OHDSI OMOP Install Notes

I encountered several issues documented (and not documented) on the OHDSI forums when implementing the OHDSI OMOP tools. My notes on my experiences, configuration, misconceptions, and issues are detailed below. My hope is that this can help improve documentation and serve as a guide for future implementers.

Configuration:

- Windows 7
- Postgresql pgadmin3 LTS by BigSQL is database
- Apache Tomcat 9.0 used as web server
- R version 1.3.5
- Ant to build Java files
- Maven to build WebAPI .war file
- Java jre1.8.0_191 and jdk1.8.0_191

Issues encountered (so far):

1. Timeline
    a. Overall, it's taken a two months to setup the OHDSI OMOP tools required to get Atlas running
    b. It would be helpful if documentation suggested a potential startup timeline such that reasonable timeline expectations could be established among various stakeholders. My advisor was hoping setup wouldn't be as cumbersome as it has been and that we could get up and running quicker with our own data.
    c. There are a lot of code repositories on github for OHDSI (>125), and it was unclear which of the tools are needed at each step in the architecture pipeline, and which are obsolete (Achilles Web) or less commonly utilized. It would be useful to know this information for implementation scoping purposes. The tutorials are helpful for a subset of these tools, but many are not covered in the tutorials.
2. Virtual machine vs native install
    a. I opted for a native install rather than using one of the provided virtual machines/docker environments as it's important for me to know how each component works
3. Database selection
    a. My advisor really wanted me to run MySQL (due to familiarity with this database), however it would not install on my machine (as its Windows 7).
    b. I've since found that MySQL isn't really one of the supported databases in the documentation, though it's hard to find when you're first deciding upon an appropriate architecture for OHDSI OMOP tools.
    c. Initial installation was done using Postgrsql
    d. Due to running Windows 7, I get some benign errors every time I try to do something in Postgresql in pgadmin3 (e.g., startup).
    e. I'll likely migrate the database to the organizations enterprise Oracle database once feasibility of the OHDSI OMOP tools is assessed via my pilot study
4. CDM & Achilles
    a. There are not really any good OHDSI tutorials or guides for doing the ETL process

    b. It took a while to figure out what schemas referred to for Postgresql, as I originally built everything in a single database
    c. You need to have all of the tables created that are part of the CDM, even if you don't plan to populate them with data
    d. You can't have any misspellings in terms of table names or variables. Make sure your names match the specs. This may manifest as you setup Achilles as errors in results.achlles_results_dist with cluster apply. http://forums.ohdsi.org/t/achilles-error-with-clusterapply/5771
    e. Running multithreaded didn't work for me, and thus I ran with 1 thread. Running 1 thread was useful in debugging where the actual issue was. Running multithreaded took too long to finish merging the results and thus I ended up killing it. For this reason, single thread ran faster.

5. Achilles Web
    a. I never actually got this functionality to work.
    b. The exporttoJson ran 3+ days so I killed it once I discovered in the forum that Atlas is pulling from the WebAPI rather than from Achilles Web (thus I didn't need this data after all).
    c. Ideally, documentation could be improved to state that Achilles Web is no longer a requirement for Atlas.
6. Athena
    a. There are several cases where symbols utilized in the vocab for foreign words are the cause of the invalid byte sequence for encoding "UTF8" error, thus I had to manually go through the vocab to delete the letters at issue. Perhaps Athena could request the database it's going to and exclude these concepts (or letters) at download so that this isn't a manual process (assuming other database can handle these characters). http://forums.ohdsi.org/t/copy-command-error-loading-vocab-v5-into-postgres-on-windows/2747/11
    b. The vocab from Athena needs to be imported into the Postgresql database in a vocab schema for certain tools to run.
7. WebAPI
    a. Preparing the schema must be done when connected as ohdsi_admin_user rather than the default/root postgres user in order for default permissions to be assigned correctly. I didn't quite understand this step due to my unfamiliarity with Postgresql, so I had to manually reset all of the permissions for those tables to accept the ohdsi_app role for viewing. Permissions are very important for WebAPI to work. http://forums.ohdsi.org/t/webapi-permission-denied-cohort-generation-info/5815
    b. Bit confusing about how to deploy. Found that Apache Tomcat worked for my purposes, but could potentially have used other solutions.
8. Atlas http://forums.ohdsi.org/t/atlas-setup-failing/5858/6
    a. Versioning: The setup for Atlas is different when you pull it from the Master branch rather than the 2.6 release version. Apparently, the 2.6 release is easier and requires less steps to get working. I'm unsure what the steps are for the master version, as I didn't get this working.

b. Configuring sources and sourceDaimons was confusing. Should have 1 source with 3 daimons. Be careful with misspellings here as well.

c. There are some useful debugging steps on the forum for debugging the vocab and CDM/results in order to figure out if things are working or not and at what stage things are failing.

d. Cohort generation step fails as results.cohort_inclusion does not exist. There is a DDL script generated automatically in Atlas for postgresql that can setup the required tables, however, this step is not documented at all in any of the install documents I've found, but appears necessary in order to build a cohort.

e. Running this DDL script may take a long time…

i. There may be a lock on the tables from Atlas, thus not allowing the tables to be created and causing your script to run forever

ii. The drug script may also run forever… I ended up killing the process after it had been running for 7+ days: http://forums.ohdsi.org/t/long-running-ddl-script-for-drugs-to-create-hierarchy-w-ingredients-and-atc/5964

iii. To address this, I tuned some of Postgresql parameters using https://pgtune.leopard.in.ua/#/

iv. I ran the following SQL code to create indices to speed up the operation:

```
CREATE INDEX concept_id ON vocab.concept(concept_id);
CREATE INDEX descendant_concept_id ON vocab.concept_ancestor(descendant_concept_id);
```

v. I then ran each piece of the query separately, which all ran in < 5 minutes each and worked on joining them together piece by piece

vi. Interestingly, the select command with everything now runs in 5:51 minutes. The full command executes in 1:18 minutes with the insert… odd. Recommend adding the above index commands

9. Patient Level Prediction

a. I was running into an issue where something changed in a dependency and the code broke, but this is fixed in the latest versions: http://forums.ohdsi.org/t/patienlevelprediction-installation-check-response-codes/6034

b. To get around the issue about continuous risk factors in existing risk stratification models, we redefined or created additional variables to capture these in a binary manner. Turns out, the variables we're using in our model are not recommended… The technical question on how to do this without creating additional variables remains unresolved. http://forums.ohdsi.org/t/implementing-existing-prediction-models-with-continuous-risk-factors/6162/2

10. Implementers forum

a. Extremely helpful. Thank you everyone who provided advice in getting this setup.