MapReduce modules to introduce parallel and distributed computing concepts

Weiwei Ge, David John, Stan Thomas

Wake Forest University
Department of Computer Science

Introduction

There is little debate regarding the necessity to include parallel and distributed (PDC) concepts in the undergraduate computer science (CS) curriculum. However there may be debate as to the best way to do so; the extent of coverage necessary; and the best approach to accomplishing one's specific goals. At Wake Forest, our approach has been to develop PDC modules for use in existing courses rather than adding PDC courses to our core requirements. We have previously developed and employed modules built on a message passing paradigm [2]. In the current work we report on efforts to develop *MapReduce*-based[1] instructional modules for introductory level courses. We believe that with this approach we can introduce students to important PDC concepts earlier in their studies. This early introduction to MapReduce can be very beneficial not only to CS-majors but also to students pursuing programs in other areas disciplines such as business or the social sciences.

Acknowledgement

This work supported in part by the NSF/IEEE –TCPP Early Adopter Initiative..

MapReduce Instructional Modules

The tasks chosen for the two instructional modules described here involve text processing. The elementary problem asks "for a large set of documents, can we produce an index that associates words with the documents in which they occur?" This inverted index problem lends itself readily to one application of MapReduce and is appropriate for students completing CS1 in Java.

The second, intermediate, problem asks "for a large set of articles from the student newspaper, which is most similar to an article of interest?" The second problem is an extension of the first and builds on its solution, requiring four passes of MapReduce. Understanding the solution to this problem requires knowledge of basic data structures and is targeted to students studying algorithms.

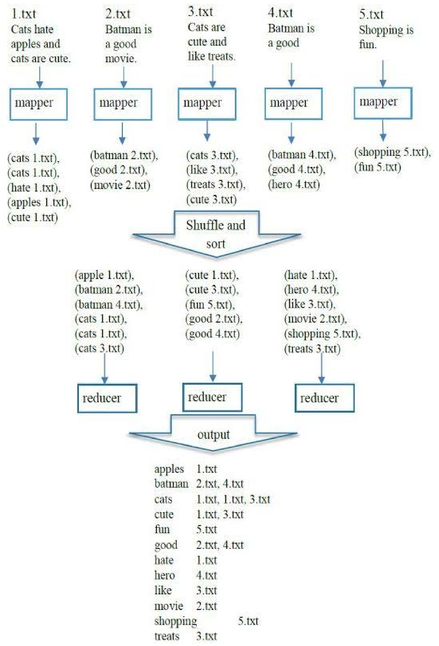

Figure 1. Decomposition of MapReduce algorithm indicating mappers, reducers and <key, value> pairs for the elementary module.

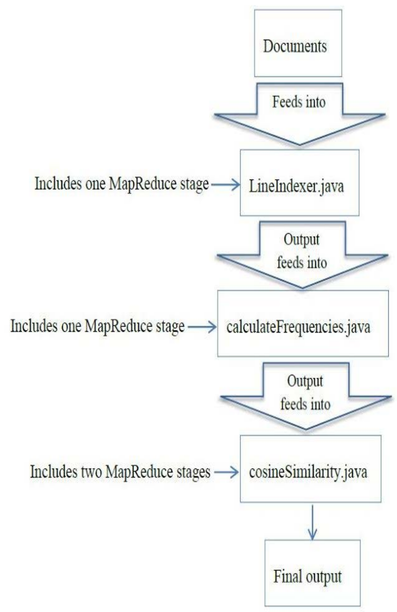

Figure 2. Data flow diagram for the intermediate module

Modules

For our purposes, a PDC module consists of three components: a written document, an instructor and evaluated student work. Each module is designed to be completed in three contact hours. A document is distributed to each student that states the goals of the module, introduces the PDC topic algorithmically and shows executable source code that implements the algorithm. The instructor is an important part of the module as there will be questions from the students that need to be answered. Student work, from reading and understanding the written document, to implementation, is crucial to students meeting the goals of a module. Students invest more in their work when their efforts are evaluated.

References

[1] Apache MapReduce Tutorial. https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html.

[2] D. J. John and S. J. Thomas. Parallel and distributed computing across the computer science curriculum. In *Parallel Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International*, pages 1085–1090, May 2014.