

LITERATURE REVIEW: XRT 0.1.0 - A Language-Agnostic Map-Reduce Runtime for Shared-Memory Systems

Erik Selin

School of Information Technology and Engineering

University of Ottawa

Ottawa, Canada K1N 6N5

erik.selin@gmail.com

October 7, 2017

1 Introduction

Map-reduce is a restrictive programming model introduced in 2004 by Google [3] and later popularized by the open-source Apache project Hadoop MapReduce developed by a team at Yahoo! [1]. At a high level, map-reduce enables developers to write efficient and highly parallelizable data processing jobs without having to deal with the complexities of parallel programming. Following the initial success of Hadoop MapReduce and the growing need for processing of so called big data multiple map-reduce runtimes have been developed for various environments. Map-reduce is becoming particularly interesting for shared-memory systems as core counts and memory availability is steadily increasing while cost is going down.

Multiple map-reduce runtimes have been developed for shared-memory systems but the focus of these systems is almost entirely to compete for best in-memory benchmarks while sacrificing usability, simplicity and cross platform compatibility [?] [?] [?]. Processing of data that does not fit in system memory is rarely if ever discussed in the current shared-memory map-reduce runtimes. In addition all of these runtimes forces the programmer to implement data processing jobs in C++ which is simply not an option for most engineering teams. Attempts have been made to bring a more flexible map-reduce runtime through the Hadoop Streaming project [2] but since it is built upon Hadoop MapReduce it suffers from major performance issues [?]. Attempts have been made to speed-up Hadoop MapReduce for shared-memory systems [?] and to add support for languages to Hadoop MapReduce [?] [?] [?]. Unfortunately none of these projects have yielded a performant language-agnostic shared-memory runtime.

XRT is a response to these perceived shortcomings in the shared-memory map-reduce community. It is a map-reduce runtime built from the ground up for high performance on shared-memory systems while offering a completely language-agnostic API. XRT is memory-first but disk-aware and uses the streaming model from Hadoop Streaming [] but with a runtime inspired by Phoenix, CilkMR and other shared-memory map-reduce runtimes [].

2 Literature Review

2.1 The map-reduce programming model

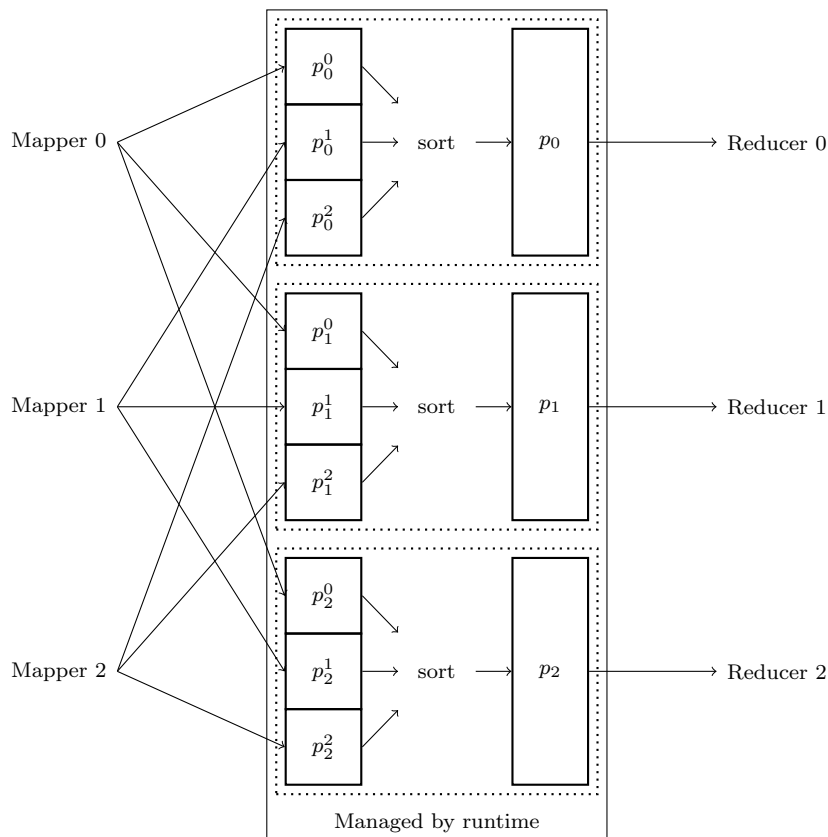


Figure 1: In map-reduce runtimes the programmer provides the mapper and reducer code while input, output, execution of mapper code, execution of reducer code, shuffling and sorting is handled by the runtime.

2.2 Map-reduce on GPUs, FPGA and Coprocessors

2.3 Map-reduce on shared-memory runtimes

XRT is a response to these perceived shortcomings in the shared-memory map-reduce community. It is a map-reduce runtime built from the ground up for high performance on shared-memory systems while offering a completely language-agnostic API. XRT is memory-first but disk-aware and uses the streaming model from Hadoop Streaming [1] but with a runtime inspired by Phoenix, CilkMR and other shared-memory map-reduce runtimes [2, 3].

XRT is a response to these perceived shortcomings in the shared-memory map-reduce community. It is a map-reduce runtime built from the ground up for high performance on shared-memory systems while offering a completely language-agnostic API. XRT is memory-first but disk-aware and uses the streaming model from Hadoop Streaming [1] but with a runtime inspired by Phoenix, CilkMR and other shared-memory map-reduce runtimes [2, 3].

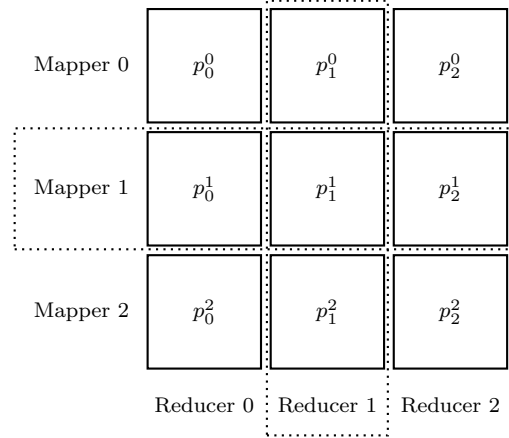


Figure 2: In shared-memory map-reduce systems memory is usually split into a matrix. Mappers are given access to a single row and partitions data into the entries of that row. Reducers are given access to a single column where all entries represents the same partition.

2.4 Language-agnostic map-reduce

2.5 Data over standard streams

References

- [1] Hadoop. <http://hadoop.apache.org/>.
- [2] Hadoop streaming. <http://hadoop.apache.org/docs/current/hadoop-streaming/HadoopStreaming.html>.
- [3] J Dean and S Ghemawat. Mapreduce: simplified data processing on large clusters. osdi04 proceedings of the 6th conference on symposium on operating systems design and implementation dalam: International journal of engineering science invention. *URL: http://static.googleusercontent.com/media/research.google.com/diunduh_pada_2015-05-10*, pages 10–100, 2004.