Starfish: A Self-tuning System for Big Data Analytics
Herodotos Herodotou, Harold Lim, Fei Dong, Shivnath Babu
Department of Computer Science, Duke University

Workload Optimization

- Processing multiple workflows on same data
- Optimization Techniques
  - Data-flow sharing
  - Materialization
  - Reorganization

- Challenges
  - Interactions of above techniques among each other
  - and with scheduling, data layout policies, and configuration parameter settings

Jumbo Operator
- Use a single MapReduce job to process multiple Select-Project-Aggregate operations over a table
- Enables sharing of scans, computation, sorting, shuffling, and output generation

Cluster Sizing for Hadoop Analytics

- Workload performance and pay-as-you-go cost under various cluster configurations on Amazon Elastic MapReduce

Starfish Overview

Starfish in the Hadoop Ecosystem
- Starfish: A Self-tuning System for Big Data Analytics
  - Magnetism: attracts all sources of data
  - Agility: adapts in sync with rapid data evolution
  - Depth: supports complex analytics needs

- Starfish makes Hadoop MADDER and Self-Tuning
  - Data-lifecycle-awareness: achieves good performance throughout data lifecycle
  - Elasticity: adjusts resources and operational costs
  - Robustness: provides availability and predictability

Components in the Starfish Architecture

- Workload Optimizer
- Elastizer

- Profiler
- Workload Engine
- Job Optimizer
- Data Manager
- Intermediate Data Mgr.
- Data Layout & Storage Mgr.

Workflow Optimization

- Challenges
  - Dataflow interactions: Data produced by a MapReduce job is consumed by other MapReduce jobs
  - Resource interactions: MapReduce jobs running concurrently in the cluster are competing for a fixed amount of resources

- Optimization Space
  - Logical optimizations
    - Vertical packing
    - Horizontal packing
    - Partition function selection
    - Join operator selection
  - Physical optimizations
    - Job-level configuration settings
    - Dataset-level configuration settings

Response surfaces of MapReduce programs in Hadoop

- Word Co-occurrence in Hadoop
- TeraSort in Hadoop

Example analytics workload for Amazon Elastic MapReduce

- Users (username, age, ipaddr)
- GeoInfo (ipaddr, region)
- Clicks (username, url, value)

Hadoop is a MAD system for data analytics

- MapReduce: choice of technology in minutes
- Avoid the middleman (system administrator)
- Pay only for the resources used

Power to the users

Burden to the users
- Choose right cluster size & resources
- Configure the cluster correctly
- Choose job configuration settings for workflows
- Meet workflow requirements

Just-in-Time Job Optimization

- Goal
  - Find good settings for configuration parameters
  - Settings depend on job, data, and cluster characteristics

- Challenges
  - Heavy use of programming languages
  - Data loaded/accessed as opaque files
  - Large space of parameter settings

- Approach
  - Profiler: Uses dynamic instrumentation to learn performance models (job profiles) for unmodified MapReduce programs
  - What-if Engine: Uses a mix of analytical models, black-box models, and simulation to predict job performance
  - Job Optimizer: Searches through the high-dimensional space of parameter settings

Workflow Optimization

- Logical and Physical Optimizations in a MapReduce Workflow