Preface

Overview

As the use of computers became widespread in the last twenty years, there has been an avalanche of digital data generated. The advent of digitization of all equipments and tools in homes and industry have also contributed to the growth of digital data. The demand to store, process and analyze this huge, growing data is answered by a host of tools in the market. On the hardware front the High Performance Computing (HPC) systems that function above tera-floating-point operations per second undertake the task of managing huge data. HPC systems needs to work in distributed environment as single machine cannot handle the complex nature of its operations. There are two trends in achieving the teraflop scale operations in a distributed way. Connecting computers via global network and handling the complex task of data management in distributed way is one approach. In other approach dedicated processors are kept close to each other thereby saving the data transfer time between the machines. The convergence of both trends is fast emerging and promises to provide faster, efficient hardware solutions to the problems of handling voluminous data.

The popular software solution to the problem of huge data management has been Apache Hadoop. Hadoop's ecosystem consists of Hadoop Distributed File System (HDFS), MapReduce framework with support for multiple data formats and data sources, unit testing, clustering variants and related projects like Pig, Hive etc. It provides tools for life-cycle management of data including storage and processing. The strength of Hadoop is that it is built to manage very large amounts of data through a distributed model. It can also work with unstructured data which makes it attractive. Combined with a HPC backbone, Hadoop can make the task of handling huge data very easy.

Today there are many high level Hadoop frameworks like Pig, Hive, Scoobi, Scrunch, Cascalog, Scalding and Spark that which make it easy to use Hadoop. Most of them are supported by well known organizations like Yahoo (Pig), Facebook (Hive), Cloudera (Scrunch) and Twitter (Scalding) demonstrating the wide

patronage Hadoop enjoys in the industry. These frameworks use the basic Hadoop modules like HDFS and MapReduce but provides an easy method to manage complex data processing jobs by creating an abstraction to hide the complexities of Hadoop modules. An example of such abstraction is Cascading. Many specific languages are built using the framework of Cascading. One such implementation by Twitter is called Scalding which it uses to query large data set like tweets stored in HDFS.

Data storage in Hadoop and Scalding is mostly disk based. This architecture impacts the performance due to long seek/transfer time of data. If data is read from disk and then held in memory where they can also be cached, the performance of the system will increase manifold. Spark implements this concept and claims it is 100x faster than MapReduce in memory and 10x faster on disk. Spark uses the basic abstraction of Resilient Distributed Datasets which are distributed immutable collections. Since Spark stores data in memory iterative algorithms in data mining and machine learning can be performed efficiently.

Objectives

The aim of this book is to present the required skills to set up and build large scale distributed processing systems using the free and open source tools and technologies like Hadoop, Scalding, Spark. The key objectives for this book include:

- Capturing the state of the art in building high performance distributed computing systems using Hadoop, Scalding and Spark
- Providing relevant theoretical software frameworks and practical approaches
- Providing guidance and best practices for students and practitioners of free and open source software technologies like Hadoop, Scalding and Spark
- Advancing the understanding of building scalable software systems for large scale data processing as relevant to the emerging new paradigm of High Performance Distributed Computing (HPDC)

Organization

There are 8 chapters in A Guide To High Performance Distributed Computing Case Studies with Hadoop, Scalding and Spark. These are organized in two parts.

Part I: Programming fundamentals of High Performance Distributed Computing

Chapter 1 covers the basics of distributed systems which form the backbone of modern HPDC paradigms like Cloud Computing, Grid/Cluster Systems. It starts by discussing various forms of distributed systems and explaining their generic architecture. Distributed file systems which form the central theme of such design are also covered. The technical challenges encountered in their development and the recent trends in this domain are also dealt with a host of relevant examples.

The discussion on the overview of Hadoop ecosystem in Chapter 2 is followed by a step-by-step instruction on its installation, programming and execution. Chapter 3 starts by describing the core of Spark which is Resilient Distributed Databases. The installation, programming API and some examples are also covered in this chapter. Hadoop streaming is the focus of Chapter 4 which also covers working with Scalding. Using Python with Hadoop and Spark is also discussed.

Part II: Case studies using Hadoop, Scalding and Spark

That the current book does not limit itself to explaining the basic theoretical foundations and presenting sample programs is its biggest advantage. There are four case studies presented in this book which covers a host of application domains and computational approaches so as to convert any doubter into a believer of Scalding and Spark. Chapter 5 takes up the task of implementing K-Means Clustering Algorithm while Chapter 6 covers data classification problems using Naive-Bayes classifier. Continuing the coverage of data mining and machine learning approaches in distributed systems using Scalding and Spark, regression analysis is covered in Chapter 7.

Recommender systems have become very popular today in various domains. They automate the task of middleman who can connect two otherwise disjoint entities. This is becoming much needed feature in all modern networked applications in shopping, searching and publishing. A working recommender system should not only have a strong computational engine but should also be scalable at real-time. Chapter 8 explains the process of creating such a recommender system using Scalding and Spark.

Target Audience

A Guide To High Performance Distributed Computing Case Studies with Hadoop, Scalding and Spark has been developed to support a number of potential audiences, including the following:

- Software Engineers and Application Developers
- Students and University Lecturers
- Contributors to Free and Open Source Software
- Researchers

Code Repository

The complete list of source code and datasets used in this book can be found here `https://github.com/4ni1/hpdc-scalding-spark`

Bangalore, India *Srinivasa K G*
September 2014 *Anil Kumar Muppalla*

http://www.springer.com/978-3-319-13496-3

Guide to High Performance Distributed Computing
Case Studies with Hadoop, Scalding and Spark
Srinivasa, K.G.; Muppalla, A.K.
2015, XVII, 304 p. 43 illus., Hardcover
ISBN: 978-3-319-13496-3