Petri Net Models to Estimate Hadoop Performance

Nowadays more and more companies deal with large amounts of raw, unstructured data during business operations. This trend is fostering the emergence of the Big Data market, which is growing at a fast 27% worldwide compound annual growth rate through 2017 and, in Europe, at a 31.96% compound annual growth rate through 2016. Moreover, nearly 40% of Big Data worldwide will likely be hosted on public Clouds by 2020, while Hadoop is expected to touch half of world data during the same period.

Apache Hadoop, open source implementation of the MapReduce framework, holds a central role in the Big Data paradigm and is widely adopted in the industry to process huge datasets. Alongside Hadoop, with its I/O bounded workflow mainly targeted at batch processing, currently in-memory frameworks such as Spark are available, enabling faster elaboration of iterative algorithms, e.g., regression, classification, and other machine learning applications. Cost effectiveness considerations encourage to share computational clusters among heterogeneous classes of workloads, but this practice gives rise to difficulties in performance prediction. Furthermore, real world applications are usually bound to meet Service Level Agreements providing, e.g., an upper bound for query execution time, thus requiring careful resource allocation.

Among other approaches, it is possible to study multi-class systems adopting Petri Nets. At the expense of a significant computational complexity, these tools allow for a great accuracy in performance prediction. In addition, Petri Nets appear good abstractions for data-intensive applications: a token circulating in the model represents well a request being processed and atomic fork/join operations and colors can be profitably exploited to express at the same time the memory, disk read/write operations, network/stream traffic, and other concurrent operations that a single request implies on the available computational resources. Then the results will be exploited to solve the capacity allocation problem and to assess possible advantages of admission control.

The aim of this thesis is the development and validation of Petri Nets models through an experimental campaign conducted on Hadoop and Spark, so as to quantify the obtained precision in performance prediction and to reach a good balance with simulation times. In particular, these models will address the multi-class admission control and capacity allocation problem, with possible extensions to design time or runtime applications, such as tools for optimal resource sizing, application development, or scheduling.