CRUSHTOOL – CRUSH MAP MANIPULATION TOOL

SYNOPSIS

crushtool (-d *map* | -c *map.txt* | –build *numosds layer1 ...*) [-o *outfile* [–clobber]]

DESCRIPTION

crushtool is a utility that lets you create, compile, and decompile CRUSH map files.

CRUSH is a pseudo-random data distribution algorithm that efficiently maps input values (typically data objects) across a heterogeneous, hierarchically structured device map. The algorithm was originally described in detail in the following paper (although it has evolved some since then):

> http://www.ssrc.ucsc.edu/Papers/weil-sc06.pdf

The tool has four modes of operation.

`-c` `map.txt`
: will compile a plaintext map.txt into a binary map file.

`-d` `map`
: will take the compiled map and decompile it into a plaintext source file, suitable for editing.

`--build` `--num_osds {num-osds} layer1 ...`
: will create a relatively generic map with the given layer structure. See below for examples.

`--test` `...`
`will` `perform a dry run of a CRUSH mapping for a range of input object`
`names`, **`see`** `crushtool --help for more information.`

OPTIONS

`-o` `outfile`
: will specify the output file.

`--clobber`
: will allow the tool to overwrite an existing outfile (it will normally refuse).

BUILDING A MAP

The build mode will generate relatively generic hierarchical maps. The first argument simply specifies the number of devices (leaves) in the CRUSH hierarchy. Each layer describes how the layer (or raw devices) preceding it should be grouped.

Each layer consists of:

```
name ( uniform | list | tree | straw ) size
```

The first element is the name for the elements in the layer (e.g. “rack”). Each element’s name will be append a number to the provided name.

The second component is the type of CRUSH bucket.

The third component is the maximum size of the bucket. If the size is 0, a single bucket will be generated that includes everything in the preceding layer.

EXAMPLE

Suppose we have 128 devices, each grouped into shelves with 4 devices each, and 8 shelves per rack. We could create a three level hierarchy with:

```
crushtool --build 128 shelf uniform 4 rack straw 8 root straw 0 -o map
```

To adjust the default (generic) mapping rules, we can run:

```
# decompile
crushtool -d map -o map.txt

# edit
vi map.txt

# recompile
crushtool -c map.txt -o map
```

AVAILABILITY

crushtool is part of the Ceph distributed file system. Please refer to the Ceph documentation at http://ceph.com/docs for more information.

SEE ALSO

ceph(8), osdmaptool(8), mkcephfs(8)