Database - Support #7058

get the FWD fork of the H2 code base under version control

01/20/2023 12:31 PM - Eric Faulhaber

Status:	WIP		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Alexandru Lungu		**% Done:**	80%
Category:			**Estimated time:**	0.00 hour
Target version:				
billable:	No		**case_num:**	
vendor_id:	GCD		**version:**	

Description

Related issues:	
Related to Database - Feature #6928: H2 UPDATE performance improvement	**Closed**
Related to Database - Feature #6582: implement multi-table AdaptiveQuery	**WIP**

History

#1 - 01/20/2023 12:37 PM - Eric Faulhaber

We have entered a new phase of H2 performance improvement work, and there is already a set of changes we have been managing for H2. Since we are stuck for the time being on the deprecated PageStore engine (for performance reasons), and it is probable that many of our performance improvements will be implemented in that area of the code, it is unlikely many of our changes will be accepted back into the main project. It is time to get our fork of the H2 code base under internal version control.

Let's start with rev 6 as our initial trunk, put revs 7 and 8 in their own discrete branches, and plan to merge the rev 8 branch to trunk after some wider testing. We will need a wiki page with instructions specific to the build of our fork.

Should we use the same branch naming convention (task number + ascending letters) as we use in FWD? It probably makes sense to add something to the branch names, so that that there is a clear distinction between H2 branches and FWD branches. This will avoid confusion in discussions.

#2 - 01/20/2023 12:44 PM - Eric Faulhaber

- Related to Feature #6928: H2 UPDATE performance improvement added

#3 - 01/20/2023 12:45 PM - Eric Faulhaber

- Related to Feature #6582: implement multi-table AdaptiveQuery added

#4 - 01/23/2023 03:17 PM - Greg Shah

> Should we use the same branch naming convention (task number + ascending letters) as we use in FWD? It probably makes sense to add something to the branch names, so that that there is a clear distinction between H2 branches and FWD branches. This will avoid confusion in discussions.

We definitely want to have the task-specific approach (task number + ascending letters). I agree, adding something to make the task branch name associated with h2 does make sense as well. Let's use <task_number><letter>_h2. This will be unambiguously H2 related while also linking back to te task in which the work is being done. We will assume a task branch without a project suffix is a FWD branch.

We'll need changes to create_task_branch.sh and archive_task_branch.sh on devsrv01 and the H2 repositories will need to be structured the same way that we use for FWD.

#5 - 01/30/2023 06:52 AM - Alexandru Lungu

Greg Shah wrote:

> Should we use the same branch naming convention (task number + ascending letters) as we use in FWD? It probably makes sense to add something to the branch names, so that that there is a clear distinction between H2 branches and FWD branches. This will avoid confusion in discussions.

> We definitely want to have the task-specific approach (task number + ascending letters). I agree, adding something to make the task branch name associated with h2 does make sense as well. Let's use <task_number><letter>_h2. This will be unambiguously H2 related while also linking back to te task in which the work is being done. We will assume a task branch without a project suffix is a FWD branch.

> We'll need changes to create_task_branch.sh and archive_task_branch.sh on devsrv01 and the H2 repositories will need to be structured the same way that we use for FWD.

I started to do the changes for the H2 project.

create_task_branch.sh and archive_task_branch.sh already have an argument named [project_path] which can be assigned to p2j_repo/fwd-h2 to make it work for our newly versioned H2. I find the branching of P2J the common case comparing to the H2 branching. Thus, we can explicitly state the project path when branching H2 and leave the default to P2J.

Otherwise, we either:

- create separate scripts for H2 (i.e. create_h2_task_branch.sh)
- add mandatory first parameter to the script (p2j or h2). However, this way we enforce the user to state p2j for each p2j branch.
- add an optional parameter to the script (p2j or h2). However, this would interfere with the [project_path] optional parameter. One is forced to state p2j of h2 even if the project path is custom.

I will not change the scripts for the moment and use the existing [project_path] to finish the version control of H2.

#6 - 01/30/2023 08:33 AM - Alexandru Lungu

- % Done changed from 0 to 80

- Status changed from New to WIP

Added version control for our H2 fork:

- redesigned code/p2j_repo/fwd-h2 to honor the active/dead/merged/trunk pattern. I think fwd-h2_experimental can be removed (?)
- set trunk to rev. 6: the one we use with FWD right now
- created two branches using create_task_branch.sh:
    - 6582a_h2 (the former rev. 7 including incremental sorting optimization)
    - 6928a_h2 (the former rev. 8 including H2 UPDATE statement optimization). I also added Radu's patch from [#6928-21](#) as a commit on this branch.
- any H2 changes can be tested against the H2 embedded tests (./build.sh test).
    - For 6582a_h2, I am aware that the planning is changed, so the "EXPLAIN" tests fail.
    - For 6928a_h2 and trunk, there are some failed tests (org.h2.test.unit.TestTools, org.h2.test.db.TestSequence and errors in testScript.sql found). However, most of them are related to the MVStore engine.
- any H2 changes can be profiled (./build.sh benchmark). However, most of the time is lost on comparing H2 with other engines. There is only a slim set of information we actually need like "Statements per second" or "GC overhead".

The former check-outs of fwd-h2 won't be bound anymore. Please redo the check-out of trunk, 6582a_h2, 6928a_h2 or create a new H2 task.

#7 - 01/30/2023 09:11 AM - Greg Shah

I'm glad the existing scripts worked.

#8 - 01/31/2023 05:09 AM - Alexandru Lungu

Greg, I've done only one documentation change to [Patching and Building Customized 3rd Party Libraries](#) (H2 section), mentioning that we use a custom fork of H2 internally. Keeping patches for each "latest" trunk revision of H2 doesn't feel feasible. Therefore, the distribution of our FWD-H2 project sources should be considered and documented there.

I intend to do a documentation for the branching system of FWD-H2, but most of the source management reference is (impressively detailed) in [Source Code Management](#). Should I refer to other wiki page or create a new one and refer to our existing "Source Code Management"?

#9 - 01/31/2023 09:08 AM - Greg Shah

See if you can just do some limited edits to Source Code Management to explain branching for FWD vs H2.

#10 - 02/01/2023 09:40 AM - Alexandru Lungu

Greg Shah wrote:

> See if you can just do some limited edits to Source Code Management to explain branching for FWD vs H2.

On it!

#11 - 02/07/2023 06:33 AM - Alexandru Lungu

I am planning to merge 7046_h2 to the FWD-H2 trunk after handling #7085-3 (test suite segregation). There are still some issues with regression testing and PageStore, so I will keep you updated here on the progress.

Once we reach a consistent testing suite in FWD-H2 trunk, I will test and integrate in the FWD-H2 trunk: 6995a_h2 (H2 NO-UNDO changes), 6928a_h2 (UPDATE improvements) and 6829a_h2 (avoid re-parsing all prepared statements). 6995a_h2 and 6928a_h2 were tested with large customer applications and show a real performance improvement. I am going to test and profile 6829a_h2 as well before merging.

If we have all these in trunk, we can do a new FWD-H2 build for FWD. Further, 6995a_h2 is considered only if we patch FWD with [#6995](#) (remove SQL Redo), so we will to integrate [#6995](#) after this whole process.

My work on [#7062](#) also has H2 changes and I will try to integrate them with this effort.

#12 - 02/08/2023 08:48 AM - Alexandru Lungu

Alexandru Lungu wrote:

> I am planning to merge 7046_h2 to the FWD-H2 trunk after handling #7085-3 (test suite segregation). There are still some issues with regression testing and PageStore, so I will keep you updated here on the progress.

Merged 7046_h2 to FWD-H2 after fixing regression tests:

- Integrated #7046 fix
- Integrated #7085 fix
- I've done only safe changes to the test-suite to allow the FWD-H2 trunk to pass them successfully (removed very specific edge cases and refactored some EXPLAIN tests to honor PageStore). Regression testing of H2 1.4.200 covered only a few features over PageStore. I added ./build.sh testMemory, ./build.sh testPersistent, ./build.sh testServer and ./build.sh testFWD to match our use-cases. These are documented in ./build.sh helper. It takes ~7min to run all testFWD.

#13 - 02/09/2023 10:53 AM - Alexandru Lungu

UPDATE: regression testing passed; merged 6995a_h2 to FWD-H2 trunk including the addition of a NO-UNDO keyword being natively supported in H2.

#14 - 02/14/2023 07:00 AM - Alexandru Lungu

UPDATE: regression testing passed; merged 6928a_h2 to FWD-H2 trunk including performance improvements for UPDATE statement, with in-memory and PageStore databases.

#15 - 02/14/2023 07:10 AM - Tomasz Domin

Will pre-built FWD-H2 revisions jars be uploaded into GCD repository ?

#16 - 02/15/2023 03:56 AM - Alexandru Lungu

UPDATE: regression testing passed; merged 6829a_h2 to FWD-H2 trunk including performance improvements regarding less recompiling of

prepared statements due to meta modifications.

> Will pre-built FWD-H2 revisions jars be uploaded into GCD repository ?

Yes. Once we have the pending branches merged into the FWD-H2 trunk, we will have a new build of FWD-H2 available through the GCD repository. At least this is what I expect; Greg makes the call.

#17 - 02/15/2023 06:08 AM - Greg Shah

For H2, Eric will push this when it is ready.

#18 - 02/15/2023 12:58 PM - Eric Faulhaber

Alexandru Lungu wrote:

> Once we have the pending branches merged into the FWD-H2 trunk, we will have a new build of FWD-H2 available through the GCD repository.

Alexandru, if there is any change to the build process, beyond just checking out the trunk FWD-H2 code and building with the current instructions, please update the 3rd party library wiki accordingly. Thanks.

#19 - 02/16/2023 06:18 AM - Alexandru Lungu

Eric Faulhaber wrote:

> Alexandru, if there is any change to the build process, beyond just checking out the trunk FWD-H2 code and building with the current instructions, please update the 3rd party library wiki accordingly. Thanks.

There is no change in the build process. I changed 3rd party library wiki to exclusively refer to our FWD-H2 approach.

#20 - 02/16/2023 09:11 AM - Alexandru Lungu

UPDATE: regression testing passed; merged 7062a_h2 to FWD-H2 trunk including support for direct access to internal H2 structures.

At this point, all the tested and profiled branches were merged into trunk. Eric, you can go ahead by building and uploading the new FWD-H2 version (from trunk/rev. 11).
7061a_h2 still needs intensive testing and optimization as it doesn't show much improvement in performance testing. Therefore, I don't suggest integrating it in the next FWD-H2 build.