Apache Drill Weekly hangout Notes

On the first Friday of every month at 8am Pacific time, we hold an online meeting where we report on the progress and also discuss current and upcoming issues and advancements, events, ideas.

This document holds historical minutes taken from meetups held up until 2017. These days, minutes are instead sent to the Drill mailing list.

Drill hangout (2017-6-27)

Link to meeting minutes

Drill hangout (2017-1-10)

Link to Meeting Minutes

Drill Hangout (2016-12-27)

Link to Meeting Minutes

Drill Hangout (2016-12-13)

Link to Meeting Minutes

Drill Hangout (2016-11-1)

Link to Meeting Minutes

Drill Hangout (2016-10-18)

Link to Meeting Minutes

Drill Hangout (2016-10-4)

Link to Meeting Minutes

Drill Hangout (2016-9-20)

Link to Meeting Minutes

Drill Hangout (2016-9-6)

Link to Meeting Minutes

Drill Hangout (2016-8-23)

Link to Meeting Minutes

Drill Hangout (2016-8-9)

Link to Meeting Minutes

Drill Hangout (2016-7-26)

Link to Meeting Minutes

Drill Hangout (2016-7-12)

Link to Meeting Minutes

Drill Hangout (2016-6-28)

Link to 6/28/16 Minutes

Drill Hangout (2016-6-14)

Link to Meeting Minutes

Drill Hangout (2016-5-31)

Link to 5/31/16 Minutes

Drill Hangout (2016-5-17)

Link to Meeting Minutes

Drill Hangout (2015-10-13)

(Thanks to notes by Daniel) -

Attendees: Andrew, Andries, Daniel, Jacques, Jinfeng, Kristine, Nathan, Parth, Venki, Zelaine

Topics:

Misc. status:
Parth is working on small fixes in the Parquet reader; is/will be having Jason review them.

Hadoop version:
Jacques brought up updating Drill to use a newer version of Hadoop (2.7.1) in order to support Amazon Web Services file system integration. (Relates to s3n to s3a protocol change.) New clients are supposed to be compatible with un-upgraded servers. MaprR needs to resolve Hadoop or Hadoop fork versions and related instructions.

Storage plug-in interface:
Jacques described plans to integrate planning aspects into the storage plug-in interface. (An example need is that a Drill plug-in for Phoenix can't incorporate Phoenix's secondary indexes into Drill planning.) Plans are to add hooks/extension point for more phases of planning, including pre-planning initialization.

Flatten:
There was discussion about how to address problems with FLATTEN. Ideally we need to implement Flatten as a table function. More discussion around the proposal needs to happen. In the future, Drill may move toward supporting standard SQL's UNNEST. One issue to resolve is how table functions related to "lateral joins." Jacques will file some JIRA reports. Also, Jacques mentioned that he is writing a "missing designs" document.

Calcite fork:
There were some updates and discussion about getting away from using the current Drill-specific tracking fork of Calcite. (Background: Drill has made a number of patches in its fork. Drill and Calcite haven't yet determined how to either add Drill-required functionality directly into Calcite or add extension points into Calcite. (An example bit of functionality is not expanding "*" in planning because of Drill's late schema binding.)) Jinfeng will look at recently added Calcite hooks to see if Drill's currently forked changes are implementable via those hooks.

Releases:
Regarding the Apache 1.2 release, Parth will sending a voting reminder.
Re future releases, there was a brief discussion about release scheduling. The target will be monthly released beginning on the **12th of the month**.

Drill Hangout 2015-10-06

Attendees: Aman, Andries, Daniel, Kris, Charlie, Julien, Jacques, Jason, Jinfeng, Matt, Parth, Sudheesh, Venki

1. Matt hitting issues with Information Schema queries against Hive. Will connect with Venki on Slack to resolve.
2. Julien reported that he's working on speeding up building and running tests, noting that build-time code generation runs twice and local Drillbits for testing take 3 second to shut down.
3. Parth mentioned an off-by-one bug in Parquet reading and that he will add more Parquet reading tests as part of the fix.
4. Aman reported a regression in performance while trying metadata caching with 400K files. This is being investigated.
5. Daniel, Jacques, and Sudeesh discussed issues underlying DRILL-2288, such as the ScanBatch.next() return value (IterOutcome) contract, handling empty JSON files, handling zero-row sources that still have schemas, how to limit the DRILL-2288 fix to

avoid needing to rework lots of downstream code, etc.

6. Sudheesh had various updates - Limit 0 and Limit 1 queries. Jacques suggestion to handle Limit 0 queries on schema aware systems to the planning phase. Perf tests on the RPC processing offloading seem to show higher memory consumption. This may simply be due to allowing more concurrent queries as result of the patch. Perf tests reveal issues on local data tunnel changes but these may be existing problems that are now showing up as a result of faster local data processing. Question to be resolved - should we merge these anyway?
7. Jason helping address some recent issues with flatten involving large number of repeated values.
8. We unanimously volunteered Sudheesh to work on the performance cluster.

Drill Hangout 2015-08-04

Attendees: Daniel, Khurram, Neeraja, Vicky, Kris, Aman, Parth, Andries, Jinfeng, Anas

- Insert and drop
    - read isolation during insert into, Aman suggested snapshot level
    - have to have some kind of lock manager
    - locking on the dot drill file
    - should this locking talk to other external programs working with Drill used by Drill?
    - Jason - why is this the lock necessary?
    - we want to merge schemas in a dot drill file, avoid gather schemas from a lot of separate files
    - insert feature will be broken into phases
    - this needs to handle schema changes to be consistent with the rest of Drill
- partition pruning is not working for some expressions
    - we will only fix for cast
    - Jinfeng thinks this should be easy enough
- handling unknown types in parquet or other external systems
    - should we fail actively, or should we give data back in varbinary
    - sould people have to wait for a release to handle new data types
    - storage plugin writers should have a clear idea about how to handle these cases
    - Jason will send a message to the list about this

- test framework
    - Rahul is working on publishing it to a public repository
    - this will include instructions on how to set up the tests on your own hardware

Jacques' view on isolation

Quick thought on insert isolation:

Let's just do a hidden directory and then rename. We can make Drill avoid reading hidden directories. No fancy work required.

With regards to Dot Drill, let's not turn this into a mini database. The complexities would be overwhelming. My recommendation is we constrain to additional metadata that cannot otherwise be divined. Beyond that, we're should use ephemeral files (similar to the parquet metadata cache where deleting doesn't impact logical outcome, may impact planning or performance). I would avoid mixing ephemeral and persistent data around the dot drill concept.

In general, if we want to store Drill's internal ephemeral metadata, lets have a discussion around the options. Also remember that not all Drill installations will use a distributed filesystem. As such, we need to think about these abstractions to support multiple types of storage systems.

Drill Hangout 2015-07-29

Attendees: Andries, Daniel, Hanifi, Jacques, Jason, Jinfeng, Khurram, Kristine, Mehant , Neeraja, Parth, Sudheesh (host)

Minutes based on notes from Sudeesh -

1) Jacques working on the following -

a) RPC changes - Sudheesh/Parth reported a regression in perf numbers which was unexpected. Tests are being rerun.

b) Apache log - format plugin.

c) Support for Double quote.

d) Allow JSON literals.

1) Parquet filter pushdown - Patch from Adam Gilmore is waiting review. This patch with conflict with Steven's work on metadata caching. Metadata caching needs to go in first.

2) JDBC storage plugin - Patch from Magnus. Parth to follow up to get updated code.

3) Discussion on Embedded types -

a) Two types of common problems are being hit -

1) Soft Schema change - Lots of initial nulls and then a type appears or the type changes to a type that can be promoted to the initial type. Drill assumes type to be nullable int if it cannot determine the type.Discussion on using nullable Varchar/Varbinary instead of nullable int. Suggestion was that we need to introduce some additional types -

i) Introduce a LATE binding type (type is not known).

ii) Introduce a NULL type - only null

iii) Schema sampling to determine schema- use for fast schema.

2) Hard Schema Change - A schema change that is not transitionable.

b) Open questions - How do we materialize to the user? How do clients expect to handle the schema change events. What does a BI tool like Tableau do if a new column is introduced. What is the expectation of a JDBC/ODBC application (what do the standards specify, if anything). Neeraja to follow up and specify.

c) Proposal to add support for embedded types where each value carries type information (covered in DRILL-3228) This requires a detailed design before we begin implementation.

4) Discussion on 'Insert into' (based on Mehant's post)

a) In general, the feature is expected to behave like in any database. Complications arise when the user choses to insert a different schema or partitions from the the original table.

b) Jacques's main concern regarding this: Do we want Drill to be flexible and be able to add columns and be able to not specify columns while inserting or do we want it to behave like a traditional Data Warehouse where we do ordinal matching and are strict about the number of columns being inserted into the target table.

c) We should validate the schema where we can (eg parquet), however we should start by validating metadata for queries and use that feature in Insert as opposed to building that in Insert.

d) If we allow insert into with a different schema and we cannot read the file, then that would be embarrassing.

e) If we are trying to solve a specific BI tool use case for inserts then we should explore going down the route of solving this specific use case, and treat the insert like CTAS today.

5) Discussion on 'Drop table'

a) Strict identification of table - Don't drop tables that Drill can't query.

b) Fail if there is a file that does not match.

c) If no impersonation is enabled then drop only drill owned tables.

Jacques' notes for #4 and #5 -

INSERT INTO

This functionality should be consistent how Drill views data and the world.

It seems like there are a number of missing foundational components that should be built before this could work "right".

First steps that should be done before INSERT INTO, EMBEDDED and DROP:

- Add a null or nullable any type to the execution flow. Don't materialize this until necessary (convert new field existence to soft schema change from current hard schema change).
- Add a INSERT INTO that works like CTAS. This is the simplest way to start and bumps many decisions to later. This will also solve the top two use cases: BI tool temporary table and advanced user use cases (but is a bit of sharp instrument)
- Implement "thorough table identification". Right now a directory of multiple types of files (potentially queryable and non-queryable)
- Add support for Parquet schema reading, merging and validation. Parquet has schema, Drill shouldn't expose as schemaless. This will set the groundwork for a number of types of validation around INSERT INTO, DROP, etc. (It will also require a full deconflicting between implicit casting behavior for the validator and the execution layer.)
- Start planning around "dot drill" (a.k.a. DRILL-3572). Many of things that need to supported to make these things work "like a database" require this.

Drill Hangout 2015-07-21

Participants: Jacques, Parth (scribe), Sudheesh, Hakim, Khurram, Aman, Jinfeng, Kristine, Sean

Feature list for Drill 1.2 was discussed. The following items were considered (disussion/ comments if any are summarized with each item):

1. Memory allocator improvements - Spillover from 1.1
2. Faster reading of Hive parquet tables - Spillover from 1.1.
3. Enhance/cleanup test framework & publish to community
   The dev team has a set of tests that are a requirement for committing. These tests should be made available so that community contributors can run them independently.
4. Additional window functions - Dev - NTILE, LEAD, LAG, FIRST_VALUE, LAST_VALUE

5. Faster metadata read for 1000s of Parquet file
6. Rowkey filter pushdown improvements
7. Support "Insert into <T> Select From"
   A longish discussion on this. (I might have missed some points). Concerns about how to maintain the previous table metadata, especially partition metadata. There was a discussion around allowing a table to have files where some files have different or no partitions from the partitions defined when the table was first created. Suggestion to incorporate the metadata in a .drill file.
   Some more clarity to be provided about the functionality. More discussion to continue in the JIRA.
8. Support "Drop table"
   A small discussion on some restrictions that will need to be imposed. In particular, not allow access to root (/) and also that we should probably validate that the table being dropped is, in fact, a table.
9. Security for the WEB UI
   Some considerations that we need to consider are whether we need to enable SSL and some form of authentication/authorization for the web UI. Also whether we need to consider the same for the REST API. A second question is whether we should include the need to limit access to workspaces defined in the web UI. One suggestion was whether we need to create workspaces similar to views (i.e defined in a .drill file) and then use the same access control mechanisms (i.e the one provided by the file system).
10. JDBC driver
11. Super bugs - flatten
    There was a discussion on whether we need to consider flatten to be an instance of a User Defined Table Function. The issues we see in flatten are related less to the flatten logic and more to handling edge cases of batch boundaries, and vectors. The idea behind supporting UDTFs would be to write the framework that handles the complexity of handling input and producing output and the UDTF itself would need to implement an input and an output operator. Flatten can then be reimplemented as a UDTF.
12. Super bugs - convert funcion/
13. Super bug - 2010: MergeJoin incorrect results. (Suggestion that the solution might be to go the UDTF way)
14. Super bug - DRILL-3121 Support interpreter based execution for hive partition pruning.

Hangout 6/30/15

Attendees: Jason, Venki, Hakim, Daniel, Kris, Andries, Vince Gonzalez

Hakim: Window functions updates in Drill 1.1. Available functions are: rank, dense_rank, percent_rank, cume_dist and row_number. Earlier versions of Drill window functions are disabled by default. After testing thoroughly in 1.1 windows functions are enabled by default.

Splunk has issues with JDBC driver. Andries is looking into the issues.

Kristine: Converting the docs to show examples and be more suitable for people using Drill on multi-node cluster rather on embedded mode. Current examples in docs mostly show using Drill in embedded mode and accessing local filesystem, giving the impression that Drill works in querying local files only. People are mostly interested in setting up and running Drill distributed mode. Kris is working on improving the docs to provide more help in that area.

Changing review process from review board to github pull requests. Jason is going to continue discussion on dev list.

Paul had a qn about scanning performance improvement. Jason answered: Auto partitioning takes care of generating partitioned files and partition pruning code automatically identifies the partitioned files and doing the partition pruning instead of just relying on filesystem organization of partitions like in Hive.

Hangout 1/21/14

Attendees: Jacques, Mehant, Jinfeng, Jason, Julian, Aman

patch out
jdbc thinning

Aman
- hash table and aggregation
  - should be done by end of week
- brainstorming with jacques to make it more usable for hash join

jason
- deciding how to implement setting options
  - do we want to define it as a physical operator?
  - should we send a single node plan through the optimizer?
- use a virtual table in cache, no operation for setting options
  - more general 'insert into table'
  - there is other metatdata that should be set elsewhere
  - implement information schema
    - information schema reader, another storage engine
    - reflect internal nature of system
    - make information schema writer
    - current do not support insert into in drill
    - sometimes more complicated to use your own database
      - can just use someone else's, like hSQl
      - otherwise bootstrap issues

- distributed backing store
  - some changes would require a restart
  - currently everything is in config files
  - everything requires a restart
  - some things that will be set at start even into future versions
    - amount of memory
  - levels: session, system, statement
    - existing session will inherit system at session start
    - system setting changes do not propagate down to existing settings
    - system character set in oracle cannot be changed after database start
  - jacques - on list of things to do
    - infinispan is better than hazelcast
      - switched to apache license
      - hazelcast pulls major features for commercialization
      - this supports major features hazelcast doesn't
      - was designed with this eventual transition in mind
      - still not a trivial transition
      - optiq wants to use infinispan
        - otiq can store Drill metadata as optiq tables
        - can just use sql to query it
        - jacques - sounds good, timing?
          - getting optiq to make a java collection to look like a table is simple
          - should be straightforward
          - wouldn't this involve a query plan?
          - you can store the plan if used a lot
            - with param binding
      - ORM
        - how fast can read infinispan data through optiq
          - with prepared plan, it would be very fast
          - otherwise we need to combine a bunch of in memory structures
          - you start with simple dictionaries
          - but then you want to put views on top of them
            - would have to rewrite this in java
            - jacques - we use data grid directly for other use cases
              - we wouldn't want to go through optiq for some of these uses
            - julian
              - using system that allows querying iterables
              - linq like, all type safe, relational not using sql
              - this is what optiq generates, throws typing out the window with JDBC result set
      - what about session options
        - should they be in cache as well

- options that affect operators running on other drillbits, should all of these options be sent as part of the plan
- are sessions going to be in distributed state anyway?
- sessions are not recoverable, query IDs are
  - only reason to recover session, want to know what happened
- put sessions in data-dictionary table, put it in the distributed cache
  - keep it there until it is cleaned, can check failed session with parameters x,y,z failed yesterday
- put it in header of plan
- should options be lists
- JSON literals in optiq?
  - UDF, parse JSON
  - can we add JSON as a literal without quotes?
  - read up on XML in sql 2003, there are wrenching changes of language to make it work
  - have a JSON UDF, single quoted string as param, can use double quotes inside
  - multi- line strings, no concatenation
- tried to add parenthesis around joins
  - after many hours, just gave up
  - sql parsing is really hard

julian
- case sensitivity
- quoting options
- lexer tripled in size because of redundant logic :p
  - after 20 line change to source
  - don't know of another engine that supports changing
-
  - something about how we index into maps?
  - there is a difference, index into map or table
  - look at how microsoft does it
- optiq
  - jdbc - was thick client
    - thin approach with avatica now
    - is there value in using avatica
    - how do we avoid jumping through hoops to make it work
      - any jdbc driver you write will have to do things
      - drill result set should be a listener
      - ordering makes sense for optiq, the situation happens to be different
      - result set should be able to see the statement it belongs to
        - otherwise thats a bug

        - jacques - the result set is created by a factory
        - if we change avatica, we have to chagne optiq to use modified versions
        - we don't support prepared statements, would be confusing to maintain
        - julian - create a prepared statement, even if its not used
            - even if it just represents that sql was parsed
            - don't want to hit server twice
            - single RPC to prepare and execute query
            - jacques - model does not support that right now
            - assume it is a solved problem, it should not require two round trips
        - julian - these can be solved, maybe not the way you think

- protobuf in source control?
    - issues with needing a specific version to run the build, decided to just put the generated files in git

Hangout 1/14/14

jacques:
- merged broadcast and others
- will be doing more merging
- updated RPC
- updated memory counting stuff a lot
    - shown memory leaks
    - hopefully squashed in day or two
    - then working on integrating optiq framework

Aman
- progress on hash aggregation
- hash table implementation
    - had talks with jacques about design

Jinfeng
- slq functions
    - postgres
    - math function, string function
    - log, exponential
    - decimal type need to be implemented based on our representation
    - pattern matching function
        - support like predicate
    - julian has suggested previously drawing on his implementations
        - he has done a lot of work on 'like'
        - look at how we can incorporate his work possibly

- if all else fails, trying to get it working with compilation copy and paste

Julian
- version 1 or optiq in April
  - so far hes just be incrementing number each month
- can read release notes for useful changes
- going to work on materialized views
- remote jdbc driver by April possibly
- first step is getting the SQL parsing on server
- then do communication between thin client and server
- julian - these could probably be done in parallel
- jacques - we should support a thrift endpoint
  - will make supporting other languages easier
  - protobuf doesn't have RPC in all languages
  - thrift is unidirectional
    - client has to use a pull model
  - julian - make sense to start with hive and mutate to what we need
    - needs are very different from JDBC driver
    - people will want both thrift and protobuf
    - thrift with callback might be an option
    - if people want bidirectional they can use protobuf
    - should be possible with current structure to make it pluggable
    - have to look more at hive protocol
      - based on thrift, impala is based on it
  - george
    - async streaming for JDBC?
      - most clients will not wait long for a response
        - fetch with timeout will be acceptable for most clients
    - desire for bidirectional?
      - map reduce job against results of drill query
      - can process data streaming out of drill
      - on server you have to buffer results
      - bidirectional, if query is waiting on single node and it dies, how do you tell a client to listen to another node
      - Drill, don't want to re-write data into row major on server
        - jdbc driver reorganizes it
      - julian - look at implementation of Cursor, zero copy read
      - jacques - two paths
        - drill solves jdbc for itself, then optiq does a generic one for other uses
        - other option is to do it once
      - take discussion offline

- or talk on mailing list, write proposal
- julian will see if he can work it into cursor

Timothy
- orc reader
  - unreleased version
  - vectorized reader
  - might just use that
  - otherwise he was copy and pasting code
  - starting with their snapshot
    - populate vectors with decoded data

Xiao
- new to drill
- working with George

Hangout 1/7/14

Mehant
- decimal status
  - pulled into working on other stuff
  - hopefully end of wednesday will have value holders VVs and functions
  - get it in without casting, we can come back to casting, as it will be a little involved
- date stuff
  - should be easier than decimal

Jacques
- RPC, changing multiplexing
  - get something out today
  - adds a third type of RPCS
  - right now
    - user/bit
    - bit is being divided in two
    - now support blocking queue, when downstream is not consuming
- will work on getting patches in
- optiq integration
  - new framework stuff
  - should make things cleaner/easier to understand
- to george
  - with optiq changes should get easier
  - george
    - wants to talk to julien about thinning JDBC
    - haven't explored changing protocol

- thrift is unidirectional protocol
- we could add a thrift driver, but this is why we chose not to use it at first
- how do clients work now?
  - everything is protobuf based
  - current client is java
  - protocol is simple, any language could implement client
  - client has to convert into record format from columnar
  - metadata?
    - working on it right now, hive integration
    - right now, cannot submit a prepared statement
    - don't know columns until read, can change during query
    - this should change in the next few weeks
- looking at building odbc driver
  - take offline for next steps
  - bigger issue, should protocol be record based

Aman
- hash aggregation
  - progress on hash table, based on proposed design
  - just finished the put/get and init
    - working on resizing and re-hashing
- aggregation
  - have not done codegen, trying to make work without code gen first
  - memory management

Jinfeng
- casting
  - on reviewboard
- scalar functions
  - looking at postres list

Tim
- congrats on committer status
- not working on hash join
  - waiting on off heap hash table
- orc file reader
  - can read stripes, can parse footers
  - interfaces are private, not liking what he has to do
  - he is just copying source
    - opened a HIVE jira, use the hive implementation
  - is he using vectorized format?
    - haven't looked at it

Jason
- optiq options
  - examples
    - explain plan logical/physical
- UserCLientConnection
  - constrained to RPC layer, may want to change this somehow
  - look at DrillConfig, see if we can use this
    - right now not modifying it, we can use it possibly for global options which will change during runtime
  - used to communicate with clients
  - add session object here
- global/session scope
  - how to branch path based on scope
  - global options should be in distributed cache

Impala
- YARN integration
- HDFS caching

Due to the holidays, Hangouts were cancelled for two weeks

Hangout December 17 2013

participants: jacques, jason, tim, Aman, Jinfeng, Abhiraj, Andrew, Steve McPherson, Steven Phillips

Jacques:
- backlog of review requests
  - anyone feel free to do reviews!!
  - Jacques - comfortable committing something reviewed by two community members
- parquet writer
  - thinking about internal representation of maps and repeated types
  - not totally conclusive
  - writer interface should be designed to handle this functionality
- deadlock in RPC
  - throttling in RPC
  - multiplex on single socket
  - separate sockets per fragments

Jinfeng
- implicit casting using function resolver from Yash
  - inject implicit casting whenever necessary
  - anytime arguments do not match
  - tested prototype yesterday, hopefully done this week

- comments from jacques on explicit cast

Aman
- design document for hash aggregation
    - including design for hash tables themselves
    - how to spill hash tables to disk
    - impact of distinct aggregations on design
    - send out later this week to mailing list

Tim
- amazon EMR, final script that is working
    - no distributed queries yet
    - multiple nodes connected to zoo keeper
- will give out link to install script
- hash aggregates and hash join
    - what is good for a hashtable?
    - options off the shelf, use locks everywhere
    - do not fit well with Drill
    - trying to learn code generator
    - meet with Aman to join efforts
- someone review broadcast sender
    - jacques - questioning if we are doing memory management right
        - should just get it in
        - we'll have to fix memory management everywhere once we have new strategy anyway
- hash table possible
    - use fastUtils
        - modify it to use bytebufs instead of on heap arrays
- question to jacques? how does code generation work
        - methods need to be implemented
        - methods bodies need to be inserted somewhere
        - simple scaler function- do setup, do eval
        - scalar function is added to setup, eval added to do_eval
        - mapping sets map function bodies to any function you want to
        - in join, you have 6-8 abstract methods, that need to be populated
        - apply multiple expression trees to each of the methods
        - the mapping set is how you map particular expressions to the functions

jason
- convert physical plan to optiq rels
- will update JIRA for reading select columns from parquet

steven phillips
- performance checks
    - will send out summary
- results are expected
    - need to improve query startup time
    - need to optimize sort

        - leaking memory in some cases
        - architecture looks sound, need to make some small optimizations
    - multi-phase aggregation
        - hive - specify update and merge functions
        - like map and reduce functions
        - tricky part is moving data between initial and final phases
        - how do we handle types between the phases

Abhiraj
- new to drill community
- looking to contribute
- can start with looking at outstanding bugs
- can take a look at the wiki of how to setup and build

Andrew
- avro storage engine
- basic avro engine working
- basic scans
- would like to put up for review
- put up on apache review, not using pull requests for Drill

Steve McPherson
- Drill bootstrap action
    - anyone tried it for setting up cluster?
    - tim trying right now
- might be useful for testing and running at scale
- Jacques, I didn't see that, did it hit the list?
- can launch a whole cluster
    - patch for fixing classpath
    - currently doing their own build
    - they can just pull form mainline after patch gets in

MapR QA starting to look at Drill

Apache Drill G+ hangout 2013-12-10 — Status

participants: Jason Altekruse, Jingfeng, Timothy Chen, Julian

Aman
- working on filed a JIRA for aggregate functions
  - sent out code review
- based on code generation for aggregate functions
  - hash aggregation work, has been talking to jacques
  - prototype in next few weeks

Jinfeng
- explicit casting
  - submitted review board request
  - working with Yash on implicit casting
  - working design for implementing implicit casting this week

John
- no updates

Tim
- Amazon on EMR
- trying to get boostrap action to work
- have to write a script that is hard to test
  - has to go back and forth with them
  - one interation has been sent recetnly that is waiting for response
- boardcast sender
  - pedning review request
- will work more on hash join next

Julian
- talked to Steve McFurson
  - said he knew Tim
  - he is intersted in Julians work on Drill Mondiran and Optiq
- went to MapR on thursday
- ODBC/JDBC driver
- breaking up Optiq
  - pull request on optiq from Jacques with API changes
  - Drill and Optiq are doing things in common
  - hoping API changes are going to allow sharing more functionality for Drill and Optiq
  - Drill got around virtual function calls
  - common sub-expression elimiation in Drill?
    - avoiding re-evaluating the same predicates/math functions
- Optiq is too monolithic
  - this is why we didn't use RexNode
- common sub expression elim
  - Rex Program
    - eliminates common sub expressions with project and filter
    - normalize expressions into common form
      - a+b = b+a
      - does not work for all operators JAva is particular about order of AND
    - constant folding

- convert const expression into simple constants
- Drill is strong in runtime
  - running generated code fast in execution

Jason
- need to look at parquet NPE

Apache Drill G+ hangout 2013-12-03 — Status

Notes missing. Anyone?

Apache Drill G+ hangout 2013-11-26 — Status

Participants: micheal (amazon), jacues, jin feng, amahn, mehant, steven, micheal, julian, parquet reader

- too many cpu cycles
- jason - push code for selecting columns!

Optiq
- first pass at relating our operator to optiq rels
- try to understand logic of larger classes
  - hive guys have similar requirement, using a subset of optiq
  - built a class for them called frameworks, basically a static method
    - provide it with code you want it to call when the environment is set up
    - rather than subclassing a bunch of things
    - current method is brittle, optiq is constantly changing
    - is frame work something in repo?
      - in master, called Frameworks
    - optiq prepare_____ is very complex
      - look at a better way to organize it
    - challenging scenarios is, we want to go from sql query to some intermediate
      - expand grammar to support other concepts in optiq
      - DDL operation
      - send it further down the path, concert to rel nodes
      - maybe convert to logical plan
      - maybe go to other rel nodes, start optimizing
      - take physical rel nodes, generate physical plan
      - want to drop out of optiq at various places

- take logical plan, transform into DrillRelNode
- do optimization on a logical plan

- all sounds reasonable
  - not just rel nodes
  - other pieces of states, like type factory
  - populated validator, needs to come along
    - type info not stored in tree, in validator
  - Julian - need to do work to decide which state is needed in each state of the pipeline
    - Jacques - would like to sit and talk about it
    - Julian - set a time next week
      - Monday, Tuesday, Thursday best

Mehant

- wrapped up the underscore map stuff
  - select * issues
- get changeset reviewed
  - remove _map from drill
- small patches to optiq
- queries will be much simpler to write
  - only difference now is just table names look like filenames

Aman

- new at MapR
- working on columnar database system at ___
- worked with query optimization, mpp optmizer
- query execution aspects, aggregations and joins, set operations
- IBM worked on OLAP
- Informix worked on Reb Brick

Jin Feng

- join order
- minor changes in Drill to use new optiq code, need update in conjars
- explicit casting
  - physical/logical plan
  - match cast code that is automatically generated
  - unit test, is working, will wrap up
  - connect with Yash, deciding type cast compatibility between types
  - allows explict casting
    - optiq needed casts to be happy but was removing them

Steven

- found memory problems
    - changes in some places fixed some of the problems
    - problem with sorts
        - thought he fixed it, cannot get memory released

Apache Drill G+ hangout 2013-11-19 — Status

Participants: Ben, Jacques Jinfeng, Jason, Julian, Steven

jacques
- parquet writer
- currently don't have thin client
  - work on separating client from rest of Drill
- evatica?
- designed to be asynchronous?
  - not specifically one way or the other
  - no thread pools or services
    - up to whoever provides the transport
    - sqlline is synchronous
  - tries to hold it memory
  - wants to print column headers
    - patch it to truncate it
    - option you can set to not retrieve the whole result set

ben
- bug fixes
  - join issue 301
- on the list NPE
  - enable error logging
  - logging to disk?
  - might be a problem with parquet reader
    - installed with binary, may not have bug fixes
  - circular buffer for logging, keeps it a reasonable size but can catch last 500, 1000 events/errors

mehant
- map stuff
  - alternate approach
  - our own drill data type
    - keep optiq validator happy, any type
    - answered in e-mail
      -

- can you select * from a table with no knowledge of columns
  - should be an error
  - jacques disagrees
    - obviously we want to select * from files
    - JDBC doesn't require record set has defined columns until return
      - cannot change meta-data while query is running
      - cannot send some records with one schema and change with later results
      - need to return a map
      - select * means expand column list with meta-data you have

jason
- file bug for memory usage in tests
  - use same base class for all tests
- defined interface for column select, filter pushdown and limit

Jinfeng
- join optimization in Optiq
  - condition is in where clause instead of on clause
  - join sequences are are having a Cartesian join
  - where optiq is enumerating join sequences
  - does it enumerate all trees?
    - multi-join rel
      - a lot of join into same relational expression
      - heuristic cost-based algorithm for join order likely to have lowest cost
    - need to apply associativity rule for join
    - swap join rule
      - if 3 tables need to
      - convert left deep tree to right deep tree
      - PushJoinThroughJoiRule
      - thing to remember
        - if you have a 5 way join, order 2^n join orderings
        - have to deal with combinatorial expansion
        - cannot use optiqs normal join ordering
        - multi-join rel should be used
          - approximation for best ordering
          - what is the threshold for use?
            - 6 joins
        - generate a few join orderings
          - could take that into next stage of processing

        - this is the approach hive is taking
        - they had same question, there is a JIRA for discussion
    - HIVE building their own RELs?
        - building their own calling convention
        - logical tree -> hive rels
- does optiq have select into or create table as?
    - stayed clear of DDL
    - does have support for insert
    - doesn't need help from optimizer
    - but syntax parsing
        - feel free to contribute it
        - no optimizer changes needed
    - insert into does need optimizer
        - last link in the pipeline needs to return number rows inserted
    - does it support hints?
        - no
- want to get optiq into apache incubator status

steven
- re-submit patch for spooling
- writing queries for performance testing

Apache Drill G+ hangout 2013-11-12 — Status

Participants: Jacques, Ben, Tim, Mehant, Steven Phillips, Steve McPherson, Jason, Harri, Michael

- Mehant working on Optique, custom functions, getting rid of MAP
- Tim talked with Steve re EMR, working on Whirr confs

→ Michael to merge Jason's note

Apache Drill G+ hangout 2013-11-05 — Status

Participants: Jacques, Jinfeng, Michael, Steve McPherson, Erik Ruben, Julian Hyde, Jason Altekruse, Mehant, Steven Phillips, John Morris

- Jacques
    - Will share roadmap
    - New committers upcoming to reduce the J-bottleneck

- Updates from Julian pending

mehant

- start looking at queries, noticable
- look at the cast problem
    - plug into sql _____
    - modifying each functions
        - comparison
        - checks types are the same
        - hacking optiq, allowing everything the
        - pointing to individual functions type checker
        - more generic way, not modify every functions type checker
        - add unit test, calls plus on int and any
            - try and get it to pass
            - modify the generic type checking stuff
            - should be a way, type checking is generic
            - uses a strategy pattern
            - can change pattern to treat 'any' types differently
            - user would need to specify enough type info when working with any plus any
            - min function
                - need to know int or string min function
                - min function is an aggregate
                - we do need to think about what our rules
                - minimize additional complexity
                - fail on undefined behavior
                - throw an exception or warning with mixed types
            - numeric plus or string plus
                - do we add a new one that works on unknown
                - initial validations, final binding to functions
                - we want to fix validation
                - choose any of the plus operators
                    - seems arbitrary
                - at some point, run time
                    - need to decide what to do
                - outside of what optiq handles for drill
                    - worthwhile to use optiq definition of operators
                    - parent definition
                        - many defs
                    - optiq
                        - multiple definitions and implementations

steven
- working on spooling of incoming batches
  - multiple phases, current phase might need to finish before loading new data from incoming node
    - dump to disk if out of memory
- john will give some input soon

micheal
- link site to google plus
- make site part of source control

Apache Drill G+ hangout 2013-10-22 — Status

Participants: Jacques, Michael Hausenblas, Lisen Mu, Yash Sharma, Jinfeng, Jason Altekruse, Harri, Steven Phillips, Timothy Chen, Julien Hyde

New employee at MapR: Jinfeng
- couple more in the next month

Jacques:
- merged limit
- clarify VVs
  - never access internal state of VV when it is invalid
- release notes

Steven:
- ordered partitioner
  - abstract out distributed cache interface
- continue to work on spooling to disk

Jason:
-semi-blocking
  - look at sort and ordered hash partitioner

Yash
- name of functions
  - separate class for operators and functions for more clarity
    - different operators have their own class files

Lisen
- fork of Drill
  - data pushed form leaves rather than pulled from root

- we have been thinking about this same problem
  - don't want to wait for IO all the time
  - pre-fetch rather than push
  - in a join you might get pushed a huge amount of data when you aren't ready for it
  - stream processing
    - alternative concept around foreman
    - not quite right for streams
    - resource allocation
      - not as much for resource requirements
- -HyperLogLog
  - space saving
  - acceptable - not precise
- data assembly - business logic
  - approximations will be important to drill
  - no serious thinking about sampling
  - certain types of scanners should support sampling
    - hard with some without reading all data anyway
    - Hbase might be easier to do a scan
  - doing it with their own business logic and statistics
    - hard to generalize

Hari
- not much for updates
- pick up with amazon ec2 docs
  - had problem where we need 8 gigs
  - cannot get it running on free micro instance
  - got it working removing the direct memory flag in POM
  - tim - out of memory exception right away
    - was this with or without changing the option for direct memory?

Tim
- wir patch in
- amp labs big data benchmark
  - having numbers for performance evaluation
  - set up on their repo for drill datasets
  - installing HDFS to all of the nodes
  - doesn't look to complicated
- cannot submit sql in distributed mode because of bad optimizer
- recent review board patches
  - describe code more completely
  - hard to review without docs

- Julien - single powerpoint slide per operator
- google doc? like the logical plan doc

Ben
- code gen portion of merging receiver
- no blockers
  - getting to code review soon

Julian
- joined hortonworks
- working on optiq
- helping hive, but also working on Drill
- making optiq everything it can be
- splitting JDBC into thin client
  - thinking about it, no implementation yet
  - right now pushing sorts down to Mongo
- jacques - session next week on JDBC?
- roadmap on optiq
  - commit logs tell some of the story
  - roadmap would be helpful
  - will put out call for optiq users like drill
  - put together feature list for next release(s)
  - next 6 months, want to be agile, but wants to be more predictable
  - Jinfeng will be working with optimizer and optiq

Apache Drill G+ hangout 2013-10-15 — Status

Participants: Jason Altekruse, Michael Hausenblas, Steven Phillips, Tim Chen, Jacques Nadeau, Ben Becker

Important note, we have finally ironed out the hangout link expiration.
Until it breaks again this is the link:

https://plus.google.com/hangouts/_/85a0b6bcc3fadfc9ce9c459d6e3c1cf0a3259045

Michael
- talks have been going well

- lots of interested users anticipating GA
- can the current release run in distributed mode?
  - response: steven
    - need to run physical plans
    - need to manually insert exchanges
    - need to add exchanges to optimizer
  - setting up cluster is easy, just connect drillbits to zooKeeper
  - right now we are focused on validating performance
  - so not as much focus on sql query submission
- if someone would write 3 or 4 steps to
  - launch 3 drillbits
  - connect to zookeeper
  - run a distributed query
    - even if its not use SQL yet
  - Michael can expand on it
  - Tim has patch for Apache Whirr, still waiting for merge
    - launches drillbits and connects to zookeeper
  - would be nice if we could run REST client at start of cluster
  - Jacques: Stateful client that sits in front of drillclient

Jacques
- empty batch issue
- update the clear contract for what record batch implementation should be used
- limit looks really close

Ben
- merging receiver operator
  - lot of progress, no real blocks

Steven
- more or less done with ordered range partitioner
  - checking in soon
- problems with hash exchange
  - for larger batches its duplicating rows
- next task spooling to disk
  - beginning of a fragment
  - doesn't solve the situation where a blocking operator has too much to handle
  - HDFS writes?
    - want to use the same disks, not just the OS disk
    - drill directories set up like MapReduce directories
    - config for each node

- could be /tmp, or a directory, or directory on data node

Tim
- patched Whirr
- no feedback on review patch tool
  - does it below in source control
  - should we have a tools directory for this and like IDE settings?
  - apache doesn't really have a concept of multiple repos for a project

Jason
- working on reader/writer
- BitWeaving integration grad students at UW Madison

Patch coming from Mehant more people from MapR will be helping with Drill

Tim
- drill release notes
- cannot edit wiki, need to talk to Apache infrastructure people

---

Apache Drill G+ hangout 2013-10-08 — Status

Participants: George Chow, Jacques Nadeau, Timothy Chen, Jason Altekruse, Steven Philips

- Jacques working on MapR's roadmap for next release, also JDBC related patches
- Yash Sharma working on math functions and planning a talk at Bangalore about Drill
- Tim has a working EC2 auto-deployment using Apache Whirr. Still blocked on Optiq problem for LIMIT patch.
- Jason fixes a Parquet writer issue, and continue to work on Writer interface.
- Steven finishing TotalOrderPartitioner work.

Apache Drill G+ hangout 2013-09-10 — Status

Participants: Harri Kinnunen, Jacques Nadeau, Timothy Chen, Yash Sharma, Lisen Mu, Georg Chow, Srihari Srinivasan, Ben Becker, Michael Hausenblas

- Jacques finishing off the M1 release
- Srihari updates on REST API, Michael to catch up and start implementing the WebUI
- Tim has a patch for LiMIT operator working end to end available on rb
- Harri on documentation. We need to consolidate the many places (Wiki, Git doc, etc.) and also provide guidance for end-users vs. dev setup

- Ben working on the Guide, will use Donuts as canonical example
- Lisen created a visual package overview, needs some sync
- Yash working on Floor/Ceil function, sorting out stuff with Julian
- Georg on ODBC/JDBC driver contributions

Apache Drill G+ hangout 2013-08-27 — Status

Participants: Harri Kinnunen, Jacques Nadeau, Timothy Chen, Francisco Freire, Ben Becker, Alec Ten Harmsel, Steven Philips

- Jacques working on better integration of schema with Optiq/JDBC, Streaming aggregate, improving basic optimizer
- Tim worked on JSONStorageEngine, and started on Limit Opereator
- Ben worked on MergeSort on single record batches, seeing bugs in multiple schema changes
- Lisen is looking into client reconnection problem, working on Semi-join (Bit vector) or Hash join based on Off-heap hash map index. Jacques said he worked on a Concurrent hashmap using off-heap memory and will try to dig that up. Can also look at Columnar sort using Radix sort.
- Steven Phillips worked on packaging, creating deb, rpm packages
- Harri continues to work on architecture documentation, and installation instructions
- Watch out for batches materialization, where Jacques found issues that after serialization value vector read expressions are pointing to outgoing batch column info instead of incoming
- Jira has issues tagged with Alpha, and please verify the issues and add/remove any as fit
- Alpha messaging?

Apache Drill G+ hangout 2013-08-20 — Status

Participants: Harri Kinnunen, Jacques Nadeau, Lisen Mu, Timothy Chen, Yash Sharma, Ben Becker, George Chow

- Jacques working on code generation and Optiq local mode/distributed mode
- Ben is working on Inner Join and relevant code gen
- Tim worked on JSON Record Reader end to end, currently adding HDFS read
- Yash met with a fellow Drill contributor, and plans to work on Mongo Storage Engine
- Harri is planning documentation of Drill, around architecture, operation
- We plan to use Optiq to also perform Physical plan optimizations, and leave our own basic optimizer simple as is.

Apache Drill G+ hangout 2013-07-23 — Status

Participants: Jason, Julian, Rahul Chitturi (eBay), Tim, Hari, Tanujit Ghosh (Barclays?), Michael

- New participant Rahul working with Tim (YARN)
- Jason worked on Parquet reader close to done (had to implement a scanner)
- Jason on getting a sql select where query to run
- JSON scanner implementation.. On at least two tests are currently failing and marked ignore. How do we fix? — Tim: now as Ben's ValueVector is available, I can integrate this
- Any update on writing physical tpc-h queries? — Sree not around
- REST update — Hari stuck with SQL, directly working on LP. QUESTION: will Drill client be factored-out? He'll send a message to the mailing list.
- Exchange sender—Ben's work
- Anybody want to start implementing DrillFuncs?—will be used for physical operators, makes it easier to work with ValueVectors
- Applying rest of optiq rels (from Jason to Julian)—some things have to be fixed
- Rahul will look into YARN

Apache Drill G+ hangout 2013-07-09 — Status

Participants: Jacques, Jason, Julian, Alexandre, Tim, Norris, Srihari

- Jacques soon to push review request for Project operator, and general function architecture
- Jason working on Parquet support
- Ben working on ValueVector
- Tim working with Jason on Parquet, wanting to map to Drill types
- Srihari working on the REST interface for DrillBits
- Julian extending Optiq in general (sort order optimization, etc), hopefully will be folded into Drill in the future.

Apache Drill G+ hangout 2013-07-01 — Status

Participants: Jacques, Jason, Julian, Sree, Tim, Lisen, Ben, Norris, Santhosh Daivajna, Michael

- Jacques work on project operator and aligning type system
- Jason working on Parquet support, then ORC
- Sree working on DRILL-47
- Ben working on ValueVector
- Tim working with Jason on Parquet, wanting to map to Drill types

- DRILL-77 progress (Hari/Michael)
- Michael reports on feedback from Hive London (re-starting queries, feedback from optimizer re cost estimate, AWS/S3) - will raise respective issues for it (+WebUI admin view or future release)
- Ask Hari to document his IntelliJ experience re setup on Wiki

---

Apache Drill G+ hangout 2013-06-18 — Status

Participants: Jacques, Ben, Jason, George Chow, Julian, Lisen, Hari, Shawn, Norris, Sree, Tim, Michael

- Jacques reports on HBase con - mentions Phoenix, PolyBase
- Jacques did a bunch of merges (JSONRecordReader from Tim, etc.) and working on filter
- Jason working on basic optimizer (LP to PP translation) and updated Website
- Tim: do we have an HBase storage impl? Jacques: David started this, if someone wants to pick it up?
- Michael (and also on behalf of Hari): working on REST API (DRILL-77), see design doc and once this is done, Michael to update/port WebUI (DRILL-58)
- Sree: looking into TPC-H for a query test framework
- Ben: work on LP builder
- Tim: got the physical ops re JSONRecordReader with JSON scan, try to run it end-to-end
- Julian: datatype discussion—as per mailing list

Apache Drill G+ hangout 2013-06-11 — Status

Participants: Jacques, Ben, Jason, Alexandre (CERN), Hari (Thoughtworks), Julian, Shawn, Norris Lee, Tim, Michael

- Hari and Michael to work on ISSUE-58, has to be splitted into two ISSUE, Hari will take care of the REST interface, Michael of the WebUI
- Jacques gives an overview on the highlights, incl. Ben's stuff
- Jason updated the Drill homepage (congrats from the group, well done!) and some progress on the code base
- Julian likes to support Optique integration
- Julian: all in the main branch now, adding tests
- Michael: working on ISSUE-58 with Hari now
- Tim: working on JSONRecordReader, quite some ops working already, some question on the scanning part (costing) open

- Tim: saw a nice benchmark we could have a look at https://amplab.cs.berkeley.edu/benchmark/
- Michael: request came up re S3 support — will file a Jira issue for it
- Alexandre: wanting to use in an existing CERN project that currently uses MapReduce-based processing. Goal is to improve latency. Currently an Oracle DB is used, but doesn't scale. Question is how to store the raw data: is it wise to store it in JSON? Other formats? Jacques: maybe use Parquet or ORC? also, benchmarking different formats would be interested
- Michael: related to Alexandre's question re which format to use, have a look at http://arxiv.org/abs/1305.6506 'Notes on Physical & Logical Data Layouts' (PDF, 5 pages)
- Jacques: Could have a look at basing this stuff on Phoenix
- Julian: is it possible, ATM, to run the whole Drill product end-to-end? Jacques: not yet, but very soon
- Discussion on ReferenceInterpreter and main engine merging

Apache Drill G+ hangout 2013-05-28 — Status

Participants: Jacques, Jason, Shawn, George, Lisen, Tim, Norris Lee, Michael

- Jacques: updated execution model, integrated Tim's, Julian's work
- Jacques: correlated subqueries should be available
- Tim: advancements in JSON record reader
- Jason now starts contributing to the codebase
- Lisen: need more test cases

Apache Drill G+ hangout 2013-05-21 — Status

Participants: Jacques, Julian, Shawn, Tim, Michael

- Jacques: goal was to have full query, now working on pushing it this week. Then, focusing on Julian's queries
- Tim: can we have a high-level description of the flow? Jacques: have a deck, will share ASAP
- Julian: Lisen's remark helped, now over that issue (aggregation/join working). Assumption: every table has a single column called 'MAP' (@ref='_MAP') containing a set of KEY-VALUE pairs, for type schema you'd use a view on it to cast it. Next up is UNION, testing with Mondrian dataset (24MB, thousands of queries) soonish.
- Jacques: plan is to create an identity planner (naive impl. or leverage Optique?). Julian: prolly a couple of days effort with Optique.
- Jacques: how to get from the LP to an Optiq PP? Julian: need to translate JSON tree into Optiq classes

- Discussion on how to bootstrap/identify the datasource in a late binding environment. Julian renders current options: SQL has foreign server concept, based on DDL; other option is ‘table function’, a UDF with parameters. In Optique they have a special table function (LISP macro analogy)
- Tim: JSON record reader—think I have a working impl, currently testing (mocking lib)
- Jacques: next thing up is also merge of SQL stuff
- Julian: makes sense to have the Mondrian 10MB compressed JSON dataset into Maven? Jacques: better to create one JSON doc per table than all in one and add to a JAR.

Apache Drill G+ hangout 2013-04-30 — Status

Participants: David, Deependra, Jacques, Shawn, George, Tim, Ellen, Ted, Pranjal, Sree, Michael

- Michael: can we follow a pattern in DLP like D(efinition)-E(xample)-N(otes)? Michael to give it a shot for some of the operators.
- Michael on DRILL-57—proposal in DLP, discuss
- Michael on DRILL-58—discuss foreman-discovery and REST interface
- Review of DRILL-57: provide a means for typing the fields, can utilise named expression (?), rule: no late-bind types
- DRILL-58 discussion: Drillbits should have a single interface for communication, which is our RPC interface. The REST interface is done in Java, part of the Drillbit JVM. Michael to come up with design (URLs, methods, payload, etc.) and WebUI sketch.

Apache Drill G+ hangout 2013-04-23 — Status

Participants: David, George Chow, Jacques, Tim Chen, Lisen Mu, Shawn O’Connor, Deependra, Julian, Ellen/Ted, Brian Enochson, Michael

- Jacques updates on progress (send pointers to David)
- Julian offers help re Optique
- Michael to contribute DRILL-57 ASAP (end of this week)
- Lisen re MySQL SE (depending on David’s SE for HBase)
- Michael: can we have a more detailed description (bridging high-level and code)? --> Jacques: once we have to core up and running (next week?)
- David: looking for a Java library for interprocess communication between JVMs

Apache Drill G+ hangout 2013-04-16 — Execution Engine Status

Participants: David, Jacques, Tim Chen, Lisen Mu, Shawn O'Connor, Sree V, Michael

- SQL parser—anyone wants to go for Union op?
- TODO-1: Jacques to share slides, done, see http://www.slideshare.net/ApacheDrill/drill-exec-work
- TODO-2: Michael to compile a list of external dependencies (such as Netflix curator, Hazelcast, etc.)
- TODO-3: Michael ISSUE-57
- TODO-4: Michael to compile test data and test queries (extend the donut re business/sales information—'crispy cream')