# DAA: A review of the specifications for the Diaspora Analytics Application

**Dr. Tamaro J. Green, Dr. Woineshet Meaza**

II Data School Report

## Abstract

This paper provides a review of data warehousing and business intelligence solutions for integration in the DAA, Diaspora Analytics Application. Yeoh and Popovič (2016) survey some of the roles and methodologies for establishing a successful business intelligence initiative. Yeoh and Popovič (2016) list some of the critical success factors to include a commitment from management, business centered, business driven, and user oriented approaches, and scalable and flexible technology frameworks. Yeoh and Popovič (2016) list some of the roles to include managers, analysts, architects, and consultants. Dobre and Xhafa (2014) describe parallel programming and distributed frameworks for big data and how their efficiency improves scalability and performance. Dobre and Xhafa (2014) also evaluate various frameworks including MapReduce, Hadoop, Hive, and Spark. Dobre and Xhafa (2014) also suggest methods and techniques for sharing data and online processing.

## Introduction

Meaza (2019) demonstrates design options suitable for building a diaspora analytics application. M. Chen, Mao, and Liu (2014) explain hybrid data warehouses implementing various types of analysis including structured, network, mobile, text, and web data analysis. M. Chen et al. (2014) also describe traditional data analysis techniques that can be implemented including cluster analysis, factor analysis, correlation analysis, regression analysis, A/B Testing, Statistical Analysis, Data Mining Algorithms, Bloom Filter, and hashing. M. Chen et al. (2014) list some of the challenges of big data to include data representation, redundancy reduction, data confidentiality, data life cycle management, and analytics. Include sources of data to include log files, sensors, and network data. M. Chen et al. (2014) also mention some of the parallel processing models to include Message Passing Interface (MPI), MapReduce, and Dryad. Tian, ÖZcan, Tao, Goncalves, and Pirahesh (2016) explain many join algorithms that can be implemented in data warehousing. Many distributed file systems fetch tables and conduct joins in the database. Some distributed file systems conduct the joins in the file system, also known as a broadcast join. To increase selectivity in certain instances, additional joins include repartition joins and zigzag joins (Tian et al., 2016). A repartition join transfers a database table to a distributed file system and conducts the join. A zigzag join algorithm alternates between the database tables and the distributed file system. Liu, Golab, Golab, Ilyas, and Jin (2016) analyze and evaluate five big data frameworks including Matlab, PostegreSQL, System C, Spark, and Hive. Liu et al. (2016) discover that C performs very well

but requires a higher programmer skillset. Liu et al. (2016) also identify that PostgreSQL with MADlib and Matlab are user friendly processing tools. Liu et al. (2016) find Hive more user-friendly than Spark.

## Review

The diaspora analytics application optimizes the capabilities of parallel programming and distributed computing. There have been many advances in data analysis with improvements to textual analysis, data visualization, and parallel and distributed computing. Sun, Bie, and Zhang (2016) define five operations for semantic relations to include generalization, addition, difference, complement, and inverse. Associate various relations for studying the semantic relational basis in scientific research for researchers, publications, emails, and publications. Sun et al. (2016) demonstrate that searching relevant information can be accomplished by increasing the connection between resources semantically and including relevant results from queries. Goudarzi and Pedram (2016) propose an algorithm that improves efficiency for cloud data centers. The algorithm ranks solutions for virtual machines based on containers, racks, and chasses. The algorithm also optimizes virtual machines based on migrations and objectives. Goudarzi and Pedram (2016) also enhance emergency operations by reducing over provisioning of resources through monitoring and reactive decision making capabilities. Goudarzi and Pedram (2016) stress the cost of energy to be empirical to designing workload distribution and calculate how to optimize costs while minimizing disruption to the service level agreements.

Leijon, Henter, and Dahlquist (2016) explain how mutual information techniques quantify information received between recipients in communication. Values can be measured as the score of correct responses and the probability of correct responses. Leijon et al. (2016) explain methods of analyzing confusion data with Bayesian statistics, applying consonant recognition testing, mutual information simulations, and confusion matrices. Leijon et al. (2016) support the Bayesian distribution over the null hypothesis to provide estimates of conditional probability of observed data. Leijon et al. (2016) test both singular and multiple conditions and approximate samples for data sets. Leijon et al. (2016) implement the Dirichlet distribution model which is often applied in machine-learning algorithms. Leijon et al. (2016) calculate standard differences from observed matrices and test significance at various levels. Krulewich, Yin, Bundick, and Zeng (2015) test parallel computing with supercomputing resources. Krulewich et al. (2015) describe financial analysis data as high frequency which can be analyzed with frequency equations. Krulewich et al. (2015) compared MPI Fortran with a multi GPU implementation and experimented with multiple GPU configurations. Krulewich et al. (2015) demonstrate that parallel computations can be implemented to computate stochastic volatility. Soares, Canizes, Vale, and Venayagamoorthy (2016) explain the potential of solving nonlinear programming problems that require a heavy work load for solving

energy resource management problems. These solutions are beneficial for managing power in energy storage systems. Soares et al. (2016) manage the demand response and the load control of power sources with optimization techniques and demonstrate the capabilities of nonlinear programming on data analysis systems.

## Method

Apache Spark is large scale data processing tool that can run on top of Hadoop. Apache Spark programs can be written in a number of programming languages including Java, Scala, and Python. Apache Spark has a shell program for command line programming in Scala and Python. Spark can read data from a variety of formats including JSON or CSV and stored in database objects called data frames. Spark can perform SQL queries and programming applications on data frames. Data frames can be a beneficial way for analyzing subsets of data. Spark can store the data frames as objects. Objects allow various operations and methods. The data frames also support queries. Being able to support queries and operations, data frames can be an efficient tool for data analysis, data integration, and data migration. Parquets allow persistence storage of data frames and can perform both read and write functions. Parquets can also assist in backup and storing data for analysis. Spark can also read data from JSON or CSV files. The data can be converted to a data frame. Processing information from JSON files can be beneficial for analyzing data from data streams or non-relational database management systems. Various databases can be run on this environment. DerbyDB is a database that is built in Java. It has a console that allows SQL queries to be entered on the command line. The DerbyDB server can be started from the command line and DerbyDB databases can also take connections from JDBC connectors. Hive is a data warehouse application for large datasets. Hive is also written in Java. After Hive is installed and running, queries can be run in Hive with a command line tool called BeeLine. DerbyDB can be set as the database type for running queries when Hive is started. Derby queries can be run from the interactive console. The console allows execution of extract/transform/load scripts to run from the command line. Derby has a schema similar to other relational database management systems. System tables manage the user and database information. There are also a number of tables or security, roles, and access. DerbyDB has additional resources for triggers and prepared statements. DerbyDB can run as an embedded framework, on the same Java Virtual Machine (JVM), or as a client/server framework, a different JVM, as an application. System properties are stored in a Java properties object. Prepared statements can be beneficial for repeated queries such as ETL queries. DerbyDB allows prepared statements through Java methods. The result set for DerbyDB queries also accepts operational calls and methods as a Java class. SQL exceptions can be displayed as error messages. DerbyDB also has support for internationalization.

Drill is a SQL Query Engine that can run on Hadoop. Drill also has a web interface for managing connections and queries. Drill

provides independent engines called drill bits for executing queries. Queries in Drill can query the physical data or the logical data. Data warehousing tools that allow for queries in a variety of formats can support data transfer to and from different data sources. Drill contains sample Region and Nation data. Drill can load this data as a parquet. Drill can be configured to access the data from a Derby database or another database. Drill can be an excellent tool for mining data from databases. Drill has various measurements for performance. Performance measures can be a beneficial tool for optimizing queries. There are many metrics that drill measures for performance. Drill also displays statistical values in metrics. Drill is a very capable tool for a data mining and data warehouse platform. Drill also maintains logs of query executions. The logs of the query executions are also available in the web interface. Drill contains sample data for regions and nations in parquet files. Drill can query these files with SQL queries in the web interface. Drill contains a list of the completed transactions in the web interface. It also displays whether the transaction passed or failed and the time of execution for the transaction. Drill can read queries from JSON and parquet files. Drill recognizes the query with the prefix dfs for a data file. Drill can display the results of the queries in the web interface. Khalifa, Martin, Rope, McRoberts, and Statchuk (2016) propose an adaption of Drill that makes the application more capable for data science and big data. Many data mining tools are available in a variety of programming languages. Dem et al. (2013) present applications of the Python language in data mining. Miškuf, Michalik, and Zolotová (2017) evaluate the number of users for open source and commercial data mining tools. SQL Squirrel can connect to a variety of databases including Hive. SQL Squirrel is one of many popular tools that can be implemented for seeding databases.

For gathering data, sensor data displays distance readings from various areas. Having test data can enable the various processes of the data warehousing strategy. Sensors that read distance information transmitted data through the MQTT protocol. ROS, MQTT, and Kafka are messaging protocols that allow for the transmission of data for the Internet of Things (Furlong, Remy, & McClendon, 2016). To gather test data, one option is to get data from devices. ActiveMQ is a messaging protocol application that can act as broker for various messaging protocols. ActiveMQ includes a web interface for monitoring the messaging protocol. ActiveMQ can act as a broker for the messaging protocol MQTT. MQTT is a protocol that handles the exchange of messages through a subscription/publication process. ActiveMQ can serve as a broker for a variety of protocols. The brokers can listen to different ports on the ActiveMQ host. Consumers can publish messages to all the subscribers of the topic in a group messaging technique. This distributed messaging allows subscribers to communicate with each other through the ActiveMQ broker. Sensor information can also be sent through cloud applications. Some cloud applications offer pre-made applications for building applications for the

Internet of Things. The BlueMix cloud platform is a IBM's cloud platform and has a number of solutions for developing IoT cloud applications (Das, Usmani, & Jain, 2015; Kobylinski, Bennett, Seto, Lo, & Tucci, 2014).

Research on integrating data warehouse solutions with cloud platforms will be in another study. Another application that is capable of delivering configuration for IoT devices is Node-Red (Olsson & Asante, 2016; Sbrizzi). Node-Red is a JavaScript application that allows the configuration of various IoT devices. Research on various configurations of sensor devices for collecting data will be provided in future research. After collecting data, the next stage in the analysis of the data warehouse applications, is importing the data into the data warehouse. Apache Tajo is a business intelligence tool that can be configured for running queries on datasets from a variety of sources. RStudio is an integrated development language for the R programming language (Racine, 2012). The R programming language is capable of data visualization with add on libraries. Apache Zeppelin is a business intelligence program that can import data from a variety of sources. Apache Zeppelin can handle streaming data and create advanced visualizations (MadhaviLatha & Kumar; Traub, Steenbergen, Grulich, Rabl, & Markl).

## Findings

A wide variety tools and methodologies exist for establishing a cable diaspora analytics application (Yeoh & Popovič, 2016). Parallel programming and distributed frameworks for big data can improve efficiency, scalability, and performance (Dobre & Xhafa, 2014). Hybrid data warehouses can perform various types of analysis including structured, network, mobile, text, and web data analysis (M. Chen et al., 2014).

This research also examined some of the open source solutions for extending big data warehouse capabilities of the diaspora analyticsapplication. Big data frameworks include Matlab, PostegreSQL, System C, Spark, and Hive (Liu et al., 2016). Parallel and distributed computing play an important role in modern data warehouses (Krulewich et al., 2015). Join algorithms are a common task in data warehouses and can be improved with distributed file systems (Tian et al., 2016). Algorithms can improve the efficiency of data management in cloud centers (Goudarzi & Pedram, 2016).

Business intelligence platforms can be implemented in text mining, knowledge management, and semantic discoveries (Sun et al., 2016). Business organizations can create a data warehouse platform on commodity hardware capable of distributed processing.

Organizations can deploy a Hadoop single node cluster or distributed node cluster with data nodes configured in a distributed file system. The named nodes cluster can be configured for a variety of storage types. Various cluster measures are available in Hadoop such as the applications statuses on clusters, the status of nodes, memory usage, virtual cores statuses, and scheduler metrics.

The Spark processing engine with the Hadoop framework provides an efficient framework for distributed data processing. The Spark framework allows programming in Scala and Python. The Spark framework also has capabilities for text mining and data mining. Organizations can run queries with Hive and connect to a variety of databases such as PostgreSQL, MySQL, Derby, and Oracle. Derby runs as a Java application and can initialize with a schema for Hive. Security measurements can also be configured for Derby. Beeline and Hive CLI provide command line interfaces for connecting to the database.

## Conclusion

There are a number of open source solutions that are available for creating a data warehouse for processing big data. Baer, Peltz, Yin, and Begoli (2015) explain how to build a data warehouse platforming implementing SparkSQL, MLab, and GraphX. Y. Chen and Bordbar (2016) propose a distributed rule engine on Spark. There are also a number of commercial tools that implement modern techniques for data management such as parallel computing, distributed programming, MapReduce, genetic algorithms, and data analysis and statistical tools. In this research, open source tools provided a platform for which to migrate from existing data management functions to more modern functions. There are a number of tools that can be evaluated for future research. Another tool that is beneficial for processing streaming data is Apache Flink (Junghanns et al., 2016). Pallonetto, Mangina, Finn, Wang, and Wang (2014) suggest research in building an API and database for managing sensor data. Future research may involve further studies on distributed and stream processing and creating a platform for real time analysis.

II Data School provides reports for applications in data science.

The authors declare no potential conflict of interest in the information presented in this review.

References:

Baer, T., Peltz, P., Yin, J., & Begoli, E. (2015). *Integrating apache spark into PBS-Based HPC environments*. Paper presented at the Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure, St. Louis, Missouri.

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications, 19*(2), 171-209. doi:http://dx.doi.org/10.1007/s11036-013-0489-0

Chen, Y., & Bordbar, B. (2016). *DRESS: a rule engine on spark for event stream processing*. Paper presented at the Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, Shanghai, China.

Das, N. S., Usmani, M., & Jain, S. (2015). *Implementation and performance evaluation of sentiment analysis web application in cloud computing using IBM Blue mix.* Paper presented at the Computing, Communication & Automation (ICCCA), 2015 International Conference on.

Dem, J., #353, ar, Toma, #382, Curk, . . . Zupan. (2013). Orange: data mining toolbox in python. *J. Mach. Learn. Res., 14*(1), 2349-2353.

Dobre, C., & Xhafa, F. (2014). Parallel programming paradigms and frameworks in big data era. *International Journal of Parallel Programming, 42*(5), 710-738. doi:http://dx.doi.org/10.1007/s10766-013-0272-7

Furlong, M., Remy, S. L., & McClendon, J. (2016, 15-17 Dec. 2016). *An empirical evaluation of the effects of IoT messaging protocols.* Paper presented at the 2016 International Conference on Computational Science and Computational Intelligence (CSCI).

Goudarzi, H., & Pedram, M. (2016). Hierarchical SLA-driven resource management for peak power-aware and energy-efficient operation of a cloud datacenter. *IEEE Transactions on Cloud Computing, 4*(2), 222-236. doi:10.1109/TCC.2015.2474369

Junghanns, M., Andr, #233, Petermann, Teichmann, N., G, K., . . . Rahm, E. (2016). *Analyzing extended property graphs with Apache Flink*. Paper presented at the Proceedings of the 1st ACM SIGMOD Workshop on Network Data Analytics, San Francisco, California.

Khalifa, S., Martin, P., Rope, D., McRoberts, M., & Statchuk, C. (2016, June 27 2016-July 2 2016). *QDrill: Query-Based Distributed Consumable Analytics for Big Data.* Paper presented at the 2016 IEEE International Congress on Big Data (BigData Congress).

Kobylinski, K., Bennett, J., Seto, N., Lo, G., & Tucci, F. (2014). *Enterprise application development in the cloud with IBM Bluemix.* Paper presented at the Proceedings of 24th Annual International Conference on Computer Science and Software Engineering.

Krulewich, D., Yin, J., Bundick, B. H., & Zeng, Y. (2015). *Performance assessment of real-time estimation of continuous-time stochastic volatility of financial data on GPUs*. Paper presented at the Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure, St. Louis, Missouri.

Leijon, A., Henter, G. E., & Dahlquist, M. (2016). Bayesian analysis of phoneme confusion matrices. *IEEE/ACM Trans. Audio, Speech and Lang. Proc., 24*(3), 469-482. doi:10.1109/taslp.2015.2512039

Liu, X., Golab, L., Golab, W., Ilyas, I. F., & Jin, S. (2016). Smart meter data analytics: Systems, algorithms, and benchmarking. *ACM Trans. Database Syst., 42*(1), 1-39. doi:10.1145/3004295

MadhaviLatha, A., & Kumar, G. V. Streaming Data Analysis using Apache Cassandra and Zeppelin.

Meaza, W. (2019). *Exploring the strategies big data analysts need to establish a diaspora analytics application.* Colorado Technical University,

Miškuf, M., Michalik, P., & Zolotová, I. (2017, 26-28 Jan. 2017). *Data mining in cloud usage data with Matlab's statistics and machine learning toolbox.* Paper presented at the 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI).

Olsson, J., & Asante, J. (2016). Using Node-Red to Connect Patient, Staff and Medical Equipment. In.

Pallonetto, F., Mangina, E., Finn, D., Wang, F., & Wang, A. (2014). *A restful API to control a energy plus smart grid-ready residential building: demo abstract*. Paper presented at the Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings, Memphis, Tennessee.

Racine, J. S. (2012). RStudio: A Platform-Independent IDE for R and Sweave. *Journal of Applied Econometrics, 27*(1), 167-172.

Sbrizzi, I. A sentiment analysis with cognitive app using Node-RED and Bluemix-Watson.

Soares, J., Canizes, B., Vale, Z., & Venayagamoorthy, G. K. (2016, 8-11 March 2016). *Benders' decomposition applied to Energy Resource Management in smart distribution networks.* Paper presented at the 2016 Clemson University Power Systems Conference (PSC).

Sun, Y., Bie, R., & Zhang, J. (2016). Measuring semantic-based structural similarity in multi-relational networks. *International Journal of Data Warehousing and Mining (IJDWM), 12*(1), 20-33.

Tian, Y., ÖZcan, F., Tao, Z. O. U., Goncalves, R., & Pirahesh, H. (2016). Building a hybrid warehouse: Efficient joins between data stored in HDFS and enterprise warehouse. *ACM Transactions on Database Systems, 41*(4), 1-38. doi:10.1145/2972950

Traub, J., Steenbergen, N., Grulich, P. M., Rabl, T., & Markl, V. I 2: Interactive Real-Time Visualization for Streaming Data.

Yeoh, W., & Popovič, A. (2016). Extending the understanding of critical success factors for implementing business intelligence systems. *Journal of the Association for Information Science & Technology, 67*(1), 134-147. doi:10.1002/asi.23366