INDIANA UNIVERSITY

High Level Architecture
For
UID/GID Mapping

Revision History

Date	Revision	Author
12/18/2012	1	jjw

Table of Contents

I. Introduction

UID/GID Mapping allows the Lustre file system to be used across clusters' heterogeneous user populations while still respecting POSIX file ownership, permissions, ACLs and quotas in a manner without credential timeouts or methods to transfer credentials across a client cluster.

II. Definitions

Cluster

File system UID/GID
The UID/GID that represents the owner/group of the file or directory as it is stored on the Lustre file system.

Client UID/GID
The UID/GID that represents the owner/group of the file or directory as it is represented on a Lustre client.

Canonical UID/GID Map
The UID/GID map held on the MGS server which serves as the master.

Working UID/GID Map
The UID/GID map held in memory on the MDS/OSS nodes.

III. Changes from Solution Architecture

None

IV. Functional Specification

Structure of Cluster Definitions and UID/GID Maps

The MGS contains a linked list of the canonical cluster definitions and UID/GID maps for those clusters. The canonical list of cluster definitions and UID/GID maps kept in memory on the MGS as a linked list of cluster list versions, with each

version in memory having a unique version identifier. Only the latest version is written to disk.

Each cluster will include a textual name for administrator identification, a file system UID and GID to assign to unmapped client UIDs and GIDs, flags indicating whether the cluster is trusted and the client UIDs and GIDs that should be used by the file system.

Each cluster definition contains a pointer to a linked list representing one or more NID inclusive ranges that comprise the cluster. Any NID can be contained in only one range for only one cluster definition in the latest version of the cluster lists.

Each cluster definition contains four pointers to red-black trees that provide fast forward and reverse mapping between client and file systems UIDs and GIDs.

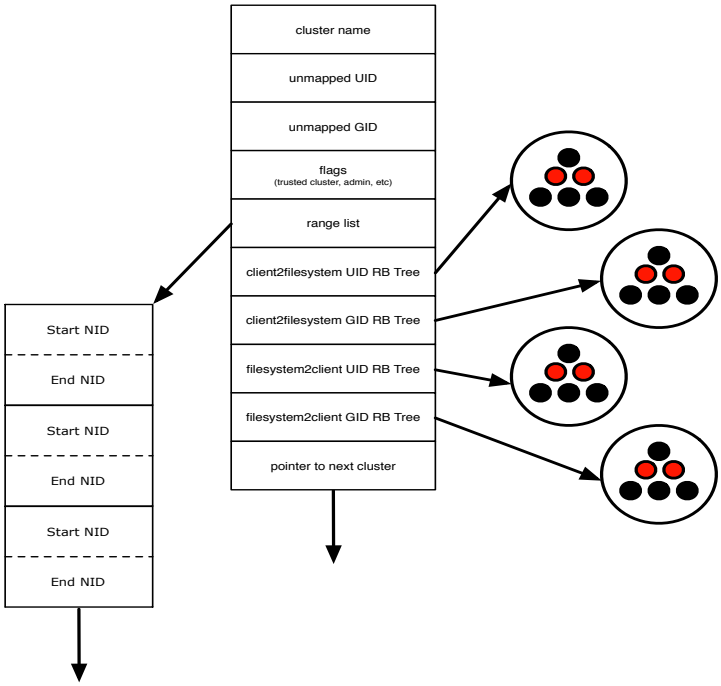

Figure 1

File system administrators will create definitions of client clusters on the Management node via the lctl command line tool.

Nathan Rutman 1/8/13 10:03 AM
Comment [1]: So after server reboot old reconnecting clients will lose their mappings.

Nathan Rutman 1/8/13 10:04 AM
Deleted: client

Nathan Rutman 1/8/13 10:06 AM
Comment [2]: Will the code check for this?

Changes to the cluster definition are made and validated on the MGS in memory before being committed to disk. Multiple changes can be made before an explicit commit reducing the number of updates that take place during complex operations. If any changes to the cluster definitions fail to validate, all changes will be discarded.

After a cluster definition has been committed, a mapping between client UIDs and GIDs and file system UIDs and GIDs can be attached to the cluster definition via the lctl command line tool.

The exported functions for administering the cluster list and the associated UID/GID maps will be encapsulated into a kernel module.

Distribution of the Cluster Definitions and UID/GID Maps

Upon initialization, each MDS and OSS node contacts the MGS, sets a callback for updates, and receives the latest version of the cluster definitions and UID/GID map.

When the map is updated on the MGS, callbacks for the update are made for each server. Each MDS and OSS server provides the version id of the cluster definitions to the MGS. The MGS will compute the delta between the version current on the server and the latest version. It will then transmit only the changes to the servers to update their cluster definitions to the latest.

Nathan Rutman 1/8/13 10:21 AM

Comment [3]: Instead, maybe we should feed a complete definition text file directly in? (Along with the uid/gid list?) FWIW, I always vote for YAML.

Nathan Rutman 1/8/13 10:11 AM

Comment [4]: We probably want /proc files to: list the clusters, print each cluster definition, print each cluster map

Nathan Rutman 1/8/13 10:17 AM

Comment [5]: If there are independent versions per cluster def, the implication is that this mechanism will handle multiple update files. If true, I'd like to suggest this mechanism be made as generic as possible, to allow for other "push" updates of configs. Maybe include a header: push ver, config type, config ver, config id, flags, data

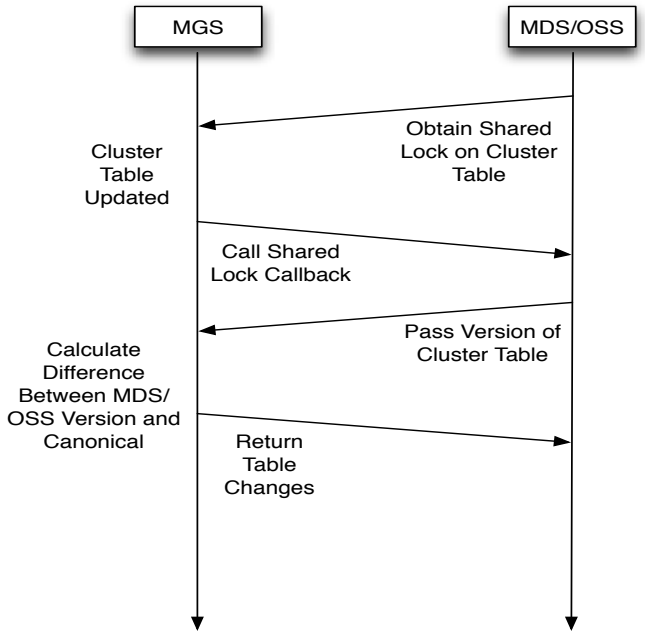

Figure 2

Client Mounting

When a client mounts the file system, it logs into each server and an export structure allocated. During this operation, the NID will be looked up in the cluster definition linked list by checking against the range list. A pointer to the cluster definition containing the NID of the connecting client will be added to the exports structure.

If a UID/GID mapping is toggled on, and the connecting NID is not in a cluster definition range, the pointer will be left NULL.

Upon client definition and map updates, the exports will be walked to ensure that the exports structure for each client contains a pointer to the correct cluster definition.

UID/GID Lookup

When a request is made of the MDT, the mapping from client UID/GID to file system UID/GID will be handled at the unpacking of the request at the beginning of its lifetime, and mapping from file system UID/GID to client UID/GID when the response is packed.

When a request is made of the OST, the UID/GID mapping needs to occur prior to a write and any quota operations, as the only reason for UID/GID information on the OST is to maintain quotas.

Nathan Rutman 1/8/13 10:20 AM
Comment [6]: Sorry if I asked before – is this true with the new quotas code?

Command Line Tools to Manipulate Clusters

Additions will be made to the lctl command line tool for cluster administration. Functionality will be added so that clusters can be defined, modified, and deleted, and maps associated with clusters can add or remove map nodes. The command line tool will only be able to manipulate the cluster list and maps from the MGS. Changes to the cluster list and map will be written to disk when the MGS is issued the command (via lctl) to commit the new configuration. The configuration will not be active nor will it be distributed to MDS and OSS servers until it is committed.

Servers will be able to list the cluster list and map content using the lctl command, but not be able to affect the map.

Additionally, the capability of a full reload of a server's working cluster list and UID/GID maps will be available via the lctl tool.

V. Implementation Milestones

To be done.