MEMORY PROFILING

Ceph MON, OSD and MDS can generate heap profiles using `tcmalloc`. To generate heap profiles, ensure you have `google-perftools` installed:

```
sudo apt-get install google-perftools
```

The profiler dumps output to your `log file` directory (i.e., `/var/log/ceph`). See Logging and Debugging for details. To view the profiler logs with Google's performance tools, execute the following:

```
google-pprof --text {path-to-daemon}  {log-path/filename}
```

For example:

```
$ ceph tell osd.0 heap start_profiler
$ ceph tell osd.0 heap dump
osd.0 tcmalloc heap stats:------------------------------------------------
MALLOC:        2632288 (    2.5 MiB) Bytes in use by application
MALLOC: +       499712 (    0.5 MiB) Bytes in page heap freelist
MALLOC: +       543800 (    0.5 MiB) Bytes in central cache freelist
MALLOC: +       327680 (    0.3 MiB) Bytes in transfer cache freelist
MALLOC: +      1239400 (    1.2 MiB) Bytes in thread cache freelists
MALLOC: +      1142936 (    1.1 MiB) Bytes in malloc metadata
MALLOC:   ------------
MALLOC: =      6385816 (    6.1 MiB) Actual memory used (physical + swap)
MALLOC: +            0 (    0.0 MiB) Bytes released to OS (aka unmapped)
MALLOC:   ------------
MALLOC: =      6385816 (    6.1 MiB) Virtual address space used
MALLOC:
MALLOC:            231              Spans in use
MALLOC:             56              Thread heaps in use
MALLOC:           8192              Tcmalloc page size
------------------------------------------------
Call ReleaseFreeMemory() to release freelist memory to the OS (via madvise()).
Bytes released to the OS take up virtual address space but no physical memory.
$ google-pprof --text \
              /usr/bin/ceph-osd  \
              /var/log/ceph/ceph-osd.0.profile.0001.heap
 Total: 3.7 MB
 1.9  51.1%  51.1%      1.9  51.1% ceph::log::Log::create_entry
 1.8  47.3%  98.4%      1.8  47.3% std::string::_Rep::_S_create
 0.0   0.4%  98.9%      0.0   0.6% SimpleMessenger::add_accept_pipe
 0.0   0.4%  99.2%      0.0   0.6% decode_message
 ...
```

Another heap dump on the same daemon will add another file. It is convenient to compare to a previous heap dump to show what has grown in the interval. For instance:

```
$ google-pprof --text --base out/osd.0.profile.0001.heap \
      ceph-osd out/osd.0.profile.0003.heap
 Total: 0.2 MB
 0.1  50.3%  50.3%      0.1  50.3% ceph::log::Log::create_entry
 0.1  46.6%  96.8%      0.1  46.6% std::string::_Rep::_S_create
 0.0   0.9%  97.7%      0.0  26.1% ReplicatedPG::do_op
 0.0   0.8%  98.5%      0.0   0.8% __gnu_cxx::new_allocator::allocate
```

Refer to Google Heap Profiler for additional details.

Once you have the heap profiler installed, start your cluster and begin using the heap profiler. You may enable or disable the heap profiler at runtime, or ensure that it runs continuously. For the following commandline usage, replace `{daemon-type}` with `mon`, `osd` or `mds`, and replace `{daemon-id}` with the OSD number or the MON or MDS id.

STARTING THE PROFILER

To start the heap profiler, execute the following:

```
ceph tell {daemon-type}.{daemon-id} heap start_profiler
```

For example:

```
ceph tell osd.1 heap start_profiler
```

Alternatively the profile can be started when the daemon starts running if the `CEPH_HEAP_PROFILER_INIT=true` variable is found in the environment.

PRINTING STATS

To print out statistics, execute the following:

```
ceph  tell {daemon-type}.{daemon-id} heap stats
```

For example:

```
ceph tell osd.0 heap stats
```

Note: Printing stats does not require the profiler to be running and does not dump the heap allocation information to a file.

DUMPING HEAP INFORMATION

To dump heap information, execute the following:

```
ceph tell {daemon-type}.{daemon-id} heap dump
```

For example:

```
ceph tell mds.a heap dump
```

Note: Dumping heap information only works when the profiler is running.

RELEASING MEMORY

To release memory that `tcmalloc` has allocated but which is not being used by the Ceph daemon itself, execute the following:

```
ceph tell {daemon-type}{daemon-id} heap release
```

For example:

```
ceph tell osd.2 heap release
```

STOPPING THE PROFILER

To stop the heap profiler, execute the following:

```
ceph tell {daemon-type}.{daemon-id} heap stop_profiler
```

For example:

```
ceph tell osd.0 heap stop_profiler
```