USING PERF

Top:

```
sudo perf top -p `pidof ceph-osd`
```

To capture some data with call graphs:

```
sudo perf record -p `pidof ceph-osd` -F 99 --call-graph dwarf -- sleep 60
```

To view by caller (where you can see what each top function calls):

```
sudo perf report --call-graph caller
```

To view by callee (where you can see who calls each top function):

```
sudo perf report --call-graph callee
```

note: If the caller/callee views look the same you may be suffering from a kernel bug; upgrade to 4.8 or later.

COMMON ISSUES

Ceph use *RelWithDebInfo* as its default *CMAKE_BUILD_TYPE*. Hence *-O2 -g* is used to compile the tree in this case. And the *-O2* optimization level enables *-fomit-frame-pointer* by default. But this prevents stack profilers from accessing the complete stack information. So one can disable this option when launching *cmake*

```
cmake -DCMAKE_CXX_FLAGS="-fno-omit-frame-pointer"
```

or when building the tree:

```
make CMAKE_CXX_FLAGS="-fno-omit-frame-pointer"
```

FLAMEGRAPHS

First, get things set up:

```
cd ~/src
git clone https://github.com/brendangregg/FlameGraph
```

Run ceph, then record some perf data:

```
sudo perf record -p `pidof ceph-osd` -F 99 --call-graph dwarf -- sleep 60
```

Then generate the flamegraph:

```
sudo perf script | ~/src/FlameGraph/stackcollapse-perf.pl > /tmp/folded
~/src/FlameGraph/flamegraph.pl /tmp/folded > /tmp/perf.svg
firefox /tmp/perf.svg
```