PERF HISTOGRAMS

The perf histograms build on perf counters infrastructure. Histograms are built for a number of counters and simplify gathering data on which groups of counter values occur most often over time. Perf histograms are currently unsigned 64-bit integer counters, so they're mostly useful for time and sizes. Data dumped by perf histogram can then be feed into other analysis tools/scripts.

ACCESS

The perf histogram data are accessed via the admin socket. For example:

```
ceph daemon osd.0 perf histogram schema
ceph daemon osd.0 perf histogram dump
```

COLLECTIONS

The histograms are grouped into named collections, normally representing a subsystem or an instance of a subsystem. For example, the internal `throttle` mechanism reports statistics on how it is throttling, and each instance is named something like:

```
op_r_latency_out_bytes_histogram
op_rw_latency_in_bytes_histogram
op_rw_latency_out_bytes_histogram
...
```

SCHEMA

The `perf histogram schema` command dumps a json description of which values are available, and what their type is. Each named value as a `type` bitfield, with the 5-th bit always set and following bits defined.

bit	meaning
1	floating point value
2	unsigned 64-bit integer value
4	average (sum + count pair)
8	counter (vs gauge)

In other words, histogram of type "18" is a histogram of unsigned 64-bit integer values (16 + 2).

Here is an example of the schema output:

```
{
    "AsyncMessenger::Worker-0": {},
    "AsyncMessenger::Worker-1": {},
    "AsyncMessenger::Worker-2": {},
    "mutex-WBThrottle::lock": {},
    "objecter": {},
    "osd": {
        "op_r_latency_out_bytes_histogram": {
            "type": 18,
            "description": "Histogram of operation latency (including queue time) +   da  ta
            "nick": ""
        },
        "op_w_latency_in_bytes_histogram": {
            "type": 18,
            "description": "Histogram of operation latency (including queue time) +   da  ta
            "nick": ""
        },
        "op_rw_latency_in_bytes_histogram": {
            "type": 18,
            "description": "Histogram of rw operation latency (including queue time)   +   da
```

```
            "nick": ""
        },
        "op_rw_latency_out_bytes_histogram": {
            "type": 18,
            "description": "Histogram of rw operation latency (including queue time)   +   da
            "nick": ""
        }
    }
}
```

DUMP

The actual dump is similar to the schema, except that there are actual value groups. For example:

```
"osd": {
    "op_r_latency_out_bytes_histogram": {
        "axes": [
            {
                "name": "Latency (usec)",
                "min": 0,
                "quant_size": 100000,
                "buckets": 32,
                "scale_type": "log2",
                "ranges": [
                    {
                        "max": -1
                    },
                    {
                        "min": 0,
                        "max": 99999
                    },
                    {
                        "min": 100000,
                        "max": 199999
                    },
                    {
                        "min": 200000,
                        "max": 399999
                    },
                    {
                        "min": 400000,
                        "max": 799999
                    },
                    {
                        "min": 800000,
                        "max": 1599999
                    },
                    {
                        "min": 1600000,
                        "max": 3199999
                    },
                    {
                        "min": 3200000,
                        "max": 6399999
                    },
                    {
                        "min": 6400000,
                        "max": 12799999
                    },
                    {
                        "min": 12800000,
                        "max": 25599999
                    },
                    {
                        "min": 25600000,
                        "max": 51199999
                    },
                    {
                        "min": 51200000,
                        "max": 102399999
                    },
```

```
                {
                    "min": 102400000,
                    "max": 204799999
                },
                {
                    "min": 204800000,
                    "max": 409599999
                },
                {
                    "min": 409600000,
                    "max": 819199999
                },
                {
                    "min": 819200000,
                    "max": 1638399999
                },
                {
                    "min": 1638400000,
                    "max": 3276799999
                },
                {
                    "min": 3276800000,
                    "max": 6553599999
                },
                {
                    "min": 6553600000,
                    "max": 13107199999
                },
                {
                    "min": 13107200000,
                    "max": 26214399999
                },
                {
                    "min": 26214400000,
                    "max": 52428799999
                },
                {
                    "min": 52428800000,
                    "max": 104857599999
                },
                {
                    "min": 104857600000,
                    "max": 209715199999
                },
                {
                    "min": 209715200000,
                    "max": 419430399999
                },
                {
                    "min": 419430400000,
                    "max": 838860799999
                },
                {
                    "min": 838860800000,
                    "max": 1677721599999
                },
                {
                    "min": 1677721600000,
                    "max": 3355443199999
                },
                {
                    "min": 3355443200000,
                    "max": 6710886399999
                },
                {
                    "min": 6710886400000,
                    "max": 13421772799999
                },
                {
                    "min": 13421772800000,
                    "max": 26843545599999
                },
                {
                    "min": 26843545600000,
                    "max": 53687091199999
                },
```

```
            },
            {
                "min": 53687091200000
            }
        ]
    },
    {
        "name": "Request size (bytes)",
        "min": 0,
        "quant_size": 512,
        "buckets": 32,
        "scale_type": "log2",
        "ranges": [
            {
                "max": -1
            },
            {
                "min": 0,
                "max": 511
            },
            {
                "min": 512,
                "max": 1023
            },
            {
                "min": 1024,
                "max": 2047
            },
            {
                "min": 2048,
                "max": 4095
            },
            {
                "min": 4096,
                "max": 8191
            },
            {
                "min": 8192,
                "max": 16383
            },
            {
                "min": 16384,
                "max": 32767
            },
            {
                "min": 32768,
                "max": 65535
            },
            {
                "min": 65536,
                "max": 131071
            },
            {
                "min": 131072,
                "max": 262143
            },
            {
                "min": 262144,
                "max": 524287
            },
            {
                "min": 524288,
                "max": 1048575
            },
            {
                "min": 1048576,
                "max": 2097151
            },
            {
                "min": 2097152,
                "max": 4194303
            },
            {
                "min": 4194304,
                "max": 8388607
            },
```

```
                {
                    "min": 8388608,
                    "max": 16777215
                },
                {
                    "min": 16777216,
                    "max": 33554431
                },
                {
                    "min": 33554432,
                    "max": 67108863
                },
                {
                    "min": 67108864,
                    "max": 134217727
                },
                {
                    "min": 134217728,
                    "max": 268435455
                },
                {
                    "min": 268435456,
                    "max": 536870911
                },
                {
                    "min": 536870912,
                    "max": 1073741823
                },
                {
                    "min": 1073741824,
                    "max": 2147483647
                },
                {
                    "min": 2147483648,
                    "max": 4294967295
                },
                {
                    "min": 4294967296,
                    "max": 8589934591
                },
                {
                    "min": 8589934592,
                    "max": 17179869183
                },
                {
                    "min": 17179869184,
                    "max": 34359738367
                },
                {
                    "min": 34359738368,
                    "max": 68719476735
                },
                {
                    "min": 68719476736,
                    "max": 137438953471
                },
                {
                    "min": 137438953472,
                    "max": 274877906943
                },
                {
                    "min": 274877906944
                }
            ]
        }
    ],
    "values": [
        [
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
```

```
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0
            ],
            [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0
            ],
            [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
```

```
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0
            ],
            [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0
            ],
            [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
```

```
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
```

```
            ],
            [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0
            ],
            [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0
            ]
        ]
    }
},
```


This represents the 2d histogram, consisting of 9 history entrires and 32 value groups per each history entry. “Ranges”

element denote value bounds for each of value groups. “Buckets” denote amount of value groups (“buckets”), “Min” is a minimum accepted valaue, “quant_size” is quantization unit and “scale_type” is either “log2” (logarhitmic scale) or “linear” (linear scale). You can use histogram_dump.py tool (see src/tools/histogram_dump.py) for quick visualisation of existing histogram data.