# Overview of Stardog

Min Chen
Indiana University
School of Informatics, Computing, and Engineering
Bloomington, IN 47408, USA
mc43@iu.edu

## ABSTRACT

Graph databases with RDF data model have been used to represent knowledge with querying and reasoning capabilities. Stardog is a java-based commercial RDF graph database that supports SPARQL query languages, data unification using Virtual Graph and reasoning based on OWL, rules and Integrity Constraints. It provides enriched inference and reasoning beyond the property graph databases with Graph DBMS model and supports integration with cloud technologies such as Amazon Web Service and Pivotal Cloud Foundry.

## KEYWORDS

hid-sp18-405, Stardog, Virtual Graph, RDF, Graph Database

## 1 INTRODUCTION

Stardog is a graph database from US-software company Complexible. Stardog has a particular focus on OWL and RDF-based systems, and supports SPARQL query language; property graph model and Gremlin graph traversal language; OWL 2 and user-defined rules for inference and data analytic; virtual graphs; and programmatic interaction via several languages and network interfaces [19]. Further, the developers of Stardog OWL/RDF DBMS have pioneered a new use of OWL as a schema language for RDF databases. This is achieved by adding integrity constraints (IC), also expressed in OWL syntax, to the traditional open-world OWL axioms [7]. Other key features of Stardog include Machine Learning and Logical Inference,Semantic Search, Geospatial Search, Integration with Amazon AWS and Pivotal Cloud Foundry etc.

The technology paper is structured as follows:

- Section 2 presents the architecture of Stardog Knowledge Graph Platform, which is the integration of Stardog database with the knowledge toolkit.
- Section 3 discusses seven of the key features of Stardog.
- Section 4 compares Stardog with two similar graph databases, Neo4j and GraphDB, which are representatives of property graph databases and RDF graph databases respectively.
- Section 5 and Section 6 summarize the license of Stardog before the conclusion.

## 2 ARCHITECTURE

The architecture of Stardog Knowledge Graph Platform, which combines the graph database with knowledge toolkit, is shown in Figure 1.

[Figure 1 about here.]

There are three broad components centered around the Stardog graph database within the Knowledge Graph Platform, namely ETL, Virtual and Applications and Analytics. Each component is designed to provide the services in a declarative way.

- ETL stands for Extract, Transform, and Load, which are three database functions that are combined to extract data out of one database and insert into another database. Figure 1 illustrates that three main types of data: structured, semi-structured and unstructured are extracted and incorporated into the core graph database: Stardog.
- Virtual refers to the mapping of relational data into the RDF database as named graphs but without materialization (as in the ETL fashion) [3].
- Applications and Analytics include generating reports from the database, querying the database and perform analysis using statistical inference and probabilistic reasoning and also built-in machine learning libraries such as Vowpal, Wabbit and XGBoost [13] [14].

## 3 KEY FEATURES

In this section, the author discusses several key features of Stardog including Virtual Graph, Integrity Constraints, OWL and Rule Reasoning, Stardog Studio, Machine Learning, High Availability Cluster, Integration with AWS and PCF etc.

### 3.1 Virtual Graph

Virtual Graph is a feature that facilitates the mapping of relational data into the RDF databases. “Stardog supports the standard W3C R2RML mapping language [21] for defining how data in a relational system maps to RDF graphs” and “the mapped triples representing the source relational data are considered to be in a named graph that is not present (i.e., not materialized) in the local RDF graph” [3], hence, the named graph is considered virtual.

When dealing with unified data sources, users could either apply ETL (Extract, Transform, and Load) after materialization of the virtual graph or directly query the virtual graph using federated queries (virtual queries). Federated query performs a translation of a SPARQL query into a SQL query and the execution is through a relational database engine [2] [6]. Key trade-offs between these two operational models are summarized as follows:

- Evaluation of queries over materialized data via ETL does not involve any communication with the source system. This in general leads to better query performance and independence of queries from the availability of the source system [3].
- Materialization, on the other hand, takes multiple steps and resources for creating and storing tuples in RDF model, which is time-consuming. Further, when the data points are modified frequently before queried, materialization will

lead to a worse performance compared to virtual queries, which is essentially real-time reasoning.

Stardog offers both ways of unifying data, federated queries and materialization. The system allows users to switch between the two and the "choices can be made on a source by source basis" [3].

### 3.2 Integrity Constraints

In Stardog, Integrity Constraints (IC) are used to validate RDF data based on constraints or rules imposed by the database users. Stardog supports multiple languages for specifying the rules including SPARQL and OWL, which allows querying and mapping these rules in SPARQL as well [19]. Implementation of such constraints allows the users to apply domain-specific knowledge to the data and align the knowledge with RDF. Integrity Constraints can then be utilized in the reasoning procedure to ensure logical consistency and explain errors, which is the advantage of RDF database over plain property graph database in general.

### 3.3 OWL and Rule Reasoning

Stardogfis OWL reasoning is based on the OWL 2 Direct Semantics Entailment Regime and Stardog performs reasoning at query time without inference materialization. In addition, Stardog provides explanation of an inference by "minimum set of statements explicitly stored in the database that, together with the schema and any valid inferences, logically justify the inference" [19]. Under the circumstances where OWLfis axiom-based approach is not adequate for the reasoning, Stardog allows User-defined rules as a complements and enhance the power of the reasoning by combining both OWL and rules into the system.

### 3.4 Stardog Studio

Stardog Studio-the Knowledge Graph IDE, which is announced early 2018, is a front end developing tool for Stardog. It includes a SPARQL query notebook, which provides "syntax highlighting, prefix auto-completion, and exporting results" [18]. In addition, users could also "execute SPARQL queries against Stardog database and view results inside Stardog studio and export query results to the file system" [20].

Stardog Studio also provides the functionality of database management and security view. These allow the users to view and administer the Stardog databases as well as user, role and permissions for the Stardog system [18].

Additional features like visualization and cluster management tools are under development and expected in future releases [20].

### 3.5 Machine Learning

With the built-in machine learning libraries such as Vowpal, Wabbit and XGBoost, Stardog could perform traditional machine learning with statistical inference and probabilistic models [14].

Further, Machine Learning has been used in two unique ways supporting the knowledge graph. First, learning methods and algorithms are applied when creating the Knowledge Graph which unifies different data sources. Second, machine learning is also utilized to obtain actionable insight from the data unified. For example, predictive analytics is used to predict nodes and edges in a Knowledge Graph, and extract patterns from the data in order to make forecast based on those patterns [13].

### 3.6 High Availability Cluster

Stardog utilizes High Availability Clusters for uninterrupted operations, redundancy and high query volume [19]. The clusters aims at mitigating the risk of failure on a single machine by automatically creating multiple copies of the service with Apache ZooKeeper as the distributed coordination tool [16]. The cluster size affects performance in two ways: larger cluster sizes perform better for reads and perform worse for writes compared to small cluster sizes [19].

### 3.7 Integration with AWS and PCF

The Stardog High Availability Cluster supports installation by Stardog Graviton, which complies to a single binary executable. This facilitates the integration with Amazon Web Services and users could easily deploy, configure, and launch a Stardog cluster on Amazon AWS [5].

Besides, "the integration with Pivotal enables applications running in Pivotal Cloud Foundry to natively connect to Stardog instances without having to manually wire apes to services" [4]. This has been made available by the announcement of Stardog Service Broker for Pivotal Cloud Foundry.

## 4 COMPARISON WITH RELATED TECHNOLOGIES

Stardog is a graph database with RDF as a primary data model. Besides Stardog, there are other leading RDF graph databases including 4Store, GraphDB and Sesame. On the contrary, there is another type of graph databases, often referred to as property graph, that applies general Graph DBMS model without RDF. Neo4j is one of the leading technology in this category. In this section, Stardog will be compared to GraphDB and Neo4j, illustrating strengths and weaknesses of Stardog both within the category of RDF database and against the other category namely property graph database. A comparison of some of the system properties of the three graph databases are summarized in Table1

[Table 1 about here.]

### 4.1 Stardog vs. Neo4j

Neo4j, as a leading property graph database (ranking #1 by DB-Engines according to Table1), has strength in the following aspects. First, it is highly flexible that most objects and relations could be represented as nodes and edges respectively in the graph. Second, it does not require schema or ontology and thus light-weighted compared to RDF databases. Finally, it has a relative simple graph structure for traversals and analysis [17].

However, there are several important features of Stardog that property graph like Neo4j could not achieve. First, Neo4j only supports materialization of data. Virtualization of data (virtual graph) cannot be performed. Second, query language used by property languages such as Cypher and Gremlin lack the expressibility and ability to yield structured views of data compared to query languages like SPARQL [1]. Stardog uses SPARQL as a main query language and also supports all of Apache TinkerPop3 including

Gremlin [19]. Finally, without RDF and OWL, property graph cannot impose integrity constraints, explanations, user-defined rules or reasoning, which are all achievable in Stardog [10] [19].

### 4.2 Stardog vs. GraphDB

Both Stardog and GraphDB support RDF models and share many important features including reasoning, user-defined rules, SPARQL query and machine learning modules. However, Stardog has the capability of Virtual Graph which avoids materialization when unifying data sources, which is a key strength compared to GraphDB. GraphDB on the other hand, has a major advantage and focus on Natural Language Processing (NLP) and text mining by providing Ontotext Platform as an integrated text analysis system [15], while Stardog only supports text analytics indirectly by providing connectors to other NLP libraries OpenNLP [19].

Besides capabilities, researchers have been testing and comparing the performance of RDF graphs including GraphDB and Stardog. Based on experiments on real data, Ledvinka, Martin and Křemen concluded that "GraphDB, (and storages performing materialization in general) has a major disadvantage in that the user has to specify inference level before actually inserting data into the storages. Real time reasoning (like Stardog), on the other hand, lets the user choose reasoning level at the query time. However, GraphDB appears to be more suitable for the object-oriented application access scenario, in which frequent data updates are expected" [11]. In a more recent study, Luyen and his colleagues compared six RDF data models: 4Store, Virtuoso, Stardog, GraphDB, Sesame and Jena Fuseki (TDB) using large RDF graphs. They found that Stardog gives the best results compared to the criteria: Data Loading, Data Search and Data Inference therefore they stated that in general outperforms the other five candidates for their Benchmark [12].

## 5 LICENSE

As a commercial software, Stardog is priced for community, developer and enterprise tiers. The community version is free with 10 databases, 25 million triples per database and the developer version offers a free 30-day trial with unlimited data or machines [16]. The enterprise version comes with a server management module and customer support by both phone and email [19].

## 6 CONCLUSION

Stardog, as a commercial RDF-based graph database, supports data unification using both materialization and virtualization methods, and allows semantic reasoning and logical inferences by utilizing integrity constraints, OWL, and user-defined rules. The advantages of virtual graph, SPARQL query, and reasoning capability has made it an alternative to property graph databases, such as Neo4j.

## ACKNOWLEDGMENTS

The author would like to thank Dr. Gregor von Laszewski for his support and suggestions to write this paper.

## REFERENCES

[1] Renzo Angles and Claudio Gutierrez. 2008. The Expressive Power of SPARQL. In *Proceedings of the 7th International Conference on The Semantic Web (ISWC '08)*. Springer-Verlag, Berlin, Heidelberg, 114–129. https://doi.org/10.1007/978-3-540-88564-1_8 Accessed: 2018-03-16.

[2] Jess Balint. 2017. VIRTUAL GRAPHS IN STARDOG 5. blog. (May 2017). https://www.stardog.com/blog/virtual-graphs-in-stardog-5/ Accessed: 2018-03-16.

[3] Jess Balint. 2017. VIRTUAL GRAPHS: RELATIONAL DATA IN STARDOG. blog. (Jan. 2017). https://www.stardog.com/blog/virtual-graphs-relational-data-in-stardog/ Accessed: 2018-03-16.

[4] John Bresnahan. 2017. STARDOG AND PIVOTAL. blog. (May 2017). https://www.stardog.com/stardog-for-pivotal-cf/ Accessed: 2018-03-16.

[5] John Bresnahan. 2017. STARDOG GRAVITON: AWS MADE EASY. blog. (Jan. 2017). https://www.stardog.com/blog/stardog-graviton-aws-made-easy/ Accessed: 2018-03-16.

[6] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. 2017. Ontop: Answering SPARQL Queries over Relational Databases. *Semantic Web Journal* 8, 3 (2017), 471–487. https://doi.org/10.3233/SW-160217 Accessed: 2018-03-16.

[7] Karlis Cerans, Guntis Barzdins, Renars Liepins, Julija Ovcinnikova, Sergejs Rikacovs, and Arturs Sprogis. 2012. Graphical Schema Editing for Stardog OWL/RDF Databases using OWLGrEd/S.. In *OWLED (CEUR Workshop Proceedings)*, Pavel Klinov and Matthew Horridge (Eds.), Vol. 849. CEUR-WS.org. http://dblp.uni-trier.de/db/conf/owled/owled2012.html#CeransBLORS12 Accessed: 2018-01-28.

[8] Kendall Clark. 2017. WHAT IS A KNOWLEDGE GRAPH? blog. (June 2017). https://www.stardog.com/blog/what-is-a-knowledge-graph/ Accessed: 2018-03-16.

[9] DB-Engines. 2018. System Properties Comparison GraphDB vs. Stardog. www. (2018). https://db-engines.com/en/system/GraphDB%3BStardog Accessed: 2018-03-16.

[10] DB-Engines. 2018. System Properties Comparison Neo4j vs. Stardog. www. (2018). https://db-engines.com/en/system/Neo4j%3BStardog Accessed: 2018-03-16.

[11] Martin Ledvinka and Petr Křemen. 2015. Object-UOBM: An Ontological Benchmark for Object-Oriented Access. In *Knowledge Engineering and Semantic Web*, Pavel Klinov and Dmitry Mouromtsev (Eds.). Springer International Publishing, Cham, 132–146. Accessed: 2018-03-16.

[12] LE Ngoc Luyen, Anne Tireau, Aravind Venkatesan, Pascal Neveu, and Pierre Larmande. 2016. Development of a Knowledge System for Big Data: Case Study to Plant Phenotyping Data. In *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics (WIMS '16)*. ACM, New York, NY, USA, Article 27, 9 pages. https://doi.org/10.1145/2912845.2912869 Accessed: 2018-03-16.

[13] Pedro Oliveira. 2017. LEARNING TO PREDICT. blog. (April 2017). https://www.stardog.com/blog/learning-to-predict/ Accessed: 2018-03-16.

[14] Pedro Oliveira. 2018. BOOSTING MACHINE LEARNING. blog. (Feb. 2018). https://www.stardog.com/blog/boosting-machine-learning/ Accessed: 2018-03-16.

[15] Ontotext. 2017. Ontotext Platform. www. (2017). https://ontotext.com/products/ontotext-platform/ Accessed: 2018-03-16.

[16] PredictiveAnalyticsToday ReviewDesk. 2017. STARDOG. Web Page. (2017). https://www.predictiveanalyticstoday.com/stardog/ Accessed: 2018-03-16.

[17] Ian Robinson, Jim Webber, and Emil Eifrem. 2013. *Graph Databases*. O'Reilly Media, Inc. Accessed: 2018-03-16.

[18] Ty Soehngen. 2018. INTRODUCING STARDOG STUDIO. blog. (March 2018). https://www.stardog.com/blog/introducing-stardog-studio/ Accessed: 2018-03-16.

[19] Stardog Union. 2018. Stardog 5 THE MANUAL. Web Page. (2018). https://www.stardog.com/docs/ Accessed: 2018-01-28.

[20] Stardog Union. 2018. Stardog Studio. www. (2018). https://www.stardog.com/studio/ Accessed: 2018-03-16.

[21] W3C. 2012. R2RML: RDB to RDF Mapping Language. Web Page. (2012). https://www.w3.org/TR/r2rml/ Accessed: 2018-03-17.

## List of Figures

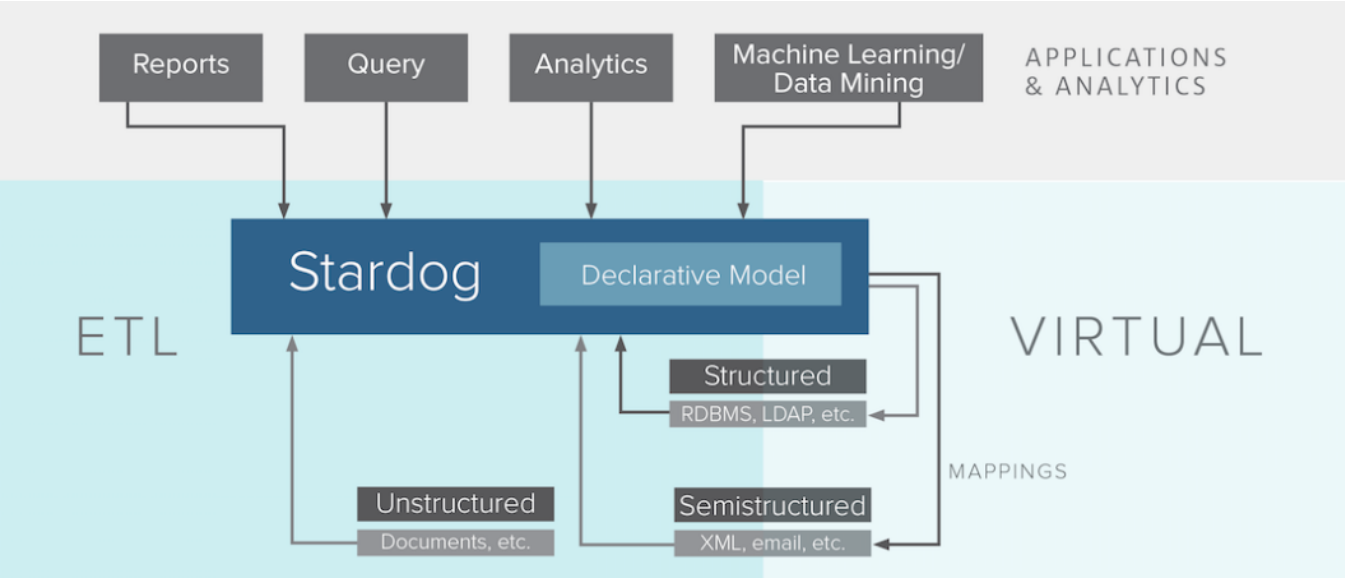


**Figure 1: Architecture of Stardog Knowledge Graph Platform [8]**

## List of Tables

**Table 1: System Properties Comparison GraphDB vs. Neo4j vs. Stardog [10] [9]**

| Name | GraphDB | Neo4j | Stardog |
|---|---|---|---|
| Database model | Graph DBMS and RDF | Graph DBMS | Graph DBMS and RDF |
| DB-Engines Ranking (Graph DBMS) | #12 | #1 | #10 |
| DB-Engines Ranking (RDF) | #7 | N/A | #6 |
| Developer | Ontotext | Neo4j, Inc. | Complexible Inc. |
| Initial release | 2000 | 2007 | 2010 |
| License | commercial | Open Source | commercial |
| Implementation language | Java | Java, Scala | Java |
| Any SQL supported | SPARQL | no | SPARQL |
| In-memory capabilities | no | no | yes |
| XML support | no | no | partially |