

Add new metrics to PingER

Introduction

We needed to add several metrics to the aggregated pinger reports. These included:

- [The Mean Opportunity Score](#)
- [Alpha a directness of connection measure](#)
- Maximum RTT (as an aid for exposing [buffer bloat](#)).

Implementation

Hourly

We wrote a script [wrapper_for_hourly.pl](#) that calls [wrap_analyze_hourly.pl](#) to analyze the gathered raw ping data from the monitoring hosts and create the hourly aggregated data. The script enables one to select the metrics to be calculated, the time frame, the ping size and the host or site. the output is the aggregated hourly data

The output directory was of the form

```
/nfs/slac/g/net/pinger/pingerreports/new/hep/maximum_rtt/maximum_rtt-100-by-node-1999-11-19.txt.gz
```

Note the "new". This enabled us to create all the aggregated reports for the selected metrics, check them out and then move or copy them to the [regular file space](#), i.e. to:

```
/nfs/slac/g/net/pinger/pingerreports/hep/maximum_rtt/maximum_rtt-100-by-node-1999-11-19.txt.gz
```

This was typically run as a batch job (since it takes a long time to run) with the output shown [here](#).

Daily

We wrote another script [wrapper_for_daily.pl](#). This calls [wrap_analyze_daily.pl](#) to read and aggregate the hourly data and create daily data.

This is also run as a batch job with the output shown [here](#).

Pingtable.pl

We also added these metrics to the pingtable.pl form.

Batch jobs for wrapper daily and hourly

There is script called [batch.pl](#) placed at [/afs/slac.stanford.edu/package/pinger/analysis/batch.pl](#) to create batch jobs for wrapper_for_hourly.pl. One can modify the output command to run for wrapper_for_daily.pl

Run the following command

```
batch.pl -y 2002 -q xxl
```

where the -y option gives the year, and -q option gives the queue to run the job in. The out put is

```

cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-1 --
set_metric 3
cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-2 --
set_metric 3
cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-3 --
set_metric 3
cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-4 --
set_metric 3
cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-5 --
set_metric 3
cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-6 --
set_metric 3
cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-7 --
set_metric 3
cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-8 --
set_metric 3
cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-9 --
set_metric 3
cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-10 --
set_metric 3
cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-11 --
set_metric 3
cmd=/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-12 --
set_metric 3

```

so one can copy and modify the batch output commands as follows:

```

/usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_hourly.pl --year_limit 2002-1 --
set_metric 3 --size 100 --by by-node # for jan 2002 analysis

usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_daily.pl --year_limit 2002-1 --
set_metric 3 --size 100 --by by-node # for jan 2002 analysis

usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_daily.pl --year_limit 2002-1, 2002-6 --
set_metric 3 --size 100 --by by-node # for jan-june 2002 analysis

usr/local/bin/bsub -q xsl /afs/slac/package/pinger/analysis/wrapper_for_daily.pl --year_limit 2002 --set_metric
3 --size 100 --by by-node # for all months of 2002 analysis

```

After creating job commands ,submit it via machine noric or yakut (pinger is not licensed for running batch jobs) using the command line interface. See wrapper_for_daily or wrapper_for_hourly.pl help for more options.

Analyze-allmonths.pl and analyze-allyears.pl

[analyze-allmonths.pl](#) uses the daily aggregated data, e.g.

```

> zmore /nfs/slac/g/net/pinger/pingerreports/hep/minimum_rtt/minimum_rtt-100-by-site-1998-01.txt.gz
Jan01 Jan02 Jan03 Jan04 Jan05 Jan06 Jan07 Jan08 Jan09 Jan10 Jan11 Jan12 Jan13 Jan14 Jan15 Jan16 Jan17 Jan18
Jan19 Jan20 Jan21 Jan22 Jan23 Jan24 Jan25 Jan26 Jan27 Jan28 Jan29 Jan
30 Jan31
sgiserv.rmki.kfki.hu www.cern.ch . . 26.000 25.000 25.000 26.000 26.000 25.000 25.000 26.000 25.000 26.000
24.000 24.000 24.000 24.000 . . . . . 25.000 25.000 26.000 25.000
26.000 26.000 26.000 25.000 sgiserv.rmki.kfki.hu www.cern.ch
...

```

So once we have the daily we do not have to go back and reanalyze data for each year, since trsrontab runs analyze-allmonths.pl each month

[analyze-allyears.pl](#) uses the allmonths aggregated data, and runs on demand or once yearly from trsrontab, so as long as analyze-allmonths.pl has run analyze-allyears is fine.

So "all" that is needed is to modify analyze-allmonths.pl and analyze-allyears.pl to analyze and aggregate the extra metrics and then run the scripts once, no need for lots of batch jobs.