Huawei Open Data Analytics Platform

Gary Verhaegen*, Charles Bonneau*, Antonios Tsaltas*

*Avenue Albert Einstein, 2a, B-1348 Louvain-la-Neuve
gary.verhaegen,charles.bonneau,antonios.tsaltas@huawei.com,
http://www.huawei.com

Abstract. New frameworks such as Spark, Tez, Flink, or Hive offer new possibilities, but using more than one framework is generally not easy. We build an abstraction layer that allows users to define their big data operations in a declarative way. This abstraction layer is backed by a platform that optimizes the workflow by carefully profiling each operation and running them, transparently, on the most appropriate framework given the description of the job and the currently available resources.

1 Introduction

For a time, it looked like the world would settle on Apache Hadoop for all its big data needs, but the Hadoop project has been split, and while its data storage part, HDFS, has indeed become the lingua franca of big data processing frameworks, its processing engine, Hadoop MapReduce, has seen a lot of competition lately. New frameworks such as Apache Spark (see Zaharia et al. (2010)), Apache Tez (based on Verma et al. (2011)), Apache Flink (continuation of Warneke and Kao (2009)), or Apache Hive offer new possibilities, along with tradeoffs and challenges.

In this work, we build an abstraction layer that allows users to define their big data operations in a declarative way. This abstraction layer is backed by a platform that optimizes the workflow by carefuly profiling each operation and running them, transparently, on the most appropriate framework given the description of the job and the currently available resources.

In Section 2, we describe the high-level API exposed to users and the general principles behind it. In Section 3, we describe the architecture of the underlying platform. Section 4 presents our demonstration scenario, and, finally, Section 5 concludes.

2 High-Level API

We expose an API rather than a GUI for our platform. This has multiple advantages: we can focus on the platform itself, support multiple clients in the future, and allow other programs to interact with the platform. These other programs may be IDEs, including textual and graphical DSL frontends.

The API is based upon the REST principles, as described in Fielding (2000). The central resource type is a dataset, which presents the user with metadata about an actual dataset present