

HBase Namespaces

HBase presently does not have a means to provide tenants with a project space in which they can create and manage their own tables. This document proposes the addition of namespaces to fill this need.

Namespace Definition

Beyond having a name which is globally unique, a namespace has the following characteristics:

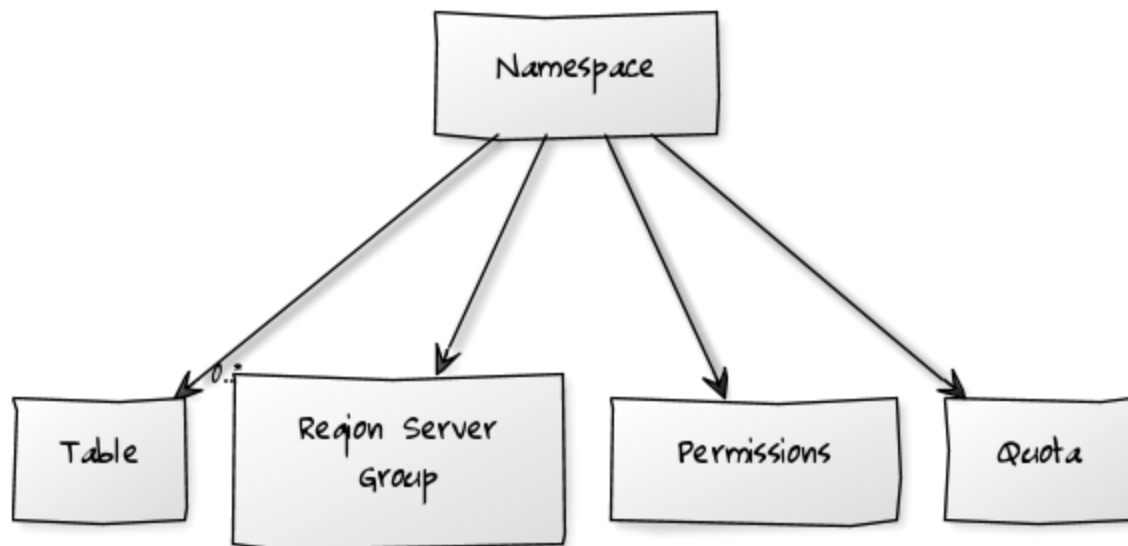


Table Membership

All tables are members of a namespace, tables with no explicit namespace will be a member of the “default” namespace. A table can only be a member of a single namespace and once defined is permanent.

User/Group Permissions

Leveraging the existing security module a namespace can have ACLs defined. Write access granted to a namespace will permit table creation for the given namespace. This provides tenants their own domain of administration within the hbase cluster.

Quota

Given the level of autonomy namespaces will provide tenants. Some level of control is required to insure that shared resources are allocated fairly. As a first step we only intend to limit the number of tables and regions a given namespace may contain.

Region Server Group

A namespace may optionally have a default region server group (see HBASE-6721). If defined tables created will automatically be members of a namespace's region server group. A namespace can reference only one region server group after which can no longer be referenced by other namespaces. This can only be set during namespace creation.

Predefined Namespaces

default - everything that does not have a namespace

system - system tables:-ROOT-, .META., ACLs. System tables are guaranteed to be loaded prior to user tables.

Namespace notation

To keep things simple, namespaces will be embedded as part of an hbase table name:

<namespace>.<table>

This command will create a table 'foo' in namespace 'myspace':

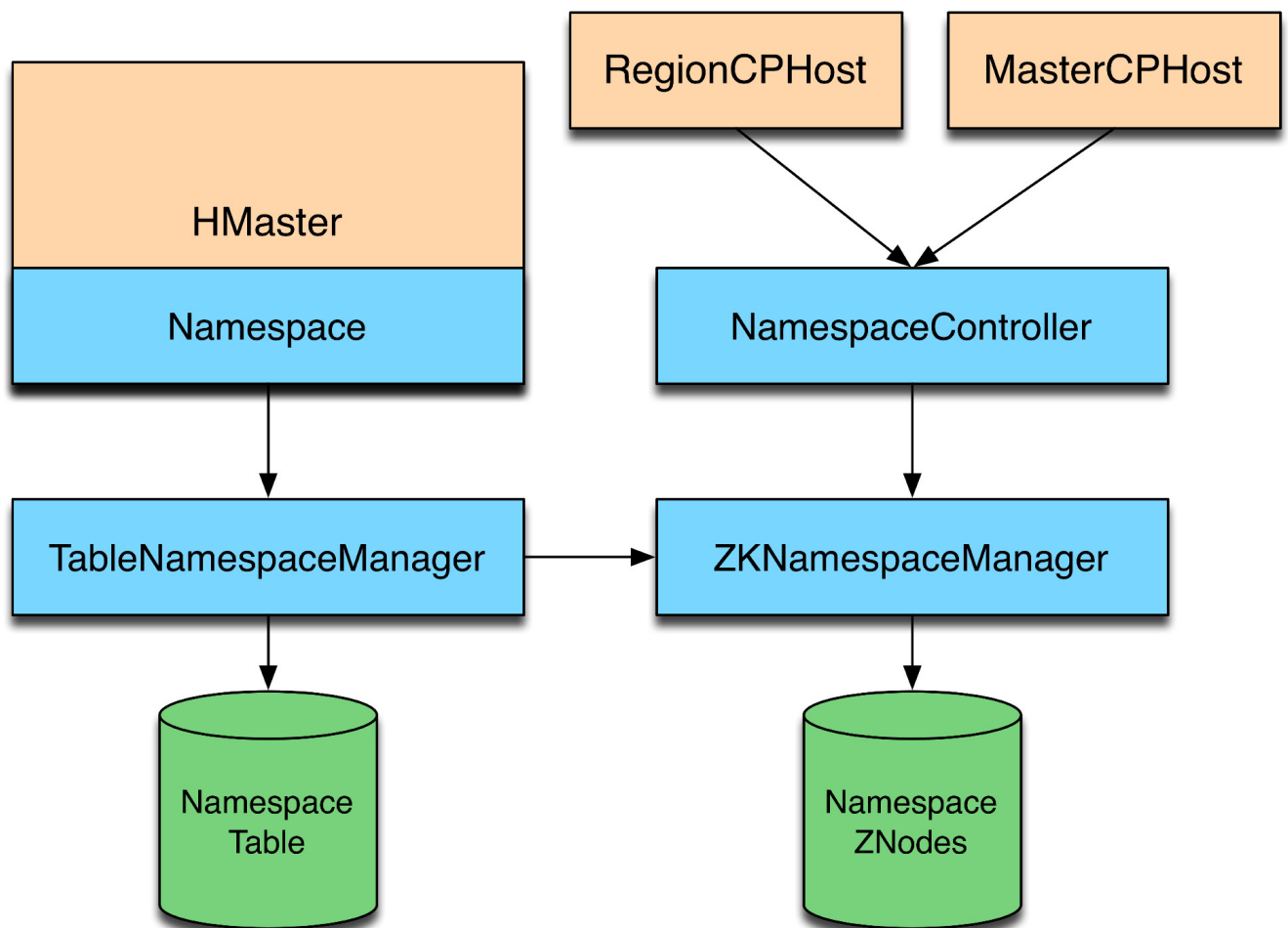
```
> create 'myspace.foo','f1'
```

*hierarchical namespaces won't be supported

CLI Syntax

```
namespace_create '<namespace>', {  
  'hbase.rsgroup.name'=>'<group_name>',  
  'hbase.namespace.quota.maxtables'=>'<count>',  
  'hbase.namespace.quota.maxregions'=>'<count>'}  
namespace_alter '<namespace>', {  
  'hbase.namespace.quota.maxtables'=>'<count>',  
  'hbase.namespace.quota.maxregions'=>'<count>'}  
namespace_drop <name>  
namespace_list  
namespace_get <namespace>  
grant '<user/group>' '<privilege>' '@'<namespace>  
revoke '<user/group>' '@'<namespace>
```

Design



The base namespace apis reside on the master with HBaseAdmin as the client. Namespace metadata is persisted in a HBase table and managed via the TableNamespaceManager helper class.

Namespace table updates are also forwarded to ZKNamespaceManager thus mirroring the meta information in zookeeper. Namespace information stored in zookeeper is read by the NamespaceController coprocessor running on region servers to correctly enforce quota restrictions.

Other changes:

- A TableName class has been added as an POJO of a fully qualified table name.
- Manual table path construction all over the source code has been replaced to use FSUtils.getTableDir

- HDFS table path is now of the form <ROOT>/data/<NAMESPACE>/<TABLE>
- Catalog table has been renamed to 'system.meta'
- Changes to the security module to support namespace level ACLs
 - Namespaces are prefixed by an '@' sign for grant and revoke commands