**Script:**

Role:

My role was to build the Raspberry PI cluster, install and learn Apache Spark as well as Scala, and develop the software to process the data produced by Kyle and Matsobane, which would ultimately be used by the presentation framework developed by Dave.

Looking at the Approach:

We wanted to distribute the work among many, commodity based, computers instead of investing in expensive, powerful single machines. Parallel computing allows us to share the amount of work among many machines, possibly allowing us to perform large sets of work in a shorter amount of time with fault tolerance.

Tools & Technologies:

Hardware:

We decided to go with a low-cost, low-power infrastructure consisting of five Raspberry PI 3’s, two 5 port switches and one 450-watt PC PSU. We wanted to show that you can build a data processing computing cluster using commodity hardware and get reasonable performance, making it a viable choice for start-ups or existing small businesses, as well we academia.

Software:

All the software used to develop and run the applications on the computing cluster is open source (apart from Oracle’s Java 8 JDK). The operating systems on the PI’s are a derivate of Debian Linux, the Apache Spark framework allowing us to run the PI’s as a cluster computing system is open source under the Apache 2.0 license, Scala is open source under a BSD-style license and JetBrains’ IntelliJ IDEA Community Edition is licensed under Apache 2.0

Design:

Each of the Scala applications that processed the data focused on different resolutions of time. The CategoryCountPerHour application read the tweet data, sorted it by time and then searched for keywords within an hourly bound context. The resultant data from this program then fed into the CategoryCountPerDay application, which then further aggregated the data to show the number of tweets per keyword per day. The CategoryCountTotal program then further aggregated that data into a total count.

Challenges:

Having to learn Scala and its libraries, learning to think in functional programming terms with regards to map-reduce, and installing, configuring and running Apache Spark in a low-memory environment like the Raspberry PI’s. I had never worked with any of these languages or technologies before.

Conclusion:

All this hardware and software is available to anybody interested in Big Data processing. The hardware is cheap and the software is free. The learning curve in the beginning can be quite steep but is ultimately very rewarding in terms of what you can achieve with so little financial investment.