

# Requirements Specification

## *Colorado Phase I: Infrastructure*

*Ellard Roush,*

*Sun Microsystems, Inc.*

*7 October 2008*

*Feature 2008/1717*

### **Revision History**

<i>Version</i>	<i>Comments</i>	<i>Date</i>	<i>Author</i>
1	Initial draft	08/1/08	Ellard Roush
2	Revise split-brain handling	8/11/08	Ellard Roush
3	No QL or RU support	8/13/08	Ellard Roush
4	Add clarification for Ed's comments	8/16/08	Ellard Roush
5	Add info about upgrades	8/25/08	Ellard Roush
6	Optional Pxf's not supported initially	10/1/08	Ellard Roush
7	Revise based upon CLARC review	10/3/08	Ellard Roush
8	Make document public	10/07/08	Ellard Roush

## **1. Goals**

The Colorado Phase I project proposes to support a "light weight" cluster in addition to the traditional enterprise cluster. Any Colorado Phase II requirements will be evaluated after Colorado Phase I completes.

One goal is to be able to support developers with a minimal amount of hardware. This can be achieved with existing capabilities. A single machine can support 3 virtual machines, where each virtual machine acts as a Sun Cluster node. The virtual machine could be an LDom Guest Domain on Sparc machines or alternatively, Virtual Box could be used. Another approach is run Sun Cluster in two virtual machines and a quorum server in a third virtual machine. This approach has already been successfully used. These two approaches provide a solution where the developer only needs one machine and no networking hardware is needed. This is very simple and inexpensive solution. No development is needed to achieve this goal.

Another goal is to support a cluster where all nodes remain operational despite a split brain. This is not allowed under the current design, and will require new capabilities. The market for this solution is applications where changes on one partition can be discarded after a split-brain scenario. It was suggested that LDAP could be a candidate for this model.

A third goal is have a small foot print. One small change is needed in this area.

## 2. Minimize FootPrint

The Sun Cluster infrastructure has one feature that consumes considerable memory. The Sun Cluster infrastructure has a **threadpool** handling incoming invocations that is quite large. The number of threads varies with available memory. A typical cluster configurations contains about 1000 threads. The number of threads enables the software to avoid a resource exhaustion deadlock that involves Pxf. When this project ships, Pxf will be optional. The **threadpool** size can be reduced significantly when Pxf is not present. We expect to reduce the number of threads from 1000 to 100. When Pxf is present, the number of threads will not change.

R1: When the cluster configuration does not support Pxf, the default number of threads in the invocation server threadpool will be reduced to 100 threads.

Note that we do not support dynamic changes in Pxf support. This check is only made when the node is booted. In order to change this configuration the entire cluster must be rebooted.

Colorado-I will not support making Pxf optional. Thus this feature will not be implemented for Colorado-I. This requirement may be part of Colorado-II or Colorado-III, but only if Optional Pxf is supported in that release.

## 3. Features

The Colorado-I release will not support all existing features. Specifically there will not be Rolling Upgrade (RU) or Quantum Leap (QL) support. Both RU and QL are features for enterprise versions of Sun software.

R2: Colorado-I will not support Rolling Upgrade or Quantum Leap from the SC3.2 product to Colorado.

R3: Colorado-I will not support Rolling Upgrade or Quantum Leap from one developer edition to another developer edition release.

The Solaris team has not decided how Zones will be supported in OpenSolaris. So Colorado-I will not support Zones. This means that Colorado will not support 1334 zones and will not support Zone Clusters

R4: Colorado I will not support non-global zones.

## 4. Weak Membership

Today, Sun Cluster supports a form of membership that ensures that only one cluster partition is active at any time. We will call this the *Strong Membership* model.

This project will add support for a new form of membership that allow all nodes to remain operational despite a split-brain condition. This will be called the *Weak Membership* model. This section describes the requirements for this new feature.

## 4.1 Overview

When all nodes are alive in the same cluster under *Weak Membership*, the system behavior is identical to system behavior under *Strong Membership*. Thus, this section just needs to describe the different behavior that occurs during a split-brain.

When a split-brain occurs, the nodes are both up and cannot communicate. Each node is ignorant about whether the other node is up. Each node forms a cluster consisting of one node. In effect, each node assumes that the other node is down regardless as to whether the other node is actually down. More specifically, each node performs a node reconfiguration and establishes a cluster with a membership of just that node. Any data service that was on the other node just prior to the split-brain is now launched on this node, and that includes plumbing IP addresses, mounting file systems, and launches programs. If both nodes can reach the wider public network, there will be a collision as the two nodes attempt to host the same IP address. If the administrator wants the data service to run on only one node, the administrator specifies only one node in the Resource Group.

While in the split-brain situation, the administrator can make configuration changes on either node.

Similarly, applications can make changes in the same file systems on different mirrors.

Eventually, repairs correct the communication failure that caused the split-brain. At that time the administrator must follow manual procedures to repair the cluster. The system does not automatically resolve data conflicts in system configuration information (CCR data). The system does not automatically resolve data conflicts in file system data. In both cases the administrator chooses the information on one node and discards the information on the other node. At least one node must be rebooted before the cluster can be reformed with all cluster nodes.

The cluster nodes cannot differentiate the case where the other node died from the case where network communications failed. So the node assumes that a split-brain happens whenever the node cannot communicate to the other node.

Refer to the following sections for more detailed information.

## 4.2 Configuration Limitations

The *Weak Membership* model will only be supported under certain conditions.

R5: Weak Membership can only be selected when there are exactly 2 nodes

When there are more than 2 nodes are present *Strong Membership* works.

R6: Weak Membership can not be selected when there is any shared storage.

When shared storage is present *Strong Membership* works.

The above requirement does not prohibit the use of local storage on other nodes that is accessed via iSCSI.

This also means that all storage is local from the perspective of fencing and hence the fencing feature is not active under *Weak Membership*.

R7: Weak Membership can not be selected when a quorum device is configured.

Weak Membership does not support quorum devices.

R8: Strong Membership can not be selected when the cluster still has unresolved CCR updates that were done during a split-brain.

This information is only recorded while Weak Membership is in effect, and this issue must be resolved before changing the membership model.

### 4.3 Installation

The installation process needs to support the ability to install any supported configuration. However, the installation process should be made easy for the common case. It is expected that the number of configurations that will choose Weak Membership will be a small portion of the total number. Sun Cluster supports 2-node clusters with no quorum devices. When an administrator wants to use Weak Membership, the admin does the normal installation, except that the admin does not configure any quorum device. After the basic installation process for Sun Cluster completes, the admin then selects Weak Membership. This means that there will be no change to the installation process, and this approach still fully supports Weak Membership.

### 4.4 *clmembership* command

A new command is needed to allow the administrator to select which membership model will be in effect.

Note that the final form of the command may change during the design phase. The System Management team may choose a different form of the command. So the requirements in this section correctly identify the functionality, but the format may change.

R9: The new **clmembership** command will enable the administrator to choose either the *Strong Membership* or *Weak Membership* model.

R10: The default membership model for the cluster will be *Strong Membership*.

R11: The **clmembership** command will have an option to display the active membership model.

### 4.5 Connectivity Check

There are conditions under which it is not desirable to allow a node to continue to remain operational after a split brain.

R12: The administrator must specify using the **clmembership** command at least one IP address that must be reachable in order for the node to remain operational after a split-brain occurs when the cluster operates under Weak Membership.

When a split-brain occurs under Weak Membership, the system will ping each IP address

configured as a Connectivity check. The node commits suicide should the Connectivity checks fail. Under some scenarios, this health check will kill one node and the other node will survive, thus avoiding the case where multiple partitions remain after a split-brain.

- R13: The **clmembership** command will have an option to display Connectivity Checks, and this information will include the IP address(es) that the node(s) will attempt to ping.
- R14: The **clmembership** command will have an option to remove a specific Connectivity check.
- R15: The **clmembership** command will remove any configured Connectivity checks when changing from Weak Membership to Strong Membership.

Connectivity checks are only used by Weak Membership and are required by Weak Membership. So Connectivity checks are part of the Weak Membership configuration information.

## 4.6 Split-Brain

- R16: Under *Weak Membership*, the nodes of the partition remain operational after a split-brain as long as the node passes ALL configured Connectivity checks.

Once a split-brain occurs under Weak Membership, then the administrator has to manually do the recovery process. One node will be the *winner* and the other the *loser*.

- The administrator selects the *winner* node, which remain in cluster mode.
- The administrator reboots the *loser* node into non-cluster mode. The admin then resolves any potential CCR data change conflict by marking this CCR as only valid until rejoin using a new utility.
- The Admin clears any CCR data change conflict on the *winner* node by selecting this node using a new utility.
- The Admin resolves any file system conflicts (refer to the GDD documentation).
- The Admin reboot the *loser* node into cluster mode.
- The cluster reforms with both nodes and normal operation resumes.

## 4.7 Cluster Rejoin

Under *Weak Membership*, we cannot allow the cluster to automatically reform. There could be incompatible changes on the partitions.

Today, once a cluster node departs the cluster, that node cannot rejoin under the same node incarnation number. This check exists in **path\_manager::update\_node\_incarnation()**.

We need to prevent a cluster from forming when there are unresolved CCR data changes from a split-brain occurring under Weak Membership.

- R17: Each node will not allow another node to join when the local node has unresolved split-brain CCR data changes that occurred under Weak Membership.

The `path_manager::update_node_incarnation()` function could be enhanced to also check for the presence of information in the CCR indicating that CCR data changes occurred during a split-brain under Weak Membership.

## 4.8 CCR Incompatibilities

Under *Weak Membership*, CCR data changes are allowed after a split-brain. The Strong Membership model ensures that only one cluster partition can be active at any time, and the Strong Membership model supports *Amnesia* protection. So the existing Sun Cluster product does not have to resolve conflicting CCR data changes.

Automatically resolving CCR data changes will not be done !

The CCR records an epoch number in the CCR epoch file, which specifies a form of a cluster incarnation number. This number cannot be used to resolve conflicts. The CCR records **gennum** for each CCR data file. The **gennum** increases serially. So a CCR data file on different nodes could be changed in incompatible ways and wind up with the same **gennum** number on different nodes.

It would be desirable to know whether changes have been made during split-brain while Weak Membership is in effect. A new field could be added to record this information (the CCR epoch data file appears to be appropriate. More investigation is needed to confirm this).

R18: When a CCR data file is updated during a split-brain while Weak Membership is in effect, the CCR will record that a split-brain CCR data change has been made.

When Strong Membership is in effect, the design prevents conflicting changes. So the CCR will never record that potentially unresolved CCR data change has been made. Naturally, this also means that an admin never has to resolve any potentially unresolved CCR data changes.

The **ccradm** utility has the ability to mark a specific CCR data file as valid only until the node rejoins the cluster, at which time the CCR subsystem will replace that CCR data file with a valid copy from the cluster. Requiring that an administrator do this operation for each CCR data file would be a lot of work, because there can be many CCR data files. The **ccradm** utility has not yet been updated to comply with the new CLI standards.

R19: A command will provide an option to mark all CCR data files as valid only until the node rejoins the cluster, at which time the system will replace these CCR data files. This will run in non-cluster mode. This command will clear any information about an unresolved CCR data change that occurred during a split-brain under Weak Membership.

The administrator uses the above command to resolve any spit-brain CCR data changes on the node whose CCR changes will not be retained.

R20: A command will support the ability to remove information indicating the presence of any unresolved CCR data changes. This command will run in cluster mode.

The administrator will use the above command to tell the system to retain the CCR data changes that occurred during a split-brain on this node. This command operates in cluster mode in order to avoid an unnecessary reboot.

R21: A command will provide an option to report whether there are any unresolved split-brain CCR data changes. This command can be run either in cluster mode or non-cluster mode.

The above command enables the administrator to determine whether there are unresolved split-brain data changes.

## **4.9 File System Data Incompatibilities**

Under *Weak Membership*, applications can continue to read and write file system data. This can lead to incompatibilities, which Sun Cluster cannot resolve. There will be no automatic resolution and no tracking as to whether any conflicts occurred. The administrator must resolve the conflicts. (Note this is really in the GDD area).

R22: There will be documented manual procedures for making the file system data of one partition the surviving file system data prior to reforming the cluster after a split-brain under *Weak Membership*.

R23: PxfS will not be supported under *Weak Membership*.

R24: Shared QFS will not be supported under *Weak Membership*.

R25: Cluster volume manager devices will not be supported under *Weak Membership*.