Went well

- No negative progress in utests
- ?No negative progress in dtests?

Went poorly

- utests didn't make it to passing
- >3 months in progress on test debt is slow
- >3 months in Cassci reports failures
- 8099 broke trunk

Changes

- Start including test debt into releases
- Quarantine tests that fail or are flapping

Discussion

utests didn't make it to passing

I think we have a process geared towards long term releases that tends to bound quick turn around.

With one month between a release and the bug fix release turn around time for a bug fix is too slow. If you sum the time it takes for an issue to be reported, fixed, and go through multiple rounds of review (and potentially scope creep) there is no way a month is enough. It seems to me like a better aim would be to have the bug fix release and feature release be on the same two month cycle, and offset if that is desirable.

In >3 months progress on test debt backlog is slow

I think we are seeing success preventing the growth of test debt. A chunk of that is in avoiding test noise and we are iterating via retrospective and code review on releasing things fully baked the first time out.

Test debt has not been scheduled into releases the same way features are. I asked Jake (who looks like the product owner for releases) to start including some amount of test debt in each release. Jake has also stated that he sees more value in the dtests then the utests so those are likely to get priority.

Not so much as to make progress, but to test out the process of selecting things and having them get done in a release. The story continues to be be that we are focused on getting 8099 and 3.0 done.

It does seem to me that the only way to get 8099 fully baked is going to involve working off some kinds of test debt. Primarily creating/updating tests to be more thorough/methodical since we can’t take for granted many things that worked in the past continue to work.

>3 months in Cassci reports failures all the time
As part of working test debt into the release process Ryan has OKed quarantining failing utests and dtests so it is obvious on developer branches and trunk when new regressions appear. We will produce a list of JIRAs for Jake to incorporate into releases.

This is not going to include regressions from the 8099 merge.

Per our definition of done new regressions are not allowed and this will make the line clearer when you are deciding whether to merge.

8099 broke trunk
I know we had to merge at some point because of the rebase pain. My only question is did our process for dealing with this kind of thing work?

I know work is being done because I can see things improving which is great and I also see several recent JIRAs that look related. What I can't tell when/if all of them will be addressed and which are going to be dismissed as "not caused by 8099 merge." That concern only exists because the tests weren't passing before 8099 got there.

CASSANDRA-8502 - Static columns returning null for pages after first

> Testing of paging with static columns was not thorough or systematic. The tests have since been improved here: https://github.com/thobbs/cassandra-dtest/blob/c0e9d0abbdc058b6e48b47972eae927059ceb04f/paging_test.py#L649-L877. In general, we need a more systematic approach to testing all possible CQL queries, as discussed above.

CASSANDRA-9335 - JVM_EXTRA_OPTS not getting picked up by windows startup environment

> I chalk this one up to 1: a feature that's rarely, if ever, used on Windows (been this way for almost a year) and 2: lack of testing on

our launch scripts. I don't believe the dividends justify the effort to beef up / setup testing specifically for all the permutations of our launch scripts. This snuck through when I did a full rewrite/overhaul of the Windows launch scripts and I expect future revisions to be smaller, more incremental, and more easily testable.

CASSANDRA-9532 introduced regressions leading to errors on certain CQL queries.

This made it into 2.1.7 and was fixed by CASSANDRA-9636, which will be in 2.1.8. The root cause was that certain types of queries were overlooked when new tests were being added, leading to incomplete coverage. The coverage was improved in 9636 but as mentioned above, a systematic approach to testing CQL queries is the real solution.

CASSANDRA-8940 Inconsistent select count and select distinct

The problem with that ticket was that the issue could only be reproduced with around 500 000 rows. So we did not created a DTest for it as it would have taken too much time.

CASSANDRA-9669 Commit Log Replay is Broken

This is a scary fundamental bug with commit log replay. I very much hope a kitchen sink test would catch something like this, but so should CASSANDRA-9162. In fact, so should unit tests.

CASSANDRA-9656 Strong Circular Reference Leaks

Whilst this did not itself affect correctness of the system, it did negatively impact correctness restoring behaviours, i.e. leak detection did not work in many scenarios. This has both been fixed, and detection of circular-references to leak-detection monitored objects has been introduced. So errors should appear in log files if this occurs.

CASSANDRA-9637 selectAndReference blocks during compaction

Debug logging has been introduced to catch these events in-the-act and help report them. As with all such emergent properties, kitchen sink testing is the best way to induce them, and since we *detect and report* them, inducing them is sufficient to catch and fix them.

CASSANDRA-9388 Truncate can refer to freed memory (returning corrupted responses or segfault)

This is a system we're never likely to stress sufficiently during kitchen sink tests. So perhaps we should have a burn test for operations like truncation?