Chapter 9

Parallelizing Large-Scale Graph Algorithms Using the Apache Spark–Distributed Memory System

Ahmad Askarian
University of Texas at Dallas, USA

Rupei Xu
University of Texas at Dallas, USA

Andras Farago
University of Texas at Dallas, USA

ABSTRACT

The rapidly emerging area of Social Network Analysis is typically based on graph models. They include directed/undirected graphs, as well as a multitude of random graph representations that reflect the inherent randomness of social networks. A large number of parameters and metrics are derived from these graphs. Overall, this gives rise to two fundamental research/development directions: (1) advancements in models and algorithms, and (2) implementing the algorithms for huge real-life systems. The model and algorithm development part deals with finding the right graph models for various applications, along with algorithms to treat the associated tasks, as well as computing the appropriate parameters and metrics. In this chapter we would like to focus on the second area: on implementing the algorithms for very large graphs. The approach is based on the Spark framework and the GraphX API which runs on top of the Hadoop distributed file system.

DOI: 10.4018/978-1-5225-2814-2.ch009
INTRODUCTION

Spark is an open source, in-memory big data processing framework in a distributed environment. It started as a research program in 2009 and became an open source project in 2010. In 2014 it was released as an Apache incubator project (Xin et al, 2013).

Spark is evolved from Hadoop MapReduce so it can be run on Hadoop cluster and data in the Hadoop distributed File System (HDFS). It supports a wide range of workloads, such as Machine Learning, Business Intelligence, streaming and batch processing. Spark was created to complement, rather than replace Hadoop. The Spark core is accompanied by a set of powerful, higher-level libraries which can be used in the same application. These libraries currently include SparkSQL, Spark Streaming, MLlib (for machine learning), and GraphX, as shown in Figure 1.

In order to efficiently use the processing resources of a cluster, Spark needs a cluster resource manager. Yet Another Resource Negotiator (YARN) is a Hadoop processing layer that contains a resource manager and a job scheduler. Yarn allows multiple applications to run on a single Hadoop Cluster. Figure 2 illustrates how Spark uses Yarn as a distributed resource manager (Vavilapalli et al, 2013).

Figure 1. Spark full stack

![Spark full stack](image1.png)

Figure 2. Yet Another Resource Manager

![Yet Another Resource Manager](image2.png)
Related Content

A Security Framework for Enhancing User Experience
[www.igi-global.com/article/a-security-framework-for-enhancing-user-experience/179895?camid=4v1a](www.igi-global.com/article/a-security-framework-for-enhancing-user-experience/179895?camid=4v1a)

Security Issues in Mobile Wireless Networks
Arif Sari (2016). *Network Security Attacks and Countermeasures* (pp. 49-77).
[www.igi-global.com/chapter/security-issues-in-mobile-wireless-networks/143966?camid=4v1a](www.igi-global.com/chapter/security-issues-in-mobile-wireless-networks/143966?camid=4v1a)

Big Data Analysis and Implementation in Different Areas Using IoT
[www.igi-global.com/article/big-data-analysis-and-implementation-in-different-areas-using-iot/201094?camid=4v1a](www.igi-global.com/article/big-data-analysis-and-implementation-in-different-areas-using-iot/201094?camid=4v1a)

The Internet of Things: Enabling Artificial Intelligence
[www.igi-global.com/article/the-internet-of-things/210624?camid=4v1a](www.igi-global.com/article/the-internet-of-things/210624?camid=4v1a)