Science is becoming a data management problem

Split if total expected runtime improves

1024

Goal

4096

When

Galaxy

Get good performance (14 hours vs. 70 minutes)

Seaflow

Assign tiny amount of work to each task to reduce skew

List of cost functions:

Express complex algorithms and

To find sweet spot,

Framework overhead

Extract

SkewReduce

Evaluate costs for

Histogram 1D

But it is hard to

Asteroid

The top red line runs for 1.5 hours

MapReduce

Toward scalable

Input

Merge cost:

Goal: Scalable analysis with minimal effort

Partition

Coordinates

Light intensities

Output

List of celestial objects

- Star

- Galaxy

- Planet

- Asteroid

- ...

Map

Features

- Partition multi-dimensional input data

- Extract features from each partition

- Merge (or reconcile) features “Hierarchical Reduce”

Finally output

How about having micro partitions?

- It works!

- Framework overhead overkills!

- To find sweet spot, need to try different granularities!

Can we find a good partitioning plan without trial and error?

Problem: Skew

Solution 1? Micro partition

Approach

Partition Plan Guided By Cost Functions

Search Partition Plan

- Greedy top-down search

- Split if total expected runtime improves

- Evaluate costs for subpartitions and merge

- Estimate new runtime

Prototype Architecture

Does SkewReduce work?

Impact of Cost Function

- SkewReduce plan yields 2 ~ 8 times faster running time

YongChul Kwon, Magdalena Balazinska, Bill Howe, Jerome Rolia (HP Labs)