Using Apache Spark for Biological Data Processing

Tigran Shahinyan
Institute for Informatics and Automation Problems
Yerevan, Armenia
e-mail: tigranshahinyan@gmail.com

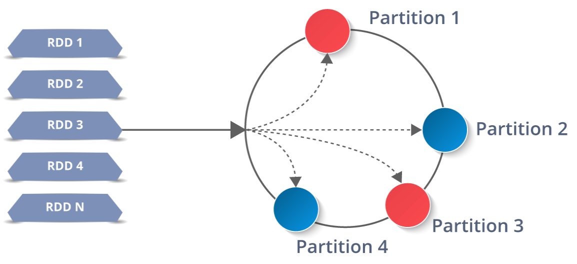
Levon Berberyan
Institute for Informatics and Automation Problems
Yerevan, Armenia
e-mail: levon711@mail.ru

ABSTRACT
Apache Spark is an open source distributed general-purpose cluster-computing framework. It's one of the most efficient technologies for processing massive amounts of distributed data. Spark provides DataSet and GraphX API-s which allow high level functions for manipulations of data distributed among the nodes of a cluster. It provides powerful optimization which considers the distributed nature of data. Currently there are sources of huge amounts of biological data which are publicly available for usage. One of such sources is UniProt providing a comprehensive, high-quality and freely accessible resource of protein sequence and functional information. Uniprot provides its semantic data in RDF format with a SPARQL interface for querying among it.
In current work we store UniProt's datasets into our cluster's distributed storage deployed in the cloud. We provide some basic functionality implemented in Spark using DataSet and GraphX API-s for Scala language to provide queries on the UniProt's data.

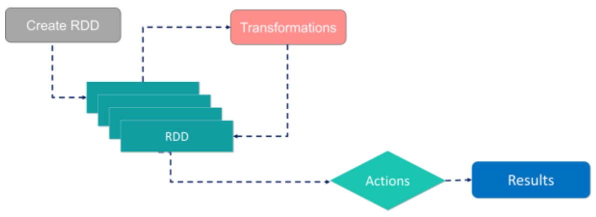
Keywords
Distributed computing, spark, semantic web.

1. INTRODUCTION
Heterogeneous distributed environments are being used for various computing tasks. One of the main challenges is efficient processing of large amounts of data which is also distributed.
There are many approaches and technologies for solving these tasks. MapReduce with its most widely used open source implementation Hadoop was one of the first successful frameworks for distributed computing [1,2]. But it turned to be slow when working with iterative tasks which demand running Map and Reduce jobs multiple times. Apache Spark is a better alternative for this type of computing tasks. Apache Spark is a unified engine for general purpose distributed computing based on the concept of resilient datasets [3]. It provides a stack of technologies for different types of distributed computing tasks. One of them is GraphX for distributed graph processing [4].
There are various techniques for data representation. Graph representation allows defining concepts and relationships between them. RDF is the Semantic Web's data representation framework which is now widely used to represent knowledge for interoperability between different communities and software [5].
In the current work we have used Uniprot's biological data in RDF representation and Apache Spark with DataSet and GraphX APIs for efficient distributed processing [6].

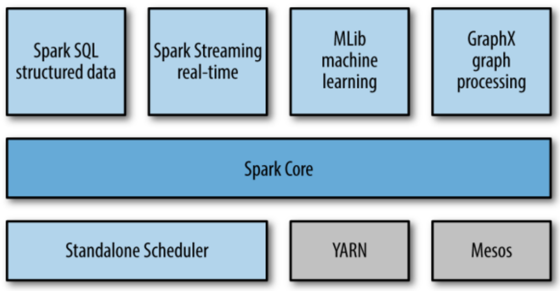
2. DISTRIBUTED DATA PROCESSING WITH APACHE SPARK
Apache Spark is based on the concept of Resilient Distributed Datasets. RDD is an immutable collection of data of different types. The power of Spark RDD-s is the distribution mechanism of the datasets and parallelization of a set of functional operators which can be applied on RDDs.



There are two types of functional operators: transformations and actions. Transformations are applied on existing RDDs and return a new RDD. Functions return values to the calling API in its representation. Transformations are computed lazily, the queries are optimized in DAGs and actually computed only when a function is called.



RDDs are immutable, which allows for easier distribution. When an RDD is broken in any step of the computing it can be recovered from previous steps and it's done automatically by the Spark engine.
The transformations can be divided into two groups: narrow and wide. Narrow transformations are the ones which do not require data transition among nodes, each chunk of the RDD is sufficient for the computation on a single node. Examples of narrow transformations are map, flatmap, filter, sample. Usage of narrow transformations is more preferable in distributed environments than wide transformations like reduceByKey, join, groupByKey, etc.
Spark provides a stack of APIs over RDD which provide optimization and API-s for various distributed computing tasks. Using Spark SQL is mostly much more efficient as the queries go through thorough optimization by the Catalyst optimizer [7]. Catalyst rewrites the query by bringing computing to the data as much as possible.



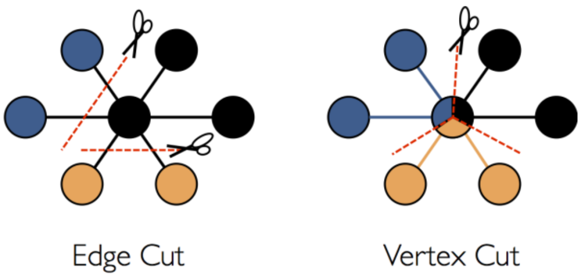
3. GRAPH REPRESENTATION AND PROCESSING OF DATA

Graph representation of data is widely used for describing various types of data from Web pages and their hyperlinks to people and their communities in social networks. It's also used to represent human and machine-readable information and share it among scientific communities. The Resource Description Format (RDF) and OWL (Web Ontology Language) are used to represent information resources modeled as a directed labeled graph, where edges represent the named link between two resources, represented by the graph nodes [8]. Resources and their properties are identified with URI references.

Uniprot provides a comprehensive, high-quality and freely accessible resource of protein sequence and functional information [6]. Some of its information sources are downloadable in RDF format.

We have used distributed RDF graphs from UniProt catalog. The data was stored in HDFS distributed file system in line based NTriples.

We have used Spark GraphX library for processing the data. GraphX provides a new API based on Spark RDD for processing graphs. It provides two new data structures for working with graphs: VertexRDD for vertex representation and EdgeRDD for edge representation. GraphX provides various optimizations, one which is vertex-cut approach to distributed graph partitioning. Rather than splitting graphs along edges, GraphX partitions the graph along vertices which can reduce both the communication and storage overhead. Logically, this corresponds to assigning edges to machines and allowing vertices to span multiple machines [4].



GraphX provides an API of functions and a bunch of useful algorithms to apply on graph data. Some of the most useful algorithms are PageRank, Triangle Counting, Component Counting.

Our research is based on using Spark DataSets and GraphX API to provide queries on RDF data stored in the distributed file system in the cluster. It's useful for aggregating descriptive data from a huge amount of information about human diseases, genes and protein structure and can be further developed and used for analytical purposes.

REFERENCES

[1] Jeffrey Dean, Sanjay Ghemawat, "MapReduce: A Flexible Data Processing Tool", *Communications of the ACM*, Volume 53 Issue 1, January 2010.

[2] Apache Hadoop, https://hadoop.apache.org/

[3] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, Ion Stoica, "Apache Spark: A Unified Engine For Big Data Processing", *Communications of the ACM*, November 2016, Vol. 59 No. 11, Pages 56-65

[4] Reynold S. Xin, Joseph E. Gonzalez, Michael J. Franklin, Ion Stoica, "GraphX: A Resilient Distributed Graph System on Spark", *Proceedings of the First International Workshop on Graph Data Management Experience and Systems (GRADES 2013)*, June 23, 2013, New York, New York, USA

[5] Berners-Lee, Tim; James Hendler; Ora Lassila, "The Semantic Web", *Scientific American Magazine*, May , 2001

[6] Universal Protein Resource. http://www.uniprot.org/.

[7] Michael Armbrust, Reynold S. Xin , Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, Matei Zaharia, "Spark SQL: Relational Data Processing in Spark", *SIGMOD'15*, May 31–June 4, 2015, Melbourne, Victoria, Australia

[8] W3C Semantic Web Activity, http://www.w3.org/2001/sw/