Accepted Manuscript

Locality-aware and load-balanced static task scheduling for MapReduce

Oguz Selvitopi, Gunduz Vehbi Demirci, Ata Turk, Cevdet Aykanat

PII: S0167-739X(17)32926-6
DOI: https://doi.org/10.1016/j.future.2018.06.035
Reference: FUTURE 4297

To appear in: *Future Generation Computer Systems*

Received date : 24 December 2017
Revised date : 5 May 2018
Accepted date : 20 June 2018

Please cite this article as: O. Selvitopi, G.V. Demirci, A. Turk, C. Aykanat, Locality-aware and load-balanced static task scheduling for MapReduce, *Future Generation Computer Systems* (2018), https://doi.org/10.1016/j.future.2018.06.035

Locality-aware and load-balanced static task scheduling for MapReduce☆

Oguz Selvitopi[a], Gunduz Vehbi Demirci[a], Ata Turk[b], Cevdet Aykanat[a,*]

[a]*Bilkent University, Computer Engineering Department, 06800, Ankara, TURKEY*
[b]*Boston University, ECE Department, Boston, MA 02215*

Abstract

Task scheduling for MapReduce jobs has been an active area of research with the objective of decreasing the amount of data transferred during the shuffle phase via exploiting data locality. In the literature, generally only the scheduling of reduce tasks is considered with the assumption that scheduling of map tasks is already determined by the input data placement. However, in cloud or HPC deployments of MapReduce, the input data is located in a remote storage and scheduling map tasks gains importance. Here, we propose models for simultaneous scheduling of map and reduce tasks in order to improve data locality and balance the processors' loads in both map and reduce phases. Our approach is based on graph and hypergraph models which correctly encode the interactions between map and reduce tasks. Partitions produced by these models are decoded to schedule map and reduce tasks. A two-constraint formulation utilized in these models enables balancing processors' loads in both map and reduce phases. The partitioning objective in the hypergraph models correctly encapsulates the minimization of data transfer when a local combine step is performed prior to shuffle, whereas the partitioning objective in the graph models achieve the same feat when a local combine is not performed. We show the validity of our scheduling on the MapReduce parallelizations of two important kernel operations–sparse matrix-vector multiplication (SpMV) and generalized sparse matrix-matrix multiplication (SpGEMM)–that are widely encountered in big data analytics and scientific computations. Compared to random scheduling, our models lead to tremendous savings in data transfer by reducing data traffic from several hundreds of megabytes to just a few megabytes in the shuffle phase and consequently leading up to 2.6x and 4.2x speedup for SpMV and SpGEMM, respectively.

Keywords: MapReduce, scheduling, data locality, load balance, map task, reduce task

1. Introduction

MapReduce [1] simplifies the programming for large-scale data-parallel applications and greatly reduces the development effort by sparing the programmer from complex issues such as parallel execution, fault tolerance, data management, task scheduling, etc. Hadoop [2], an open source implementation of MapReduce, has been used in production environments of many big companies and is deployed on clusters that can scale up to tens of thousands of cores. Its generality, ease of use and scalability led to a wide acceptance and adoption in many fields.

A MapReduce job consists of map, shuffle and reduce phases which are carried out one after another by multiple parallel tasks that process data in parallel. The map tasks process the input data and emit $\langle key, value \rangle$ (KV) pairs. In the shuffle phase, the KV pairs are communicated and then they are sorted according to keys, thus grouping the values that belong to the same key. The reduce tasks then process the

☆This work was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under Grant EEEAG-115E212 and ICT COST Action IC1406 (cHiPSet).

*Corresponding author

Email addresses: reha@cs.bilkent.edu.tr (Oguz Selvitopi), gunduz.demirci@cs.bilkent.edu.tr (Gunduz Vehbi Demirci), ataturk@bu.edu (Ata Turk), aykanat@cs.bilkent.edu.tr (Cevdet Aykanat)