

Ericsson Clustering Model Proposal

Editor

Ibrahim Haddad [Ibrahim.Haddad@Ericsson.com]

Abstract

This is a draft document meant to spark discussing on the topic. It proposes a clustering model and includes answers to some questions raised on the mailing list. The document is kept to minimal size to focus on key issues.

Table of contents

Background	2
Clusters in the telecom space	2
Requirements	2
Proposed Clustering Model	3
Why adopt this model?	4
Advantages of the proposed clustering model	5
Target applications	5
Cluster Middleware	6
Relation to DSI	7
Conclusion	7
References	7

Background

Today's commercial and telecommunication environments are increasingly adopting clustered servers to gain benefits in performance, availability, manageability and scalability. In the general term, a cluster is a collection of servers that work together to solve a problem and share resources, such as disks, file systems and devices. The resulting benefits of a cluster are greater or more cost-efficient than what a single server can provide. There are several benefits of clustering:

1. *High availability* through redundancy and failover techniques, which isolate or reduce the impact of a failure in the machine, resources, or device.
2. *Manageability* through appropriate system management facilities that reduce system management costs and balance loads for efficient resource utilization.
3. *Scalability and performance* through expanding the capacity of the cluster by adding more servers, of in terms of servers, adding more processors, memory, storage, or other resources to support growth and to achieve higher level of performance.

In addition, the usage of commercial off-the-shelf based cluster systems has a number of advantages such as a better price/performance ratio when compared to specialized parallel supercomputers, deployment of the latest mass-market technology, as it becomes available at low cost, and added benefits from latest standard operating system features, as they become available.

Clusters in the telecom space

The telecommunication industry's interest in clustering originates from the fact that clusters address carrier-class characteristics such as guaranteed service availability, reliability and scaled performance, using cost-effective hardware and software.

Linux is becoming the operating system of choice for many carrier-class platforms running telecommunication applications. There are many benefits of deploying Linux, such as an improved price-to-performance ratio, which translates into support for open architectures and standard interfaces, in addition to support for multiple hardware architectures, and result in significant cost savings.

Requirements

What is the "strict set of requirements" that telecom platforms have which automatically excludes most types of clustering models? Are these the type of clusters where a service may not be available while the service is failing over to another active node?

Without being absolute about these requirements, they can be divided in these 4 categories:

- Short failure detection and failure recovery
- Guaranteed availability of service (99.999% - or less than 5 minutes of service unavailability to end-users)
- Short response times with no (or rarely any) "actions" that require long execution times
- Security requirements

Proposed Clustering Model

Clustering

Clustering is the use of multiple, interconnected nodes, to form what appears to users as a single highly available system.

Nodes

Nodes are independent computers (they can be COTS) that cooperate over network interconnections to provide a service.

Nodes do not share memory banks, IO channels, nor buses, or anything alike. They do share access channels, buses, ... they share access but typically are not physically wired to share the same memory for instance. One example of shared resources is disks.

A cluster node, or node, can be diskless or it can have local disk storage.

All nodes share access to a common disk volume.

A node is a standalone server that can have one or more CPUs or can be an SMP machine.

A cluster is a collection of interconnected nodes that offer a service to users (or subscribers). The service is required to be available to users at all times.

Are all members of the cluster "equal"?

All members of the cluster are equal. Yet, from a logical or physical point of view some nodes might have a different significance in the cluster.

For example, not all processors will necessarily run at the same speed or have the same amount of memory. There might even be some specialized HW on some nodes (e.g. encryption/decryption chips). The cluster would be able to make use of these parameters to restrict/prioritize the execution on certain nodes.

How many nodes would be in this cluster?

The cluster would contain at least two nodes.

The maximum number of nodes is only dependent on system limits that we want to impose ourselves. For example, if we decide to limit in the protocol the member number parameter to 8 bits, we would have max 256 nodes. As the principle is to share nothing, we should be able to increase the number of nodes to a very high number.

Storage & Disks & SAN Support

On nodes with disks, private (i.e. local) storage is allowed (depends on scenario, application, ... etc).

On diskless nodes, these nodes share access to a common pool of storage (RAID support is a must).

A distributed file system is required.

SAN support is desirable.

A sensibly replicated storage (RAIDs etc) makes it transparent to applications to add/remove disks/nodes from the system.

IP Address Scheme

The selected IP-addressing scheme must allow transparent load balancing and service migration as well as explicit access to a named node through a named interface.

Separation between OS and MW

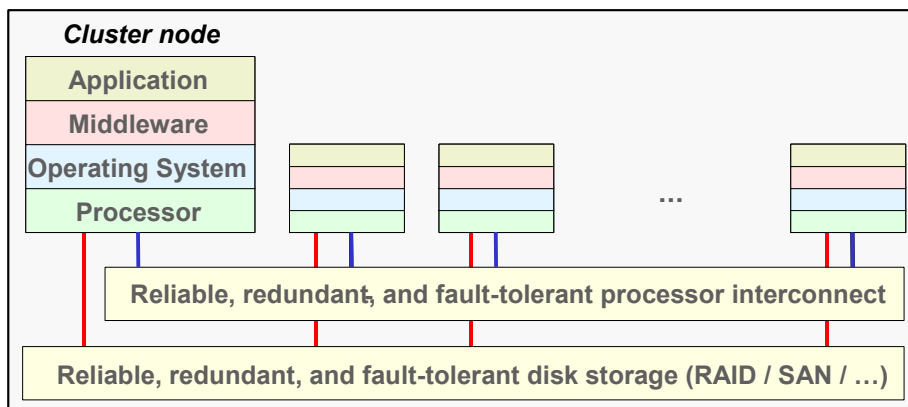
The division of work between OS and HA-middleware should be clearly defined.

Redundancy scheme

Which scheme should we to follow for redundancy?

How will it affect the cluster/platform configuration?

Cluster logical view



Why adopt this model?

1. A general cluster model that defines a clustering solution with “loosely coupled”¹ nodes.
2. It is not a specialization; rather it is a basic cluster model that could be extended (in the future) with specialization techniques (if required).
3. Using loosely coupled nodes as the base of the clustering solution gives more flexibility than a tighter coupling between nodes.

For applications that would need tighter coupling (e.g. high performance computing applications), we can make additions to a loosely coupled cluster to tighten the system with

¹ See references for a paper that defines the characteristics of loosely coupled nodes.

an extra layer of software (middleware). On the other hand, a tightly coupling would inhibit applications, like telecom applications and web applications to use a tight cluster.

It is easier to add a tight coupling layer to a loosely couple cluster as the base then the other way around.

Advantages of the proposed clustering model

1. The loosely coupled cluster model is a basic clustering technique that is suitable for the type of applications CGL servers will host. Specializations are not excluded but could be handled later and separately.
2. Following this model, the probability of a failed shared component affecting the availability of the service or the availability of system does not exist.
3. When performing software or kernel upgrades, the procedure will be done on each node separately without affecting the availability of the service.
4. In case of a hardware fault, a specific node will be affected; it will be replaced or fixed without affecting the uptime (no unscheduled downtime).
5. In case of a software fault or bug on a certain node, the specific node will be affected. The platform will still be providing service through the other nodes.
6. In case of hardware upgrades, each node will be upgraded separately without affecting service availability.
7. You can increase the number of nodes as your load/traffic increases.
8. This model eliminates the node being a single point of failure.

Target applications

What are the target application(s) or platform(s) for this cluster proposal?

From the basic system (without specialized extensions), applications with short response times like (but not exclusively) Web servers, Authentication Authorization and Accounting (AAA), Policy servers, Home Location Register (HLR), Service Control Point (SCP), Application servers (J2AS).

What telecom application(s) or platform(s) would not work with this cluster proposal?

Base stations or other “hard real-time” applications or nodes.

What about specialized applications that require special middleware?

When you have applications that require specialized HW, either you can co-locate them on the same nodes that have their specialized HW, or the software part of the applications could be located on any nodes if “proxy” functionality is installed on each node that has the specialized HW.

Cluster Middleware

What information would these nodes need to share?

Share is probably not the right word. Rather each node needs to be aware of the cluster topology; nodes share configuration information (CPU, Memory, etc), and have a common naming tree. These are known only to the platform.

Applications do not need to be aware they are executing in a cluster. The word “share” could lead one to believe that every node needs to know and have immediately available the whole cluster naming tree or cluster topology. It is not the case.

What are the advantages of a node being a "member" of this cluster as opposed to just a farm of distributed compute engines?

Short failure detection and failure recovery, replacement of SW and HW with minimal disturbance, single entry point for operation, administration and maintenance of the cluster.

What is required to "manage" the cluster? Is it a requirement to be able to administer the cluster as a single system?

It is not a requirement; but being able to administer the cluster as a single system is definitely a benefit.

Is there any "cluster" software being used in this cluster proposal or are these just nodes operating in a distributed environment?

The proposal is not only a set of nodes operating in a distributed environment; there is additional SW involved in the failure detection, recovery, cluster (re-)configuration, etc.

Stateful failover: One aspect that I didn't see brought out, that does separate the serious contenders from the bystanders, is the idea of stateful failover, where the session data is maintained across a failover and state information is preserved. Telco protocol stacks are stateful, but web servers are not, and many clustering packages only aim at web services.

Stateful failover is needed in quite a few applications in CGL environment. That work is being done in Service Availability Forum. SAF works on API specification documents and the support for stateful failover is one of the very important aspects there. SA Forum is operating system agnostic, but everything they do there is applicable to CGL environment as well. Depending on the result of SAF, we are likely to include their result in the proposed solution

A highly available service vs. a highly available system:

Typically, aside from having a HA system, the service (application) provided by the system must be HA. End users do not really care if the system is up and running. They care that they are able to use the services or applications provided by the system. Therefore, both the system and applications must provide HA features.

Is there any "cluster" software being used in this cluster proposal or are these just nodes operating in a distributed environment?

There are different pieces: one of them is TIPC.

It also depends on what you mean by "cluster" software. Do you mean application "cluster-aware"? Middleware "cluster-aware"? OS software "cluster-aware"?

The idea should be to hide the "cluster-aware" part of the software as low as possible in the software stack. The less the applications are aware of the cluster, the better are the chances to be able to use an Off The Shelf software and use it as is in the cluster with enhanced capabilities (HA, scalability, etc.).

Relation to DSI

Does the security part of the proposal (DSI) rely on the notion of cluster membership?

DSI has no requirements imposed at the cluster model to be used. In fact, DSI can be used with normal PCs that do not even form or are part of a cluster.

DSI is being proposed to OSDL in a separate document.

Conclusion

The proposed clustering model is a general, non-specialized model that has been demonstrated to be the most suitable for telecom platforms, especially when it comes to meeting high availability requirements.

By going forward with this model, OSDL CGL WG will insure that its clustering model will be targeted for carrier-grade platform and telecom applications; in addition, it is aligned with the telecom views of clusters. In addition, OSDL CGL WG will not be creating a new clustering model; rather, it will adopt an existing model that provides a general tested and proven solution. In the event that OSDL CGL WG at one point in the future needs some specific specialized cluster model, this model can be easily re-architected to meet specific needs.

References

OSDL CGL Glossary

http://www.osdl.org/lab_activities/carrier_grade_linux/glossary.html

Ericsson Linux Site (includes links to TIPC, DSI, and AEM)

<http://www.linux.Ericsson.ca>

TelORB

<http://www.telorb.com>

Overview of Distributed Systems (include definition of loosely coupled nodes and SMP)

<http://www.cs.ucla.edu/classes/spring00/cs111/section-kampe/distcomp.pdf>