Proposed Database Schema

The primary keys for these tables are not shown; I will likely be implementing the database using the data modeling framework for whatever Web backend we wind up selecting, which will set up primary keys and table join relationships automatically.

Table specific notes:

patch: I propose that for this schema patches will be stored on the filesystem content-addressable by the sha256 hash of the patch file.

patchset: "branch" refers to corresponding repo (master -> dpdk, dpdk-next-net -> dpdk/next-net, etc.). When a patchset is initally identified from a group patches (before it is actually applied), the branch, commit_id, and tarball fields will be NULL. Tarballs will be stored as files named after patchset_id of patch 1 of the series (note that this is not necessarily the lowest patchworks id in the series depending on the order that patchworks “absorbed” the patches). Tests run against git master will have a “fake” patchset table entry with patchworks_id set to NULL, and “branch” and “commit_id” set accordingly, and the tarball will be named after the branch and commit_id fields.

test_run: Overall result for an entire test run. Tests are identified by their timestamp and the environment that they were run on. “Official” tests were those kicked off by the appropriate triggers in Jenkins against git master or publicly submitted patches. “Unofficial” tests are manually run via Jenkins or manually running the script.

test_run_result: Result entry for a patchset. A patchset passes if there is a result for every measurement which is either PASS or N/A.

environment: This should represent everything about where the test was run and a new environment needs to be created every time this changes (e.g., kernel or compiler update). I gathered the list of fields by looking at the existing performance reports on the DPDK website. This can be used for verification, to allow the test environment to be reproducible, and to ensure that all comparisons are within an identical setup. Custom field can be included via the environment_field table (which could be further split into a table which defines the field and a table which associates the field with a value for each environment, if we decided we wanted to go that far).

measurement: A single measurement which can be applied to any patchset. We can use values like (name: “BUILD”, higherIsBetter: TRUE, expectedValue: 1, deltaLimit: 0) to verify non-performance conditions, such as the build succeeding for the given environment.

Test Result Entity-Relationship Diagram

Last Update: 11/28/2017 2:49 PM by Patrick MacArthur