

Proposed Database Schema

The primary keys for these tables are not shown; the database has been implemented using Django and exposed via a REST API, so the primary keys are all unique small integers.

Table specific notes:

patch: Caches some of the metadata from the Patchwork site.

patchset: This tracks the patchsets as a whole. `expected_patch_count` is used to tell when all patches have been received in Patchwork and this database and the patchset can be processed.

tarball: Once a tarball is generated for a patchset, it gets an entry in this table. "branch" refers to corresponding repo (master -> dpdk, dpdk-next-net -> dpdk/next-net, etc.). Tarballs will be stored as Jenkins artifacts; filename is the URL of the artifact in Jenkins.

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. **Parameter** is entirely contained within a measurement and is used to give input parameters such as frame size, descriptor count, etc. specific to a particular measurement.

Test Result Entity-Relationship Diagram

Last Update: 2/1/2018 2:37 PM by Patrick MacArthur

