EMS Annual Meeting Abstracts
Vol. 14, EMS2017-484, 2017

Low-cost and scalable computing framework for meteorological data

Laila Daniel, Jussi S. Ylhäisi, Mikko Rauhala, and Tarja Riihisaari
Finnish Meteorological Institute, Helsinki, Finland (jussi.ylhaisi@fmi.fi)

With the ever-increasing amount of weather data available for timely and accurate predictions, the rise of Big Data framework such as Apache Spark can provide scalable and low-cost solutions for various meteorological applications.

Apache Spark is an open source project that started in 2010. It provides a unified distributed data processing framework that incorporates a variety of processing workloads to support both batch and interactive processing. A basic data sharing abstraction in Spark called "Resilient Data sets" (RDDs) provides optimization, fault-tolerance, scalability and ease of composition for the various processing tools for developing applications. Apache Spark supports libraries in Scala, Java, Python and R that allow applications to readily combine SQL, machine learning, streaming and graph processing. Apache Spark can be deployed in a standalone machine or can be connected to Hadoop filesystem, or to cloud storages such as Amazon EC2. Numerous deployments of applications based on this framework have been reported in the recent literature.

We setup a simple Apache Spark framework using a single laptop with 8 CPUs, each with 4 cores and a memory of 16 GB. We apply this framework for Model Output Statistics, where multiple linear regression algorithm is applied to the ECMWF HRES forecasts and the surface temperature T2 is estimated. We use the linear regression algorithm implemented in the machine learning library of Apache Spark. We use SparkR, an R frontend for Spark. R supports advanced data analysis on data frames and has a large number of packages on statistical analysis and data visualization. As R is single threaded, processing large datasets using R results in poor performance. SparkR, a combination of Spark and R libraries, constitutes a flexible tool for statistical analysis of Big Data and is highly suited for several meteorological applications. In our experimental study, we notice that when the number of observations are more than 8 million, linear regression done with SparkR takes less time in estimating T2 compared to the time taken using built-in fitting in R.

We initially plan to extend our simple framework with one laptop to a cluster of four laptops. We also plan to experiment with other machine learning algorithms such as principal component regression, K-Means and similarity algorithms. These algorithms are readily available in R and have to be adapted for SparkR.