

Pinger Schema Queries

Ghulam's queries regarding Schema

1. There should be two main tables raw and data separately. Raw will have node id, ping sequence no, rrtts, packet sent, packet received similar to what is currently in raw flat files. It will then have multiple tables to store either yearly or 6 months data in each table. Similarly for analyzed data there would be multiple hourly tables, multiple monthly tables and so on. Each table will be similar to analyzed data as in pinger a flat file. In such a case it will be different from PerfSonar where everything is in one table and if we place everything what is in raw table into analyzed table, there will replicate data (like 4 columns min/max/avg rrtts and seqno) however it will be similar to PerfSonar.

Ghulam/Zafar/Sadia>

Having one table has a drawback on 32-bit systems. The size limit for a table is 4 GB. This can be overcome but it can create performance bottlenecks (for example during read operations for loading data on PingER webpage). The size of data will grow everyday. Some approximations to support this claim:

- 2.1 MB flat file per site * 65 monitoring sites = ~137 MB per day.
 - `-bash-4.1$ ls -lh /nfs/slac/g/net/pinger/pingerdata/hep/data/pinger.slac.stanford.edu/ping-2012-06-08.txt.gz`
 - `-rw-rw-r-- 1 pinger iepm 2.1M Jun 9 01:04 /nfs/slac/g/net/pinger/pingerdata/hep/data/pinger.slac.stanford.edu/ping-2012-06-08.txt.gz`
- 137 per day * 30 days = 4.1 GB per month
- This is a rough estimation for size of data table. Others such as host and meta-data tables were not yet considered.
- Possible solutions include dividing MySQL tables in terms of months, regions or weeks (to make it more scalable in case monitoring sites increase in future).
- To shard is also better for performance in future and ensures sustainability by design.
 - As the data increases, queries will take longer (especially for read operation for loading data onto PingER webpage).
 - Sharded tables mean data can be loaded in parallel using Perl threads.

2. The other query is timestamp. timestamp is unique key in pingerDB assigned by Ghulam. So

e.g

`pinger.slac.stanford.edu 134.79.240.30 www.eldjazair.net.dz 193.194.64.71 100 1208217601 10 10 217.063 218.753 223.276 0 1 2 3 4 5 6 7 8 9 219 219 217 218 218 218 217 223 217 218`

`pinger.slac.stanford.edu 134.79.240.30 www.eldjazair.net.dz 193.194.64.71 1000 1208217611 10 10 219.238 221.357 227.909 0 1 2 3 4 5 6 7 8 9 220 220 221 221 220 227 219 222 219 219`

For above two records, metaid would be 1 and timestamp as 1208217601 for both. Database won't allow to add two duplicate records. However it's not duplicate, it's for two different packet sizes.

What Ghulam suggested is to add another column as data_id with auto increment property. It will give each record a unique number.

What I believe could be done is we make packet size also unique key.

Ghulam/Zafar/Sadia>

Making "packet_size" a joint Primary Key should work. [Latest schema here.](#)

What is left to be done

1. Getdata.pl is currently saving raw pings after every hour rather than after every half an hour. Modification is required to store data for every half an hour means every ping (total 48 for one pair per day)

Zafar/Sadia>

To resolve the issue of "how to" save raw data, we discussed and agreed in our previous meeting that we shall continue storing the raw data in flat files. This is in addition to storing analyzed data in MySQL database. This has two advantages:

- We won't need to modify the data collection part, thus making the current problem simpler.
- We will have raw data in its original form in case anything breaks down. This ensures backup.

2. Once if schema is decided, data can be transferred from old flat files into table. But first need to decide what would be in raw data table and how to store previous 10 years raw and analyzed data.

10 years data would be in single raw and analyzed data or what would be the best way to disseminate it?

Zafar/Sadia>

We shall continue using flat files for raw data. We will shift to MySQL database for analyzed data. A Perl script can be used to transfer analyzed data from flat files to the database.

Zafar 6/19/2012.

The webpage also contains final schema. We believe this information should be good enough to tweak and deploy database oriented archive site while preserving raw data as flat files. If the people who will pursue this have doubts then me and Sadia can help them resolve those doubts. Moreover I believe this shouldn't take more than 1-2 months (from development/fine-tuning to deployment).