Accepted Manuscript

Enabling fast failure recovery in shared Hadoop clusters: Towards failure-aware scheduling

Orçun Yildiz, Shadi Ibrahim, Gabriel Antoniu

PII: S0167-739X(16)30028-0
DOI: http://dx.doi.org/10.1016/j.future.2016.02.015
Reference: FUTURE 2968

To appear in: *Future Generation Computer Systems*

Received date: 26 September 2015
Revised date: 14 January 2016
Accepted date: 20 February 2016

Please cite this article as: O. Yildiz, S. Ibrahim, G. Antoniu, Enabling fast failure recovery in shared Hadoop clusters: Towards failure-aware scheduling, *Future Generation Computer Systems* (2016), http://dx.doi.org/10.1016/j.future.2016.02.015

Available online at www.sciencedirect.com

ScienceDirect

Future Generation Computer Systems 00 (2016) 1–19

Future Generation Computer Systems

Enabling Fast Failure Recovery in Shared Hadoop Clusters: Towards Failure-Aware Scheduling

Orçun Yildiz, Shadi Ibrahim*, Gabriel Antoniu

INRIA Rennes - Bretagne Atlantique Research Center, France

Abstract

Hadoop emerged as the *de facto* state-of-the-art system for MapReduce-based data analytics. The reliability of Hadoop systems depends in part on how well they handle failures. Currently, Hadoop handles machine failures by re-executing all the tasks of the failed machines (i.e., executing recovery tasks). Unfortunately, this elegant solution is entirely entrusted to the core of Hadoop and hidden from Hadoop schedulers. The unawareness of failures therefore may prevent Hadoop schedulers from operating correctly towards meeting their objectives (e.g., fairness, job priority) and can significantly impact the performance of MapReduce applications. This paper presents Chronos, a failure-aware scheduling strategy that enables an early yet smart action for fast failure recovery while still operating within a specific scheduler objective. Upon failure detection, rather than waiting an uncertain amount of time to get resources for recovery tasks, Chronos leverages a lightweight preemption technique to carefully allocate these resources. In addition, Chronos considers data locality when scheduling recovery tasks to further improve the performance. We demonstrate the utility of Chronos by combining it with Fifo and Fair schedulers. The experimental results show that Chronos recovers to a correct scheduling behavior within a couple of seconds only and reduces the job completion times by up to *55%* compared to state-of-the-art schedulers.

Keywords: Data Management; Scheduling; Failure; MapReduce; Hadoop, Preemption

1. Introduction

Due to its simplicity, fault tolerance, and scalability, MapReduce [1] is by far the most powerful programming model for data intensive applications. The popular open source implementation of MapReduce, Hadoop [2], was developed primarily by Yahoo!, where it processes hundreds of terabytes of data on at least *10,000 cores*, and is now used by other companies, including Facebook, Amazon, Last.fm, and the New York Times [3].

Failures are part of everyday life, especially in today's data-centers, which comprise thousands of commodity hardware and software devices. For instance, Dean [4] reported that in the first year of the usage of a cluster at Google there were around a thousand individual machine failures and thousands of hard drive failures. Consequently, MapReduce was designed with hardware failures in mind. In particular, Hadoop handles machine failures (i.e., fail-stop failure) by re-executing all the tasks of the failed machines (i.e., executing recovery tasks), by leveraging data replication.

*Corresponding author
Email address: shadi.ibrahim@inria.fr (Shadi Ibrahim)