Refactoring OHDSI cohort queries for performance: lessons from VA study participation

Benjamin Viernes[1,2], Marc A. Suchard[1,3], Patrick R. Alba[1,2], Katherine R. Simon[4,5], Michael E. Matheny[4,5], Scott L. DuVall[1,2]

1: VA Informatics and Computing Infrastructure, Department of Veterans Affairs, Salt Lake City, UT; 2: Department of Internal Medicine Division of Epidemiology, University of Utah School of Medicine, Salt Lake City, UT; 3: Department of Biostatistics, University of California Los Angeles, Los Angeles, CA; 4: VA Informatics and Computing Infrastructure, Tennessee Valley Healthcare System, Nashville, TN; 5: Department of Biomedical Informatics, Vanderbilt University, Nashville, TN

Current Word Count: 1034, Max = 1000

Background

OHDSI network studies leverage a "…global, open-science community that is committed to generating real-world evidence to both support clinical decision-making and advance the methodology within this field." [1] The United States Department of Veterans Affairs (VA) participates in network studies that advance scientific knowledge that have the potential to improve the health of Veterans while also serving the OHDSI and research communities due to the size and diversity of the VA population (Table 1). VA is the largest integrated healthcare system in the United States.

Executing OHDSI network studies in VA require substantial computing resources. One of the systems approved for conducting VA research is the VA Informatics and Computing Infrastructure (VINCI). VINCI supports more than 9,000 researchers across the United States working on almost 5,000 research projects. [2] Operating in VINCI requires that changes are made to the SQL generated from ATLAS (Circe SQL) in order for the queries to complete. [3] Circe SQL contains many common table expressions (CTEs) and nested sub-queries, which gain further complexity as correlated inclusion and exclusion criteria increase.

CTEs and subqueries require large amounts of data to be held in memory, especially those with multiple calls and complex joins as many Circe SQL queries contain. In servers with many users competing for memory, substantial lags or frozen/timed-out queries can occur. Although it requires space in SQL Server's tempDB, utilizing temporary tables to reduce the number of sub-queries and CTEs in complex queries can limit the amount of space required in memory at any given time.

VA recently completed a study that included 41 cohorts to evaluate potential adverse effects to COVID Vaccines. Many of these study cohorts could not be instantiated by running Circe SQL in VINCI, requiring an alternative method. In this Collaborator Showcase, the complexity of Circe SQL is compared to a method that replaces nested subqueries with indexed temporary tables to avoid multiple calls to OMOP tables joined to cohort code sets.

Methods

A new process was created to ingest Circe SQL from an OHDSI Study package, replace sub-query calls to OMOP domain tables into separate queries that result in clustered column-store indexed temporary

tables, and output custom-indexed SQL that can be used within the same OHDSI package to instantiate all cohorts within VINCI. [4]

Execution of all 41 study cohorts were done using the Circe SQL and the custom-indexed SQL methods and then compared. We attempted to instantiate cohorts from all 82 SQL scripts. The method and execution duration were recorded, in addition to some cohort details, including the number of concepts per cohort code set from each OMOP domain and the standard SQL temporary tables (of the 10 possible) used in the inclusion/exclusion criteria. Since Circe SQL cohorts vary in complexity, we evaluated the number of concepts and codesets, the domains utilized, and the correlated inclusion criteria. Using descriptive methods, we compared the difference in execution duration across methods for those that were able to execute completely. For the Circle SQL cohorts that were not able to complete in VINCI, we compared the differences in the inclusion criteria required between those cohorts and the other cohorts that finished without issue.

Results

29 of the Circe SQL and all 41 of the custom-indexed SQL completed and generated cohorts. The cohort sizes were the same between methods for all SQL that completed. Execution time for Circe SQL ranged from 5 to 638 seconds (mean [SD] 131.3 [168.0]) for those that completed compared to 7 to 1481 seconds (311.2 [367.8]) for custom-indexed SQL. The mean execution time was 180.2 seconds shorter for the Circe SQL method for those that completed. Differences in execution time are visualized in Figure 1. In the cohorts where the Circe SQL did not instantiate a cohort, the custom-indexed SQL ranged from 123 to 1481 seconds (318.8 [375.0]) seconds.

Among the 12 Circe SQL cohorts that did not complete in VINCI, several observations were made. All contained at least 3 correlated criteria sections, while only 4 of the 29 that did complete contained 3 or more correlated criteria. Nine of the 12 Circe SQL cohorts that did not complete used OMOP Visit domain inclusion criteria combined with OMOP Condition domain criteria, while none of the 29 that did complete included any OMOP Visit domain criteria.

Conclusion

Although the Circe SQL frequently executed more quickly on average when they did complete, 12 of the 41 Circe SQL cohorts did not complete. When these cohort queries were refactored using indexed temporary tables to replace subqueries, they initially took more time to build and index, but could then be re-used, which was beneficial in the scripts that re-used the same subqueries multiple times. All cohorts executed in VINCI using custom-indexed SQL. Without this refactoring process, it would not have been possible for VA to perform this network study.

Some OHDSI community members have brought up these issues [5] [6], and these methods may be useful to spur further research and provide some direction for changes. Additional research is necessary to understand whether the method ATLAS employs to translate cohort definitions to SQL scripts can be improved further, whether within VINCI or other environments that participate in OHDSI network studies.

Limitations

The inability to control the server and adjust for other users and costs, especially given that it took several days to run all 82 queries. There is further depth that would be beneficial to understanding the relative effect of the VA server on these queries including the architecture of VA OMOP compared to other OMOP instances and OMOP best practices. These are not described in this abstract.

Bibliography

[1] O. H. D. S. a. Informatics, "OHDSI Studies," [Online]. Available: https://data.ohdsi.org/OhdsiStudies/. [Accessed 08 06 2023].

[2] U. D. o. V. Affairs, "VA Informatics and Computing Infrastructure," 2023. [Online]. Available: https://www.research.va.gov/programs/vinci/default.cfm. [Accessed 08 June 2023].

[3] O. H. D. S. a. Informatics, "Circe-be Github," OHDSI, 13 January 2023. [Online]. Available: https://github.com/OHDSI/circe-be. [Accessed 08 June 2023].

[4] V. I. a. C. Infrastructure, "vinci-ohdsi VA Tools," December 2023. [Online]. Available: https://github.com/vinci-ohdsi/VaTools/. [Accessed 08 06 2023].

[5] OHDSI, "circe-be Issues," August 2022. [Online]. Available: https://github.com/OHDSI/circe-be/issues/172. [Accessed 08 06 2023].

[6] OHDSI, "OHDSI circe-be Commit," September 2022. [Online]. Available: https://github.com/OHDSI/circe-be/commit/a87a63ccec65f16363ecd6eadd8bfbfea4ffba2d. [Accessed 08 06 2023].

Tables and Figures

Table 1. VA OMOP Domain tables used within 41 study cohorts

OMOP Table	Row Count	Column Count	Data Size(Mb)	Index Size(Mb)
MEASUREMENT	20,018,341,479	28	842,973	1,611,091
PROCEDURE_OCCURRENCE	3,211,481,795	18	119,508	258,461
CONDITION_OCCURRENCE	3,379,540,042	20	138,768	271,987
VISIT_OCCURRENCE	4,251,754,175	22	171,081	342,047
DRUG_EXPOSURE	7,031,141,405	28	344,168	566,009
DEVICE_EXPOSURE	262,307,924	20	10,113	21,117
OBSERVATION	1,251,843,930	23	46,839.00	100,770
PERSON	26,568,156	23	1,079.00	981

Figure 1. Cohort SQL Execution Time in Seconds