Notes from OHDSI OMOP Hadoop Working Group

10/28/16

sdolley@cloudera.com

These are formatted notes. Raw notes also available on OHDSI Github. This call was not recorded. Future calls, best efforts will be made to record them.

Roll Call

* Albert “Al” Pivonka
* Aaron Galaznik
* Brack Coalson
* Nagakalyana Eskala
* Lee Evans
* David Ficenec (pronouned Fis-nek)
* Charity Hilton
* Vojtech Huser
* Steve Lyman
* Malcolm McRoberts
* Tom White

Not in attendance but interested: Derek Kane, Frank de Falco

Use Cases one might use Hadoop for?

* ETL platform – some parts of Hadoop could be used for ETL or to stage ETL
* Storage/warehouse – use Hadoop project(s) as a place to store base data
* Analytics engine – a method for running analytics. One benefit mentioned was speed.
* Natural Language Processing - Existing member, in the OHDSI NLP Working Group, in using Hadoop for NLP
* OMOP CDM but not for real world evidence - use of CDM solely as a data model, untethered to any metrics that OHDSI/OMOP has today
* Search - SOLR is part of Hadoop ecosystem (based on Lucene which “everyone uses”). Regenstrief puts structured data in SOLR and runs it without having Hadoop stack underneath SOLR. Search would be a use case enabled by Hadoop.
* Sensor data – Hadoop might be a good fit for the kind of data that comes off sensors or wearables, however the OMOP CDM would need to accommodate sensor data first before trying Hadoop on it.
* Method for non-SQL approach to tasks - Doing things outside of SQL (might be ‘support for code based data access rather than SQL based). (While someone thought Hadoop might be good for a use case of working outside of SQL, someone else said SQL is so ingrained into many parts of OMOP that you will never be SQL free, even if you call SQLRender or do other things, SQL is in there.)
* Lee: creating pre-aggregated data sets (such as frequently used cohorts or cohorts of interest), there is a need there and could be a benefit. Could tie to Achilles.

Topics

How can one know that a Hadoop stack in OMOP is working?

* Typical test is run full Achilles library, if Achilles will run then it is done

Known issues

* SQL Render works in SQL-92, does Hadoop support SQL-92?)
* Someone tried to port to Google BigQuery and ran into problems with temp tables.  If there are Hadoop projects that don’t support temp tables, this could be an issue for those projects to be used
* Tom had to convert date/time into strings.  Impala has timestamp but OMOP CDM does not have a timestamp all the time. So strings might not be good.

Getting it working

* CSV file inside \_\_ and drivers. If you are good at R and have Hadoop, can make change to csv file and it will work?
* SQL Render engine uses R (and wrapping Java functions). Uses csv to translate SQL Server dialect to other dialects.
* Different data sources can be compared via code to see how they are different
* Tom White did something to create Hive tables that map to OMOP CDM.  Hive has a metastore, so table definitions live there.  Impala hits Hive metastore. SQL hits Impala, not Hive.
* Patrick Ryan committed that 10/27/16
* Tom looked at windowing functions in Impala (this troubled Google Big Query?)
* One could add two different new translations (?) in csv file, one for Hive and one for Impala?

On selecting a ‘stack’, which open source projects (HDFS, Hive, Impala, others) are best used or not

* OHDSI stack centered on SQL.  Can’t do it all in R.
* Look at other open source projects that serve same function: HDB (hortonworks), Hbase, Tez, Splice Machines (not open source?),  - whichever ones are worked on probably need to support SparkSQL.
* Options for not using HDFS, access Hive Metastore for table definitions and use R
* For ad hoc, use Impala and SparkSQL, and for batch use Hive.
* Is there a need to support Hive queries (beyond just using Hive metastore)?
* Need to support S3.  If ETL is hitting S3 as the source.  Appending new data as it comes in.  Dataframes in Apache Spark (using for ETL) has worked well in some ways.
* YARN and MESOS are different.  We may not need to ‘handle’ this because other systems ‘above’ those work out how to handle them without intervention.
* We need to tell people what file formats work best and worst per data/etc.  “Cuts a lot of wasted \_\_\_\_”.

Questions that arose, some with answers:

* Vojtech: shouldn’t our goal be first, making existing tools work on Hadoop; second: Hadoop is opening up new tools?
* Is Hadoop only for OMOP CDM? Or are there other OHDSI projects that would leverage Hadoop?
* What are the cases for graph databases? These technologies are interesting in the concept and terminology space.
* How does use of Hadoop for ETL relate to cloud vs on-prem or differ cloud vs on-prem
* What is Cloudera roadmap for SparkR in CDH? Also for SparklyR what is the roadmap vis a vis CDH?
* Looking at non-Hadoop ‘tools’ to improve speed, performance. Patient level prediction or Cohort, optimization.  To think about: if our goal is performance, than widen the focus to include other areas outside of Hadoop.
* How can we get access for members to a cluster? Answer: Quickstart VM allows you to run a cluster on a machine.
* What is roadmap for Cloudera in Hive and Impala to improve SQL? From Tom: general direction that Hive and Impala are adding more functionality (e.g. windowing functions) over time. Cloudera has been assuming HDFS would be storage, new one coming called Kudu, allows updating a database like normal db’s.  Hadoop db’s today cannot update existing rows, Kudu does and available immediately.
* Could one, on a query by query basis, choose between Impala and Hive.  But data stored once in HDFS.  Table structures would be stored once in Hive metastorm?
* Think about data that is not traditionally healthcare data, what is roadmap to merge different kinds of data (e.g. linked data?)?
* Can we use notebooks to re-create some of these tools?
* Taking queries and analysis that we have, some rewriting? Cannot natively take R code today and make it SparkR? Support of data frames might be intermediate goal?

NEXT STEPS

1. Post these notes
2. Reformat these notes
3. Get a list of things to vote on
4. Get votes
5. Get list of coding
6. Next time record meeting!!