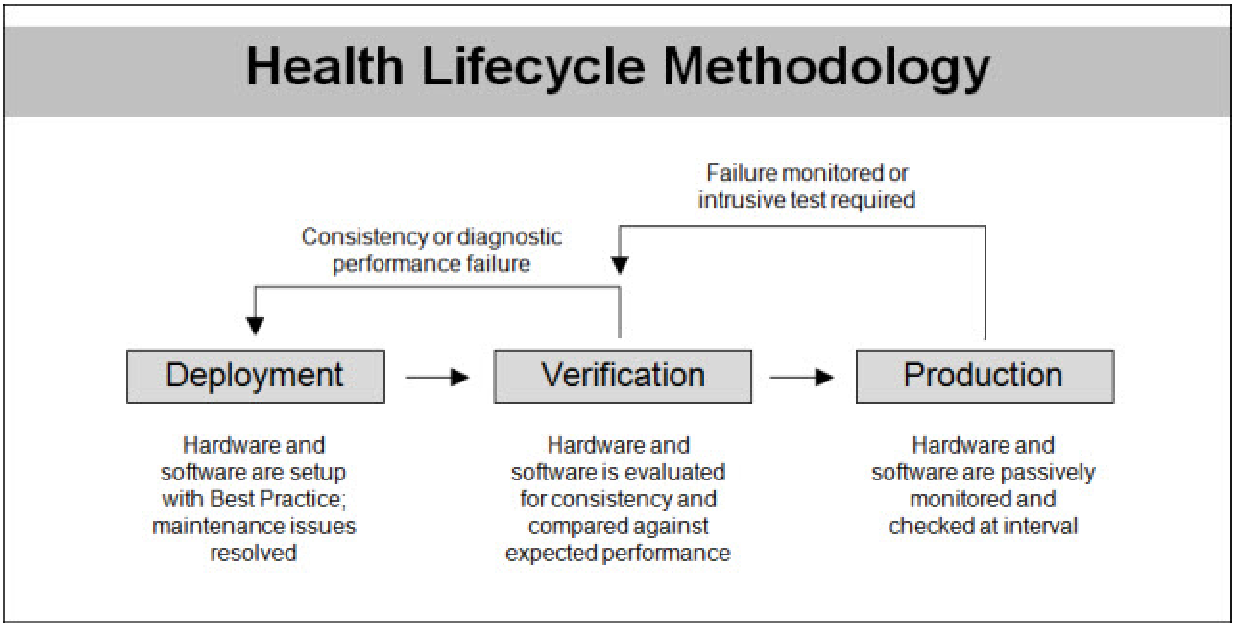
Health Lifecycle Methodology
Failure monitored or intrusive test required
Consistency or diagnostic performance failure
Deployment
Verification
Production
Hardware and software are setup with Best Practice; maintenance issues resolved
Hardware and software is evaluated for consistency and compared against expected performance
Hardware and software are passively monitored and checked at interval

Figure 2. Health lifecycle methodology

It is an HPC rule of thumb that the slowest device determines the overall performance of the cluster. Therefore, extensive verification and continuous monitoring is crucial to obtaining and maintaining a healthy cluster. After a maintenance action, there is no excuse for overlooking one poorly performing memory DIMM that throttles all parallel jobs. It is important to have tests in place to ensure that all critical components work according to their expectations. Most of the testing should be done before combining them with other similar components.

Integration

The following solutions are also part of the cluster delivery solutions in the areas of file system management, cluster installation and management, and networking:

- The General Parallel File System is a high performance, shared-disk file system that provides data access from all nodes in a homogeneous or heterogeneous cluster of IBM UNIX servers running either the IBM AIX® or the Linux operating system.
- xCAT is open source cluster computing management software that is used for the deployment and administration. It provides the following functions:
    - Provisions an operating system on a different architecture, such as Linux, AIX, IBM System x®, and IBM Power Systems™.
    - Creates and manage clusters.
    - Installs and manages many cluster machines in parallel.
    - Remotely manages a system through a remote console or distributed shell.

For more information about xCAT, see the following website:

http://xcat.sourceforge.net/

- The OpenFabrics Enterprise Distribution (OFED) is open source software for RDMA and operating system kernel bypass applications. OFED includes kernel-level drivers, channel-oriented RDMA and send/receive operations, operating system kernel bypasses, and a kernel and user-level application programming interface (API).
  - OFED is available for many Linux distributions, including Red Hat Enterprise Linux, SUSE Linux Enterprise Distribution, and so on.
  - OFED comes with many user space tools that can be used to verify and diagnose the state of InfiniBand interfaces and topology.

For more information about OFED, see the following website:

http://www.openfabrics.org/

Supported platforms

The CHC toolkit should be installed on the xCAT server because many of the individual checks and framework use xCAT commands. In addition, some of the health checks are MPI parallel applications, which run using the POE runtime environment. Here are the required software packages:

- xCAT
- POE environment
- Python 2.6

Although many of the initial tools work in both the x86 and POWER solutions, the initial toolset concentrates on x86 solutions. There is nothing in the initial framework that precludes integrating tools that are geared toward the POWER solution.

Ordering information

The IBM Cluster Health Check tools can be downloaded from the IBM HPC Central at the following website:

http://ibm.co/1hmVBzd

Related information

For more information, see the following documents:

- *IBM HPC Cluster Health Check*, SG24-8168
  http://www.redbooks.ibm.com/abstracts/sg248168.html
- IBM HPC Central
  http://ibm.co/1hmVBzd

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law : INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document was created or updated on December 17, 2014.

Send us your comments in one of the following ways:

- Use the online **Contact us** review form found at:
  ibm.com/redbooks
- Send your comments in an e-mail to:
  redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

This document is available online at http://www.ibm.com/redbooks/abstracts/tips1078.html .

Trademarks