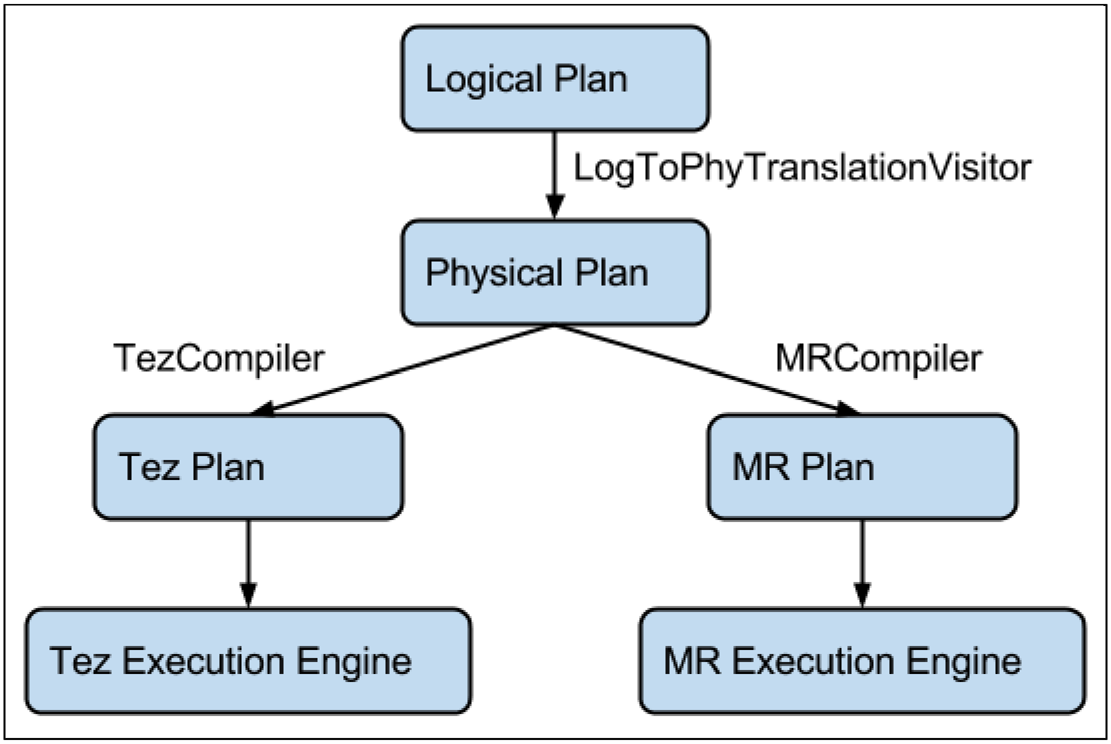
Pig on Tez

Table of Contents

Summary

The Pig/Tez integration will be based on Achal Soni's work. But one difference will be translating physical plan directly to Tez plan rather than translating MR plan to Tez plan. Complete decoupling between Tez and MR plans will provide not only cleaner implementation but also more flexibility for future improvements.



In addition to the front-end changes, Tez processors (PigProcessor) will be implemented in the back-end. This allows us to translate Pig queries to more optimal Tez DAGs.

Design

Frontend

blocked URL

Backend

PigProcessor.java

```
public class PigProcessor implements org.apache.tez.runtime.api.LogicalIOProcessor { ... }
```

Scope of phrase 1

Pig will take the same approach as what Hive is taking in the first phrase. The specific goals include:

- Make core Pig operators (including join, group-by, etc) work.
- Implement MRR optimization (Multiple reduce-stage jobs).
- Implement MPJ optimization (Multi-parent shuffle joins).

Functional requirements of phase 1

Functional requirements are almost identical to those of Hive on Tez, which can be viewed here.