RHEEM: Enabling Cross-Platform Data Processing (Paper Critique)

Organizations normally perform complicated and expensive operations to move their codes and information across different platforms. For this reason, businesses have to go beyond the boundaries of single data processing platform such as DBMS and Hadoop. This led to the invention of RHEEM, this is a general-purpose cross-platform system for data processing that decouples applications from underlying platforms. It features include, an interface to compose data analytic tasks easily, a novel cost-based optimizer which is able to find the most efficient platform, an executor that efficiently orchestrates tasks over different platforms.

Rheem is the first to tackle the challenges related to general purpose cross platform systems, it's goal is to run data analytics on multiple platforms efficiently. It provides a set of native API for developers to build applications, which includes in Java, Scala, Python, and REST. However, Rheem does not support any stream processing platform. It relies on fault tolerance therefore during transportation of data it is susceptible to failure. Rheem follows an optimization approach that splits an input task into subtasks and assigns each subtask to a specific platform, which in turn minimizes the runtime or monetary cost.

A cross-platform system has to be extensible and flexible in order to adapt to constant changes, how does Rheem cater for new updates or changes? How does Rheem ensure that data is not lost during transportation? Can Rheem be trusted to deliver data in a fast and efficient way? How does Rheem fight the Man-In-Middle attack?