Clustering of OpenVMS installations for high availability

Norman Kluge
norman.kluge [at] student.hpi.uni-potsdam.de

Hasso-Plattner-Institut, Potsdam
Lecture: Dependable Systems
Summer term 2010

Key words: OpenVMS, Cluster, Availabilty, Dependable Systems

1 Introduction

Nowadays services have to be high available. Datacenters are often fault tolerant. That means they are for instance tolerant against hardware failures, software failures or electricity failures. But what happened if a hole datacenter is destroyed. High availabilty means that the service still has to be online. OpenVMSCluster offers concepts of so called disaster tolerance, means even if a datacenter fails, the service never stops the beat. That is ensured by different approaches covered in this paper.

In the first section, some terminology of OpenVMSCluser are introduced. In the second one, some concepts, how OpenVMSCluser ensures high availabilty, are covered. In the las section, some practical experience is mentioned.

2 OpenVMSCluster Basics

In this section some basic terminology of OpenVMSCluster is described.

2.1 Benefits

There are some benefits resulting from clustering of OpenVMS installations. First point is, that the different Workstations can share resources (e.g. disks, network connections, ...) among them. The sharing concept is very important for another important benefit: the promise of high availability. With this and nonstop processing an OpenVMSCluster guarantees that the services running on it are always available and responding the user. That correlates with the benefits of scalability, performance and load balancing, which means that the load is spread into the cluster and all available workstations are working on the task. There are also some security features offered by OpenVMSCluster. How OpenVMS guarantees these features is discussed later.

2.2 Hardware Components

In OpenVMSCluster systems there are gernerally three kinds of hardware components:

Computers are the core of an OpenVMSCluster. They are the ones that doing the computation in the cluster, but a maximum of 96 of them are allowed in the cluster. Their hardware architecture have to be VAX, Alpha or Intel Itanium (IA64).

Storage devices are shared devices among the computers in the cluster. Remote devices are accessed by a MSCP oder TMSCP server (they are discussed later). Storage devices could be Digital Storage Architecture (DSA) disks, RF series integrated storage elements (ISEs), Small Computer Systems Interface (SCSI) devices, Solid state disks or tape devices. As controllers and I/O servers are HSC, HSJ, HSD, HSZ or HSG possible.

Interconnects are the physical connection between the components of the cluster. OpenVMS Cluster systems are supporting a variety of interconnects: Ethernet, Asynchronous transfer mode (ATM), Fiber Distributed Data Interface (FDDI), Cluster Interconnect (CI), Digital Storage Systems Interconnect (DSSI), Memory Channel, Small Computer Systems Interface (SCSI) and Fibre Channel (FC). The last two are only for connecting storage devices.

2.3 Software Components

Above the operating system layer, OpenVMS Cluster systems have additional software components. These services are all running on every node in the cluster. They are called System Applications (SYSAPs).

Connection Manager maintains cluster integrity when a node join oder leave the cluster.

Distributed lock manager synchronizes the operations in the cluster system, especially these on the distributed file system and the distributed job controller.

Distributed file system allows to share the access of mass storage.

Distributed job controller makes queues available across the cluster.

MSCP Server allows to access disks from everywhere in the cluster.

TMSCP Server allows to access tapes from everywhere in the cluster.

2.4 Connection

The structure of the communication system in an OpenVMS Cluster is described in the System Communications Architecture (SCA). The top layer is formed by System Applications (see above). They are communicating ether with the Connection Manager or directly with the System Communication Services (SCS). The lowest layer is formed by port and device drivers, that are relying on the physical interconnect.

3 OpenVMSCluster Concepts

In the following sections, some concepts of the OpenVMSCluster architecture are discussed.

3.1 Integrity of Cluster Membership

The main objective of cluster management is to observe which nodes are active. This job is mainly done by the connection manager. The Connection Manager runs, like all SYSAPs, on every node in the cluster. So, every node observes every other node, if they are still accessable. If not, it sends a message to the other nodes, which are still responding. If they can communicate with the missing member, it is tried to repair the connection link to the missing cluster (On software layer - e.g. an alternative connection path) If the others are also can't access the missing member, it has been failed. The first node which has been noticed the member absence, coordinates the organized removing of that member. This process is called State Transition. For instance, the Lock Manager has to release all the locks which were hold by the lost member. The Connection Manager also checks while the State Transition if the resulting cluster has still the quorum. Otherwise, the cluster goes into blocking state.

If a node joins the cluster, it contacts a member of the cluster. This cluster member is called the advocate of the new member and starts, like in the losing scenario, a State Transition to add the new node. He proposes the reconfiguration of the cluster. Now, every member tries to establish a connection to the new node. After that and an optional reconfiguration of data structures (e.g. for the locking database), the node joins the cluster.

3.2 Sharing

An advantage of clustering computers is the sharing of data among them. Sharing itself leads again to advatages. For instance, data can processed in parallel or data can be back-upped and recovered if a disk fails. This and the next section describes these concepts implemented in OpenVMSCluster.

The Distributed File System of OpenVMS is able to share a local file system among the cluster. Every disks in the cluster is mounted into a global file system. By mounting a device into the cluster, it is required that the device has an unique name. Otherwise the device can not be mounted. The Distributed File System uses the Distributed Lock Manager to coordinate cluster-wide file access.

The Distributed Lock Manager is responsible for synchronizing the access to cluster-wide resources. A particular Lock Manager is responsible for a resource tree. All requests for a locks for an object in this tree is send to the node with the responsible Lock Manager. If a node is lost, all locks hold by this node are released by the proper Lock Manager. The Lock Manager also implements a deadlock detection algorithm.

The MSCP and TMSCP Server makes a disk accessable to a non-local computer. So every computer could access the disk via this service.

3.3 Volume shadowing

With volume shadowing, OpenVMSCluster is able to backup a disk on different locations in the network. It can be seen as implementation of RAID 1, but over a network. A disk can be shadowed to two other disks. Often, these disks are accessed via a MSCP server. In a cluster 500 shadow sets can be present.

3.4 Queuing

The Distributed job controller implements the queuing mechanism in OpenVMSCluster. Users can send jobs to a queue. For instance, a queue can represent the hole cluster (DECnet-Plus only), a specific node or a processor on a node. Generic queues makes it possible to balance the load in the cluster.

4 Project work

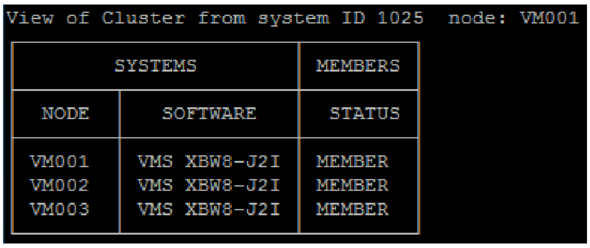
Within the lecture Dependable Systems, my task was to bring three OpenVMS instances into a cluster.

4.1 The setup

The setup were three OpenVMS virtual machines, running version 8.3-1H1 for IA64 systems. First task was to configure them to let them join a cluster. Some problems with this are described below. Second task was to bring this machines into one cluster. The rusult is shown in figure 1: Three nodes within a cluster.

4.2 Problems which hinder the nodes to join the cluster

It is no problem to create a one node cluster. But that makes not that sense. A OpenVMSCluster should consists of a minimum of three nodes (because of the quorum). Afterwards, some problems encountered while setting up the cluster and their solutions are shown.



```
View of Cluster from system ID 1025  node: VM001
┌────────────────────────┬──────────┐
│        SYSTEMS         │ MEMBERS  │
├─────────┬──────────────┼──────────┤
│  NODE   │   SOFTWARE   │  STATUS  │
├─────────┼──────────────┼──────────┤
│ VM001   │ VMS XBW8-J2I │ MEMBER   │
│ VM002   │ VMS XBW8-J2I │ MEMBER   │
│ VM003   │ VMS XBW8-J2I │ MEMBER   │
└─────────┴──────────────┴──────────┘
```

Fig. 1. The not that much exciting result of the task

Non-unique node names was the first problem which has to be solved. While invoking cluster_config.com (see below), OpenVMS asks for a node name. This has to be unique. But this node name has to be still assigned to the node. `modparams.dat` has to be edited and, very important, `autogen.com` has to be called to write the changes into the node configuration.

Non-unique disk names was another problem. After the second node has joined the cluster, the system notified the error:

%SYSINIT−E− volume already mounted on a differently−named device

After that, the system reboots and after the restart, the error appears again. The problem was that the nodes volumes all had the same name, but the Distributed File System need a unique per-volume name. So, the solution was to rename all volumes via the `set volume /label=volname` command (the command asks which volume should be renamed).

Blocking activity of nodes without quorum is the last problem mentioned here. If a part of the cluster has no quorum, it changes to blocking activity. Thats the behavior which is expected, because a cluster without quorum must not answer to the user. The problem is that a blocking node does nothing - not even executing console commands. So, if a cluster with three members is configured to wait for four quorum votes, it will never start working. The only solution is to reboot the nodes and to start the system console - a console with limited features and with non-cluster boot. There the cluster and other parameters can set manually. After that and reboot, the cluster started working.

4.3 Invoking cluster_config.com

If the machines are correctly configured, it is simple to bring them into a cluster. The only invoked command is `cluster_config.com` (alternatively `cluster_config_lan.com`). The following steps has to performed:

1. Choose add or form a cluster
2. Specify the nodes name
3. Specify which cluster the node should join (an integer)
4. Specify the clusters password
5. Decide if there is a shared SCSI interconnect (in my case: no)
6. Decide if this nodes provides its disk for satellites
7. Decide if this node will have a quorum disk

After that procedure, `autogen.com` is invoked to enable this configuration and the node is rebooted. Ideally, the node joins or forms a cluster and if it has already quorum, it starts working.

References

1. Baldwin, L.L., Hoffman, S., Miller, D.D.: OpenVMS System Management Guide. Elsevier Digital Press, second edn. (2004)
2. Hewlett-Packard Company: OpenVMS User's Manual (June 2002)
3. Hewlett-Packard Company: OpenVMSCluster Systems (June 2002)
4. Hewlett-Packard Company: Guidelines for OpenVMSCluster Configurations (January 2005)
5. Parris, K.: Using OpenVMS Clusters for Disaster Tolerance. Hewlett-Packard Company

creative commons

Attribution-NonCommercial-ShareAlike 3.0 Unported

You are free:

- to Share - to copy, distribute and transmit the work
- to Remix - to adapt the work

Under the following conditions:

- Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- Noncommercial. You may not use this work for commercial purposes.
- Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.
- For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.
- Any of the above conditions can be waived if you get permission from the copyright holder.
- Nothing in this license impairs or restricts the author's moral rights.

The document was created by CC PDF Converter