USING THE PG-UPMAP

Starting in Luminous v12.2.z there is a new *pg-upmap* exception table in the OSDMap that allows the cluster to explicitly map specific PGs to specific OSDs. This allows the cluster to fine-tune the data distribution to, in most cases, perfectly distributed PGs across OSDs.

The key caveat to this new mechanism is that it requires that all clients understand the new *pg-upmap* structure in the OSDMap.

ENABLING

To allow use of the feature, you must tell the cluster that it only needs to support luminous (and newer) clients with:

```
ceph osd set-require-min-compat-client luminous
```

This command will fail if any pre-luminous clients or daemons are connected to the monitors. You can see what client versions are in use with:

```
ceph features
```

A WORD OF CAUTION

This is a new feature and not very user friendly. At the time of this writing we are working on a new *balancer* module for ceph-mgr that will eventually do all of this automatically.

Until then,

OFFLINE OPTIMIZATION

Upmap entries are updated with an offline optimizer built into `osdmaptool`.

1. Grab the latest copy of your osdmap:

   ```
   ceph osd getmap -o om
   ```

2. Run the optimizer:

   ```
   osdmaptool om --upmap out.txt [--upmap-pool <pool>] [--upmap-max <max-count>] [--upmap-de
   ```

   It is highly recommended that optimization be done for each pool individually, or for sets of similarly-utilized pools. You can specify the `--upmap-pool` option multiple times. “Similar pools” means pools that are mapped to the same devices and store the same kind of data (e.g., RBD image pools, yes; RGW index pool and RGW data pool, no).

   The `max-count` value is the maximum number of upmap entries to identify in the run. The default is 100, but you may want to make this a smaller number so that the tool completes more quickly (but does less work). If it cannot find any additional changes to make it will stop early (i.e., when the pool distribution is perfect).

   The `max-deviation` value defaults to *.01* (i.e., 1%). If an OSD utilization varies from the average by less than this amount it will be considered perfect.

3. The proposed changes are written to the output file `out.txt` in the example above. These are normal ceph CLI commands that can be run to apply the changes to the cluster. This can be done with:

   ```
   source out.txt
   ```

The above steps can be repeated as many times as necessary to achieve a perfect distribution of PGs for each set of pools.

You can see some (gory) details about what the tool is doing by passing `--debug-osd 10` to `osdmaptool`.