Trove and PVFS2

PVFS Development Team

June 28, 2006

1 Motivation and Goals

The purpose of this document is to describe the use of Trove in PVFS2.

PVFS2 deals with four basic types of objects:

- directories
- metafiles
- datafiles
- symlinks

We will discuss how Trove is used to store these objects in the upcoming sections. Additionally we will cover how PVFS2 bootstraps from the Trove perspective; that is, how it obtains a "root handle" and so on.

2 Current Implementation (03/21/2003)

This section describes the implementation as of the date above. Eventually, when the implementation catches up with the long term plan, this entire section will probably disappear.

2.1 PVFS2 Objects

At this time the type of an object is stored in at least one place, the dataspace attributes. These can be retrieved using `job_trove_dspace_getattr`. Additionally, as we will see, directories and metafiles store metadata as keyvals.

Directories are of type `PVFS_TYPE_DIRECTORY`. They are actually stored as two dataspaces in the current implementation. The first dataspace is used solely to store the attributes of the directory (under the key `metadata`, as a `PVFS_object_attr` structure) and, if entries have been created, the handle of a second dataspace where the directory entries are stored (under the key `dir_ent`, as a `PVFS_handle` type). The second dataspace is marked as type `PVFS_TYPE_DIRDATA` to differentiate it. This one holds the directory entries, with keys being short names of

files in the directory and values being the corresponding handle (stored as a `PVFS_handle` type). This dataspace is created lazily when the first entry is created (in the `crdirent` state machine).

Metafiles are made up of a single dataspace and are of type `PVFS_TYPE_METAFILE`. Basic attributes are stored in a keyval under the key `metadata`, as a `PVFS_object_attr` structure, just as in the directory case. An additional keyval (`datafile_handles`) stores the array of datafile handles as `PVFS_handle` types. A final keyval (`metafile_dist`) stores the distribution information (in some arbitrary format at this time).

Datafiles are made up of a single Trove dataspace of type `PVFS_TYPE_DATAFILE`. Currently there are no attributes stored for datafiles, and all data is stored in the bstream.

Symlinks are not currently implemented, but the intention is to use a keyval to hold the target of the link.

2.2 Bootstrapping

WHERE DO WE GET THE ROOT HANDLE?

WHAT ELSE?

3 Long Term Plan

This section describes the (probably moving) target for how we will use Trove to store PVFS2 objects. Eventually, as we progress, this will start to describe interesting things such as storing small files in the metafile...

3.1 PVFS2 Objects

The biggest overall change is the move to using dataspace attributes for storing basic metadata for PVFS2 objects.

Directories are made up of a single dataspace. Dataspace attributes are used to store basic metadata. Keyvals in the dataspace are used to store the directory entries in (short name, handle) format as before.

Metafiles are made up of a single dataspace. Basic metadata is stored in the dataspace attributes as with directories. The keyval space is used to store additional attributes, including the list of datafile handles (TODO: WHAT ELSE?).

Datafiles are made up of a single Trove dataspace, with basic metadata stored in the dataspace attributes and data stored in the bstream (TODO: WHAT KEY?).

Symlinks are made up of a single Trove dataspace with basic metadata (if any?) stored in the dataspace attributes and target stored as a keyval (TODO: WHAT KEY?).

3.2 Bootstrapping

WHERE DO WE GET THE ROOT HANDLE?

WHAT ELSE?