Level Based Storage Quotas

This is a design for storage quota of OpenStack Swift. An metadata named “x-account-meta-quota” which was stored in account database and set by the reseller administrator defines the quota level of account. An middleware named Quota reads the quota data from its configuration file and compares every update request with the usage and quota-level information requested from proxy. This middleware also have the precise or non-precise mode to deal with the performance and accuracy.

Quota Data

Quota data is an json string which described the quota of different quota level. For example:

```
    {
        "container_count": {
            "default": 5,
            "L1": 10,
            "L2": 25
        },
        "object_count": {
            "default": 200000,
            "L1": 500000,
            "L2": 1000000
        },
        "container_usage": {
            "default": 2147483648,
            "L1": 10737418240,
            "L2": 53687091200
        }
    }
```

There are three kinds of quota data. The “container_count” means the max number of containers in an account; The “object_count” means the max number of objects in a container; The “container_usage” means the max bytes of usage in a container.

The “default”, “L1”, “L2” are the quota-level. If the “x-account-meta-quota: L1” is set in account_A’s database, the max number of containers of account_A is 10, the max number of objects in account_A’s container is 500000, the max size of account_A’s container is 10 Gigabytes. If there is no “x-account-meta-quota” set in account_B, the account_B will have the quota data with default level.

The json format of quota data was stored in the configuration file of the Quota middleware. So, we should restart the proxy-server after any change of quota data.

Quota Level

Quota-Level is an metadata named “x-account-meta-quota” of account. For the security reason, only the reseller administrator can modify this metadata. So, there will be some modification in tempauth and keystoneauth.

Handling Request

PUT Container

- To get quota-level and container_count from memcached. If memcached is empty, to get them from proxy and cached in memcached.
- To compare the container_count + 1 with the quota_data.container_count[quota_level]. If the container count is over quota, the client will receive HTTP 403 response.
- If precise_mode is true, delete the account’s memcache.

PUT Object

- To get quota-level, container_usage, object_count from memcached. If memcached is empty, to get them from proxy and cached in memcached.
- To compare the current container_usage with the quota_data.container_usage[quota_level]. If over quoted, return forbiden.
- To compare the container_usage + content_length with the quota_data.container_usage[quota_level]. If the container usage is over quota, the client will receive HTTP 403 response.
- To get quota-level and container_count from memcached. If memcached is empty, to get them from proxy and cached in memcached.
- To compare the object_count + 1 with the quota_data.object_count[quota_level]. If the container count is over quota, the client will receive HTTP 403 response.
- If precise_mode is true, delete the container’s memcache.

DELETE Container

- If the prescise_mode is true, delete the memcache of account;
- If the prescise_mode is false, do nothing;

DELETE Object

- If the prescise_mode is true, delete the memcache of container;
- If the prescise_mode is false, do nothing;