

Galore-DB Requirements

This document outlines the functional and non-functional requirements for Galore DB.
we can categorize all the requirements into

- Functional
- Performance
- Compatibility
- UI

Functional Requirements

- galore db should be available as one of the service through the server
- galore db should support data distribution across machines(machines could be of different platforms)
- galore db file format should be platform independent
- galore db files should be loosely coupled to each other to the maximum extent possible. To explain further, the db files can be completely independent of each other or they can all dependent on one directory file.
- the database size should always be a order of magnitude less than the original data.
- should support a caching mechanism that makes the recent data to be accessed faster.
- should always operate within the resource limits imposed by the end user like
 - memory limitation, specified by the user
 - disk size limitation, specified by the user
- should have a decent periodic notification(alerts) system showing info like
 - disk space usage
 - memory usage
 - disk overhead
 - cpu overhead
 - compression overhead
 - encryption overhead
 - compression percentage
 - encryption percentage
 - overall size of data on the disk
- external tools to do the consistency check of the database; report any errors in the data files and possible solutions(automatic/semi-automatic)
- support database aggregation (main use case adding two captures)
- support data partitioning (should have multiple partitioning techniques)
- should support data indexing
- should have a very simple interface for modules capture, replay, loadminer, result handler etc.,

- should support database compacting. this is required as the user after the capture would like to minimize the db by cutting its size. Now, we should be able to a better job specially knowing all the data ahead of the time.
- should be encapsulated very well in a nice wrapper, in case we want to change/re-implement the db again.
- the db should always support the old releases/old data file formats. If there is any changes to the db software, then the needed db tools for upgrade should also be developed as part of the release.
- initial version should support 1TB size database.

Performance Requirements

- should be faster than any of the conventional databases(this should be obvious, but we should make sure that we dont screw it up)
- should be scalable linearly by increasing any or all of the following resources
 - Memory
 - Multiple Disks or db across multiple machines
 - CPUs
- db should also be scalable by increasing the number of threads with in the limitations of the above resources
- decent degradation of performance when the resources are very limited.
- should be lock-free for obvious performance reasons.

Compatibility Requirements

- should have a tool to convert the data files into xml or json.
- should also be able to export the entire data to any conventional database as per the customer request.
- should have a parser to read the datafiles in java and python.

UI Requirements

- user should be able to see his/her entire data anytime and anywhere through the galore UI
- offline UI tools to see the data in the db.(this do not need the all the other modules to be around)
-