¡Insert Title¿
IN4355 - Functional Programming

David Hartveld - 1181092
Johan Laanstra - 1509268
Michael de Jong - 1314793

November 8, 2012

1 Introduction

Functional programming is becoming increasingly more used in programming languages. Besides functional languages such as Scheme and Haskell, relatively new languages like Javascript and Scala are also adding features from functional programming to their repetoire. Another example of this is LINQ (Language Integrated Query) and its use of lambdas in C#, but there are lots of other non-functional languages who support some form of functional programming constructs. To get familiar with functional programming principles and languages we built a small grid computing framework to do map-reduce computations. MapReduce is in spired by the map and reduce functions commonly used in functional programming.

This article focuses on the functional aspects, details, pitfalls and shortcomings of languages we came accross while building the grid computing framework with functional languages. We will discuss the implementation of a simple map-reduce algorithm and how we distributed the map and reduce steps over several browser clients. We chose word count as the algorithm to implement, because of its simplicity and the clearly defined operations for the map and reduce steps. Johan

2 Map-Reduce

MapReduce is a (functional) programming model for processing large amounts of data over a distributed network of nodes. Probably the most known implementation of MapReduce is Hadoop [**?**]. Data can be manipulated by defining a map and reduce funtion. A map function transforms data from one form into another and a reduce function combines multiple data entries into a single entry. MapReduce essentially works by splitting the potentially huge input data up into several partitions. Each partition is then sent off to a different worker in the network, which in turn applies the map function on the data partition.

This is called the map phase. The result of the map operation is usually stored temporarily on disk. Once all the partitions have been processed by the nodes in the network, the reduce phase is started. The nodes all read a specific range of the output from the map phase, and then apply the reduce function on that data. The resulting data is sent to an output file. An overview is given in figure 1. Once the framework which implements MapReduce has completed both phases the result can be read. This programming model allows developers to solve problems quickly while leveraging a cluster of computers, which with a single computer would take significantly longer. Obviously there is a penalty for having to use the network, but this is offset by the fact that the map-reduce cluster is capable of solving problems with very large input sets.

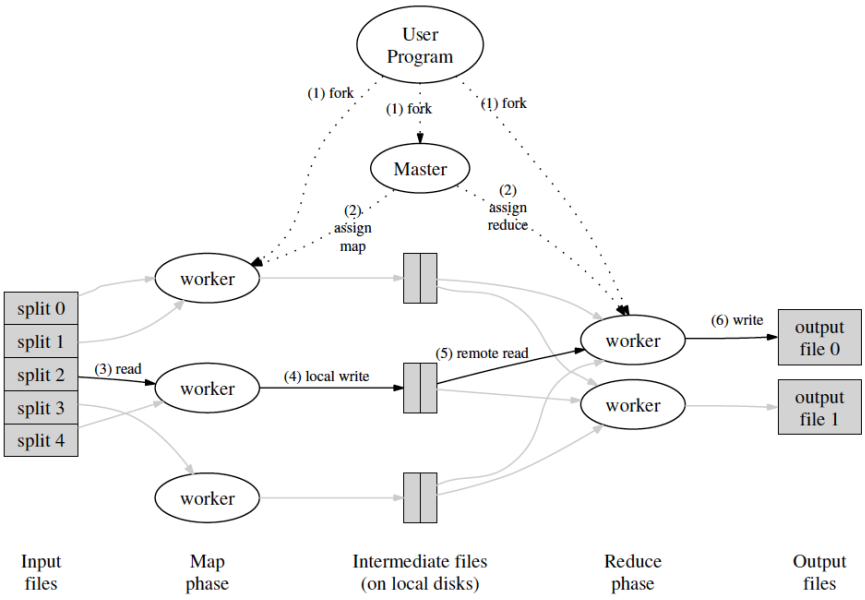

Figure 1: Image from: MapReduce: Simplified Data Processing on Large Clusters, 2004, by Jeffrey Dean and Sanjay Ghemawat

Michael

3 Grid computing

Grid computing is a form of distributed computing where clients donate computing power to solve problems that to big to solve for a single machine. One of the most famous grid computing projects is SETI@Home (Search for Extraterrestrial Intelligence), in which PC users donate unused CPU cycles to find signs of extraterrestrial life in signals from outer space. Such a grid consists of participating clients and can be seen as a virtual supercomputer. The clients in

a grid are usually not connected to each other, but in stead communicate with a master node or some other sort of scheduler.

Grid computing has several advantages compared to supercomputers. Since programs run on normal computer hardware, no special software is needed. Also debugging is a lot easier since this can be done on a single client. Grid computing eliminates the complexity of shared memory and shared storage space, because this approach forces the developers to think differently about their storage and memory.

Johan

4 So we heard you like platforms.

4.1 Components

David

4.2 How does it work?

Johan

5 Fay: Fay ain't yavascript

5.1 Overview

Our initial intention was to limit ourselves to the use of Fay [**?**] in stead of javascript. Fay is a programming language which contains the grammer and syntax of the functional programming language Haskell, and some additional grammer and syntax which enables you to store variables and interact with javascript. Before you can use Fay, you have to install the Fay compiler. Once that is installed you can compile Fay source files into Javascript source files using the command line.

5.2 Problems

The documentation on the Fay website seems to be outdated, as the compiler arguments have changed. Luckily the Fay compiler can tell you what the command line options are. However this was not our only problem with Fay. Some of the compiler's features did not work as specified. Supposedly it should be possible to format/beautify the produced Javascript code using the command line argument "–pretty", but this doesn't seem to be doing anything. The same applies to the "–library" argument which ensures that in the resulting Javascript source code, the main function is not immediately called.

But most problematic of all was the fact that Fay produces Javascript code which is wrapped in a function scope to avoid leaking scope. This means that we cannot call any functions compiled by Fay from external Javascript source

code. Even if this scope problem was fixed, we'd still be unable to do this since the generated Javascript functions have human-unreadable names like "626261". Because of this we've had to implement most of the client code dealing with the communication and computation in Javascript in stead of Fay. Michael

6 Distributed sort

Johan

7 Scala

David Michael

8 Future work

David

References

9 Appendix A: Analysis of Game of Thrones word count