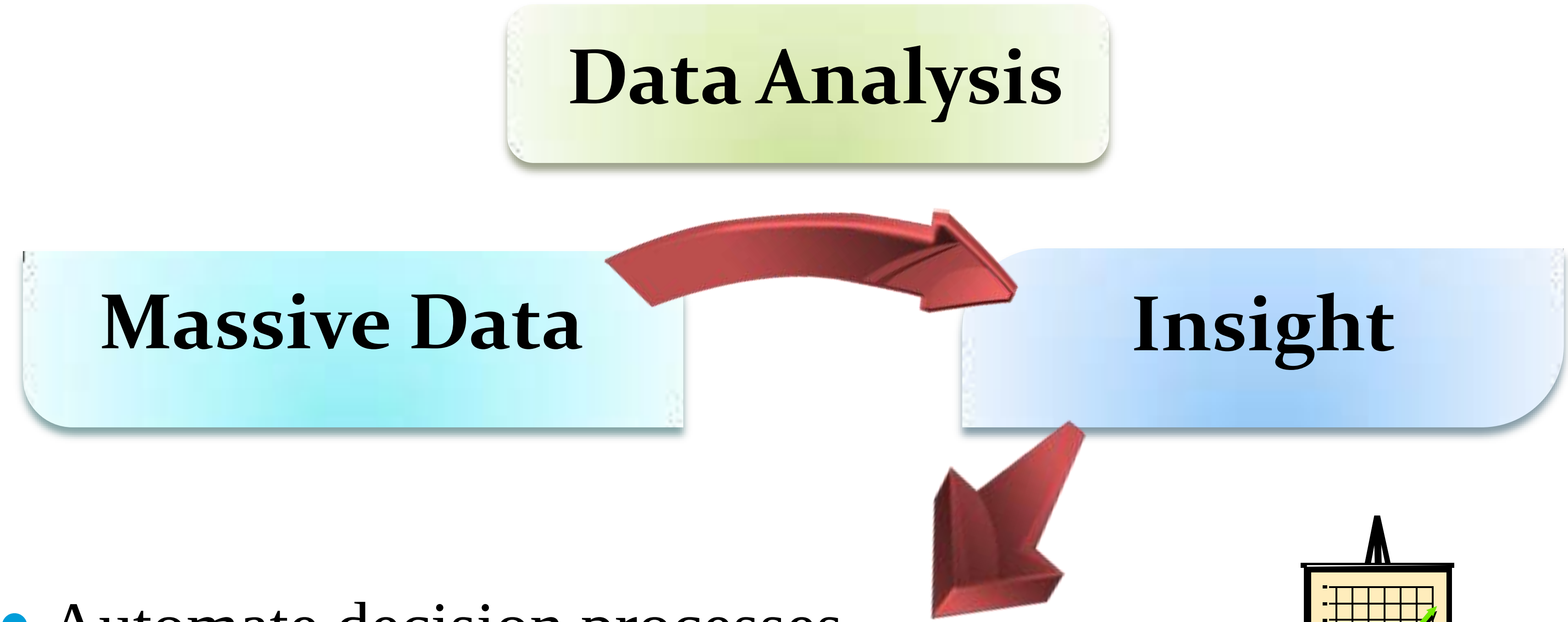
Starfish: A Self-tuning System for Big Data Analytics

Herodotos Herodotou, Harold Lim,
Gang Luo, Nedyalko Borisov, Liang Dong,
Fatma Bilgen Cetin, Shivnath Babu

Duke University

Analysis in the Big Data Era



Data Analysis

Massive Data

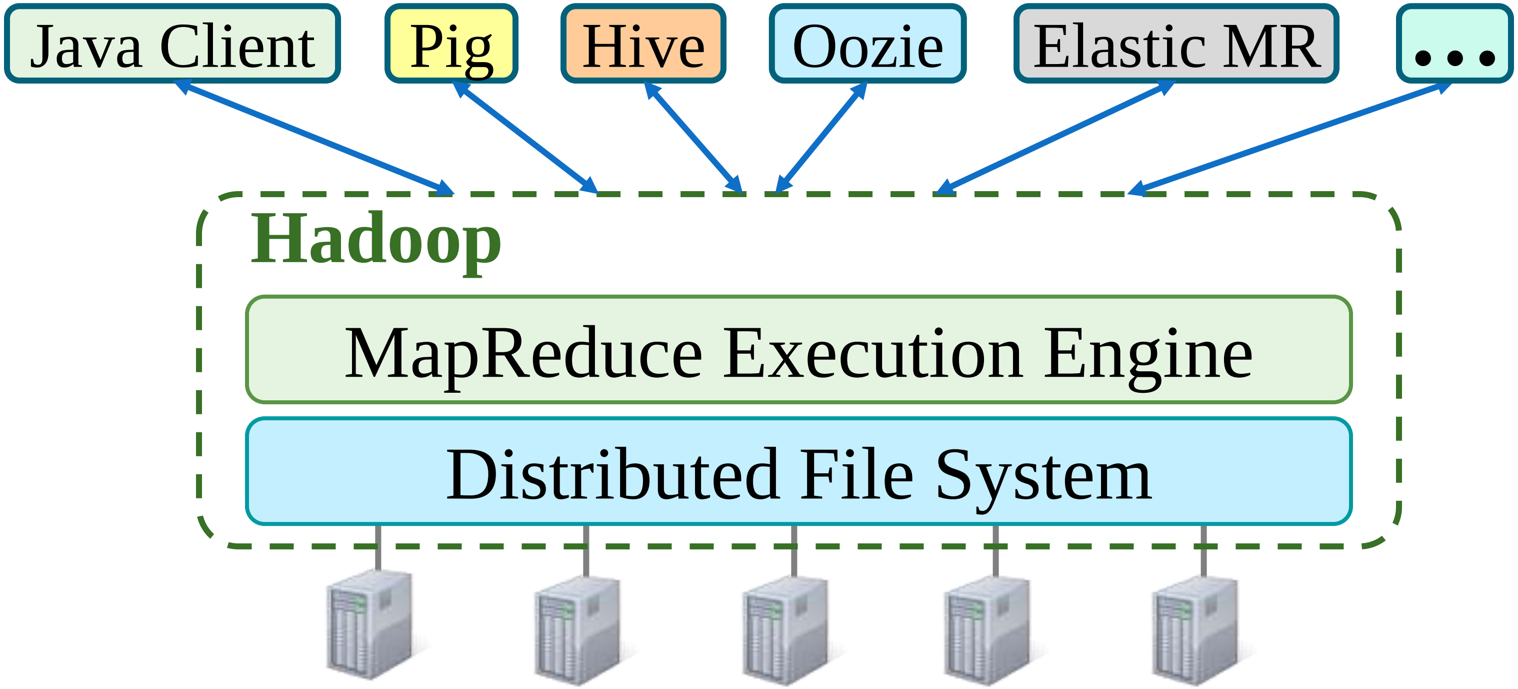
Insight

- Automate decision processes
- Increase cost savings and revenue

Key to Success = Timely and Cost-Effective Analysis

Analysis in the Big Data Era

- Popular option
  - Hadoop software stack

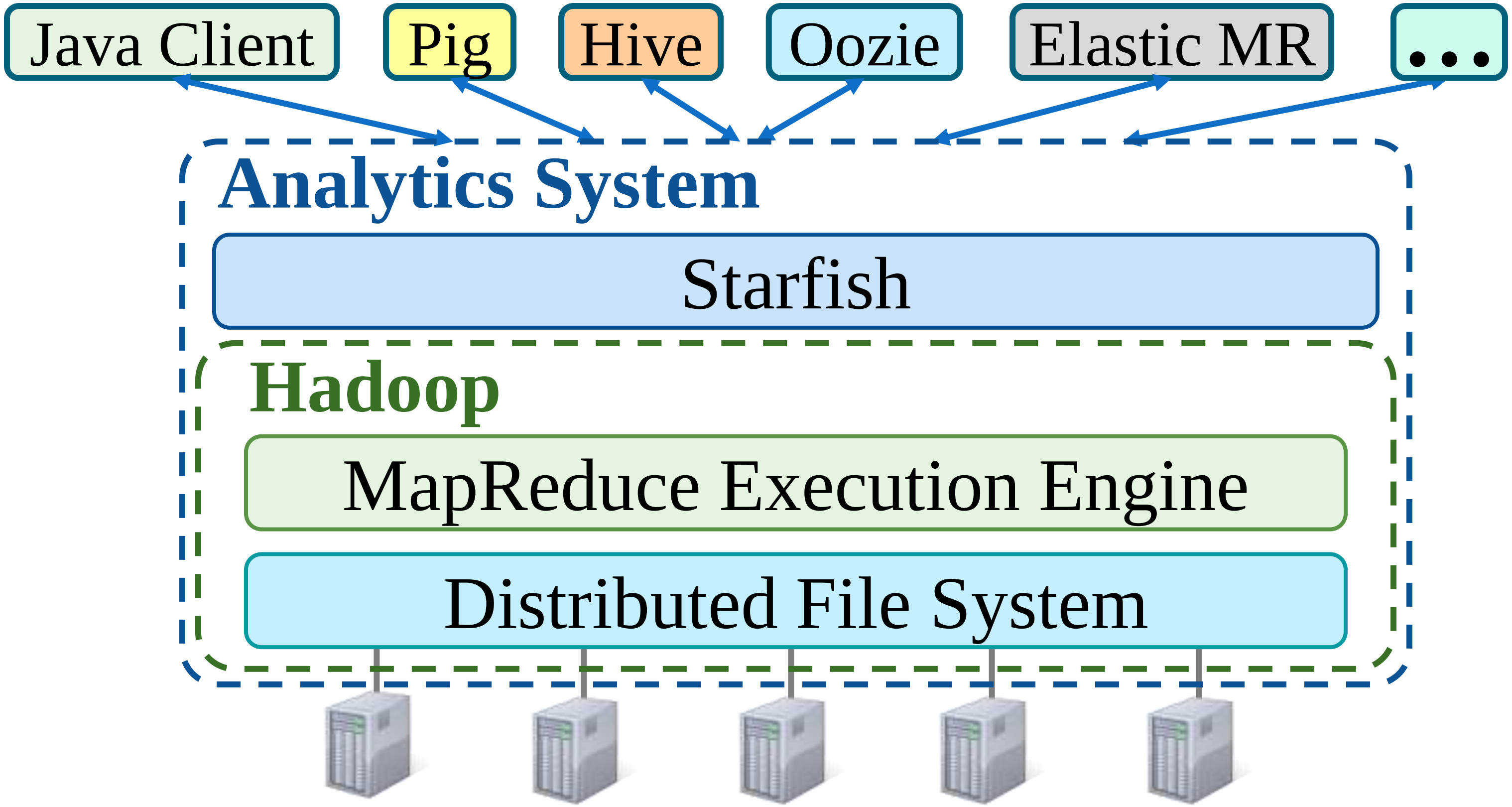


Analysis in the Big Data Era

- Popular option
  - Hadoop software stack
- Burden on the users
  - Responsible for provisioning & configuration
  - Usually lack expertise to tune the system
- Challenges
  - Tasks expressed in general-purpose programming languages
  - Input data stored as files and interpreted at run-time

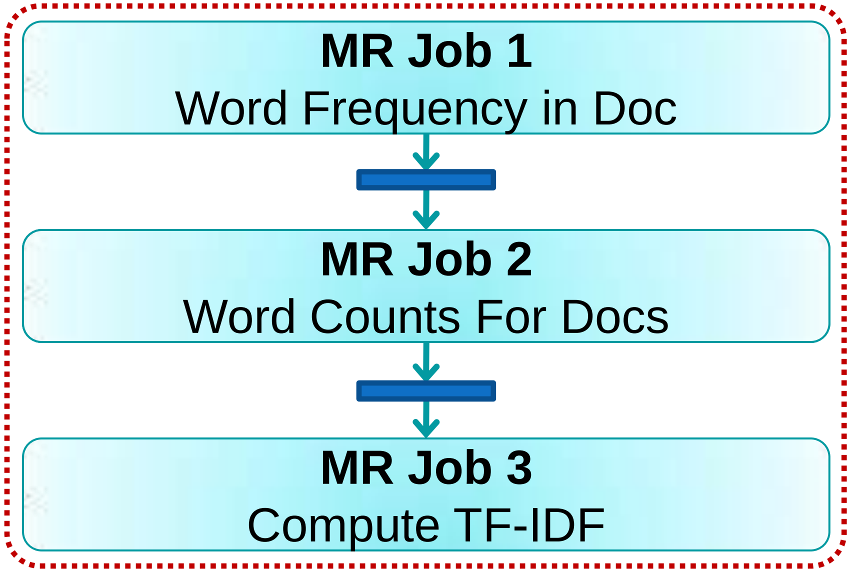
Starfish: Self-Tuning System

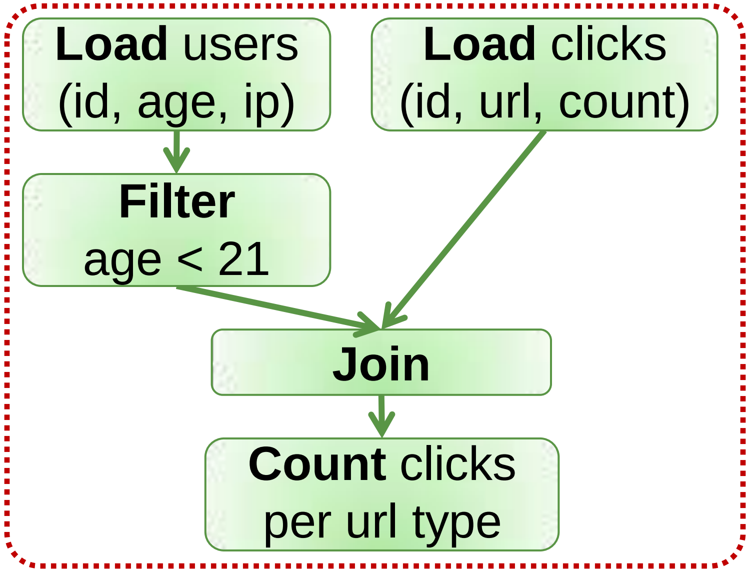
- NOT our goal: Improve Hadoop's peak performance
- Our goal: Provide good performance automatically



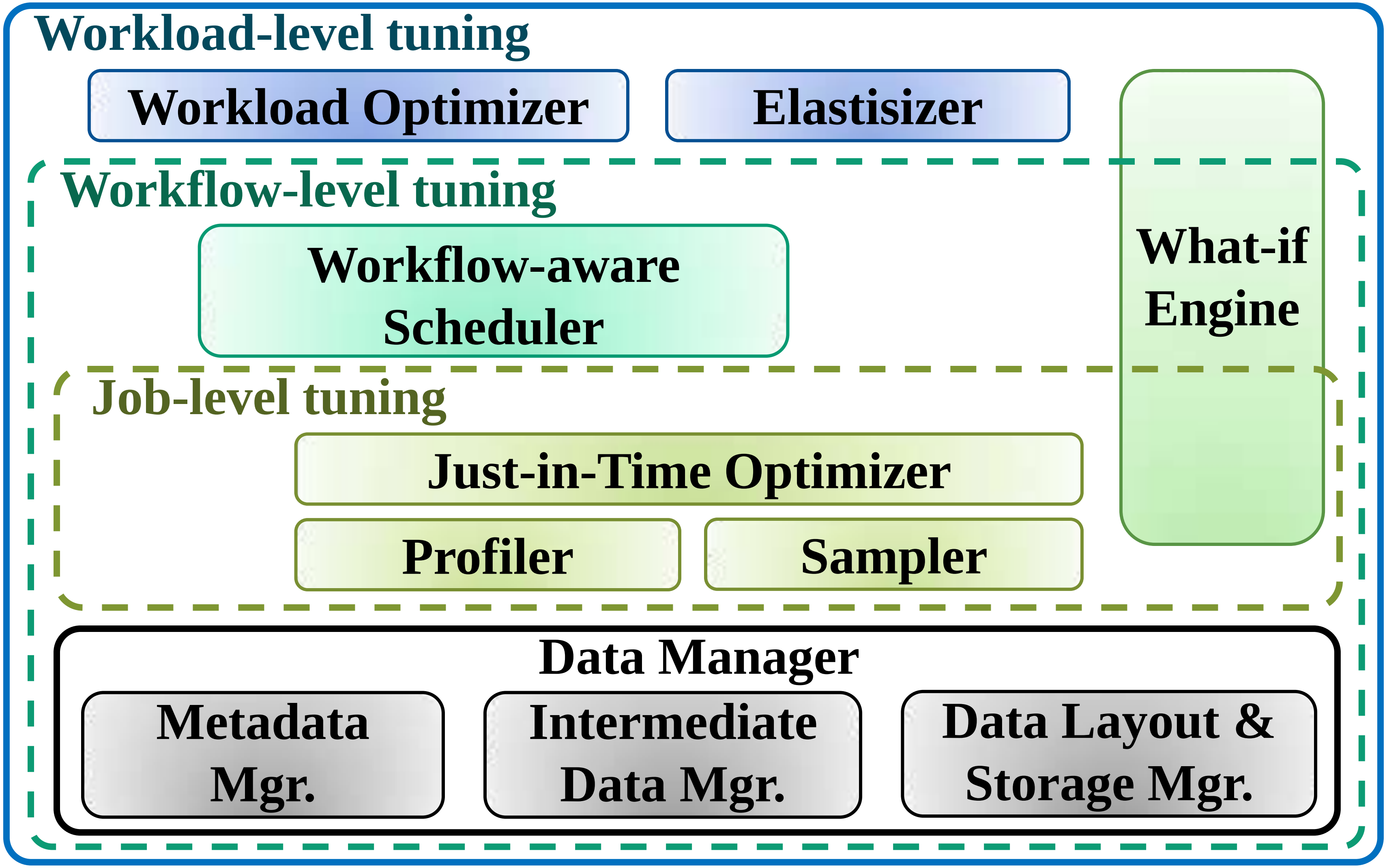
Workload on a Starfish Cluster

- MapReduce (MR) Job
- Workflow
  - Physical: directed graph of MR job nodes
  - Logical: directed graph of SPJA & UDF nodes
- Workload: Collection of workflows





Starfish Architecture



Workload-level tuning

Workload Optimizer

Elastisizer

What-if Engine

Workflow-level tuning

Workflow-aware Scheduler

Job-level tuning

Just-in-Time Optimizer

Profiler

Sampler

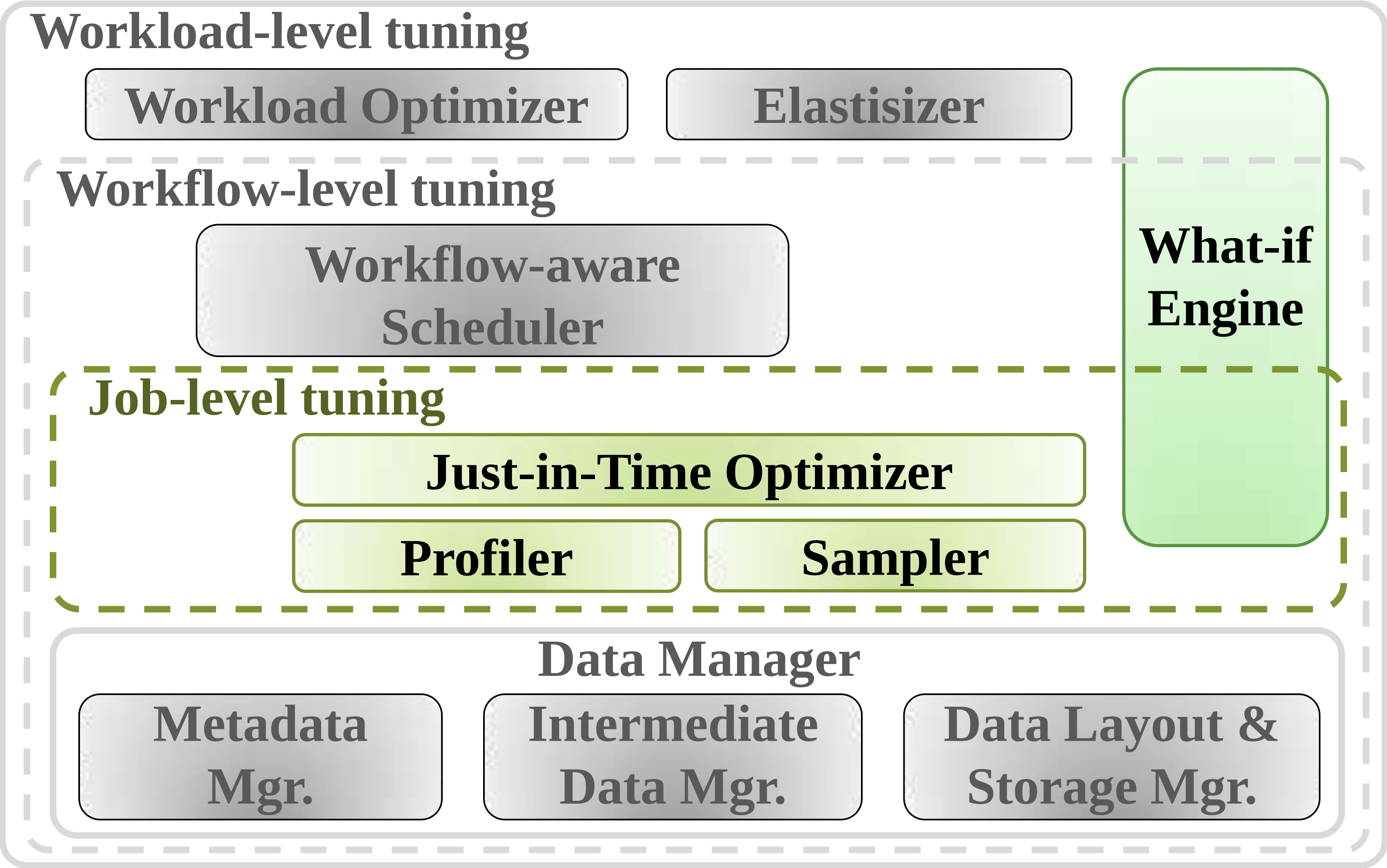
Data Manager

Metadata Mgr.

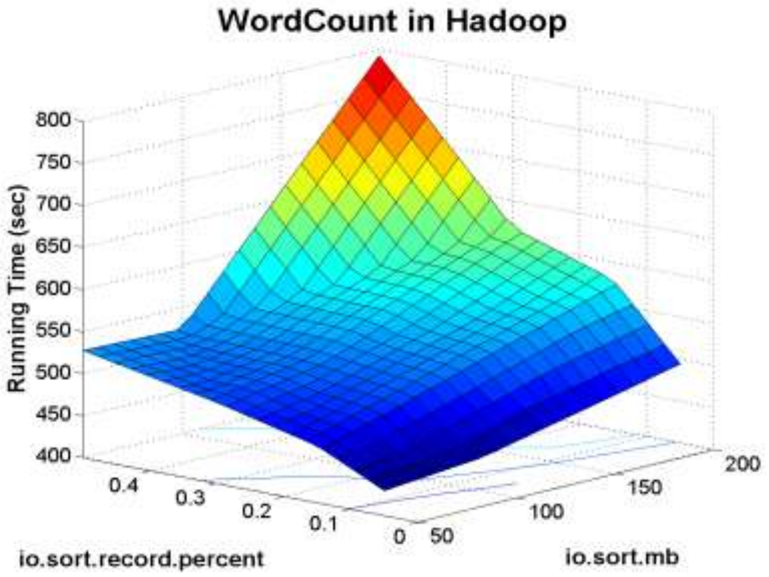
Intermediate Data Mgr.

Data Layout & Storage Mgr.

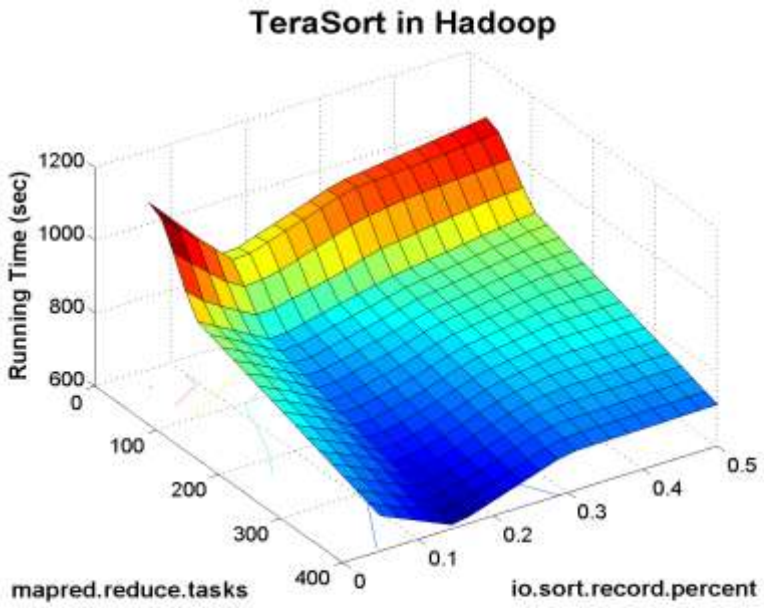
Starfish Architecture



Job Configuration Parameters



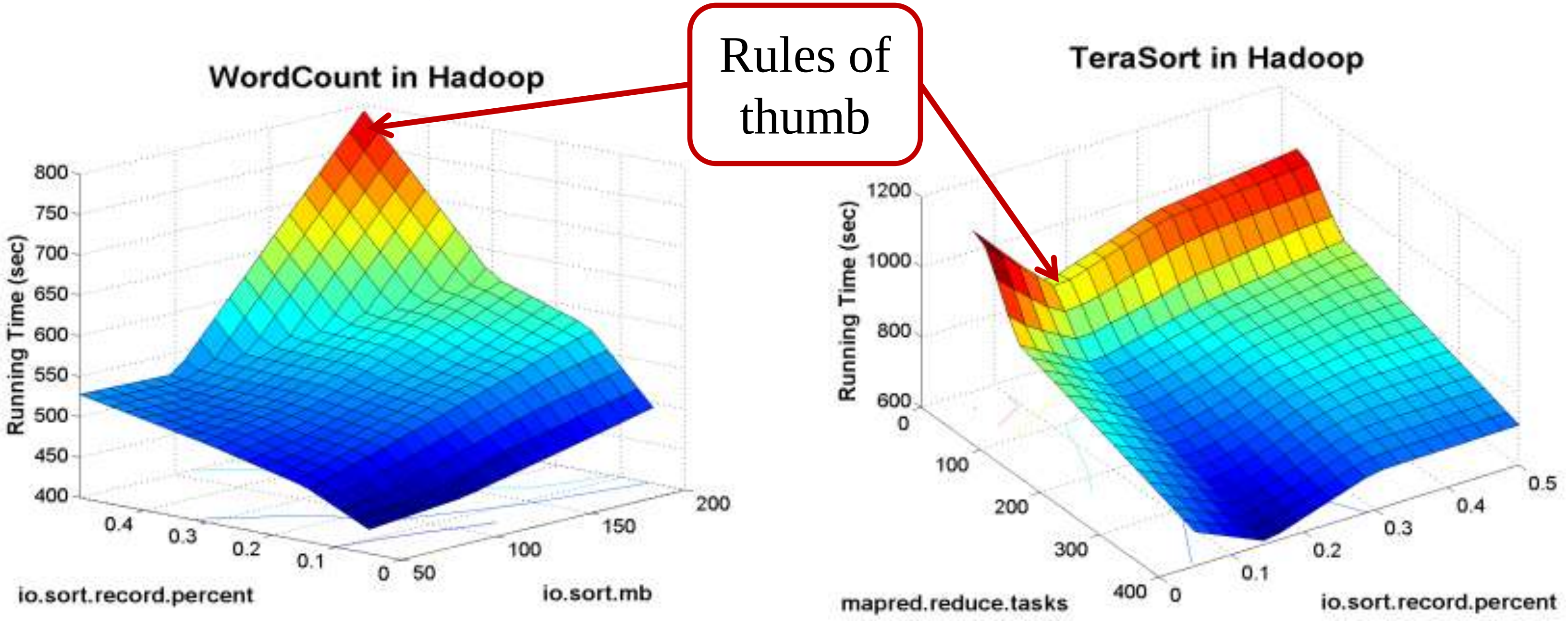
WordCount in Hadoop



TeraSort in Hadoop

- Over 190 parameters
- Many affect performance in complex ways
- Impact depends on Job, Data, and Cluster properties

Current Approach


Rules of thumb

- Rules of thumb
  - *mapred.reduce.tasks* = 0.9 * number_of_reduce_slots
  - *io.sort.record.percent* = 16 / (16 + average_record_size)
- Rules of thumb may not suffice

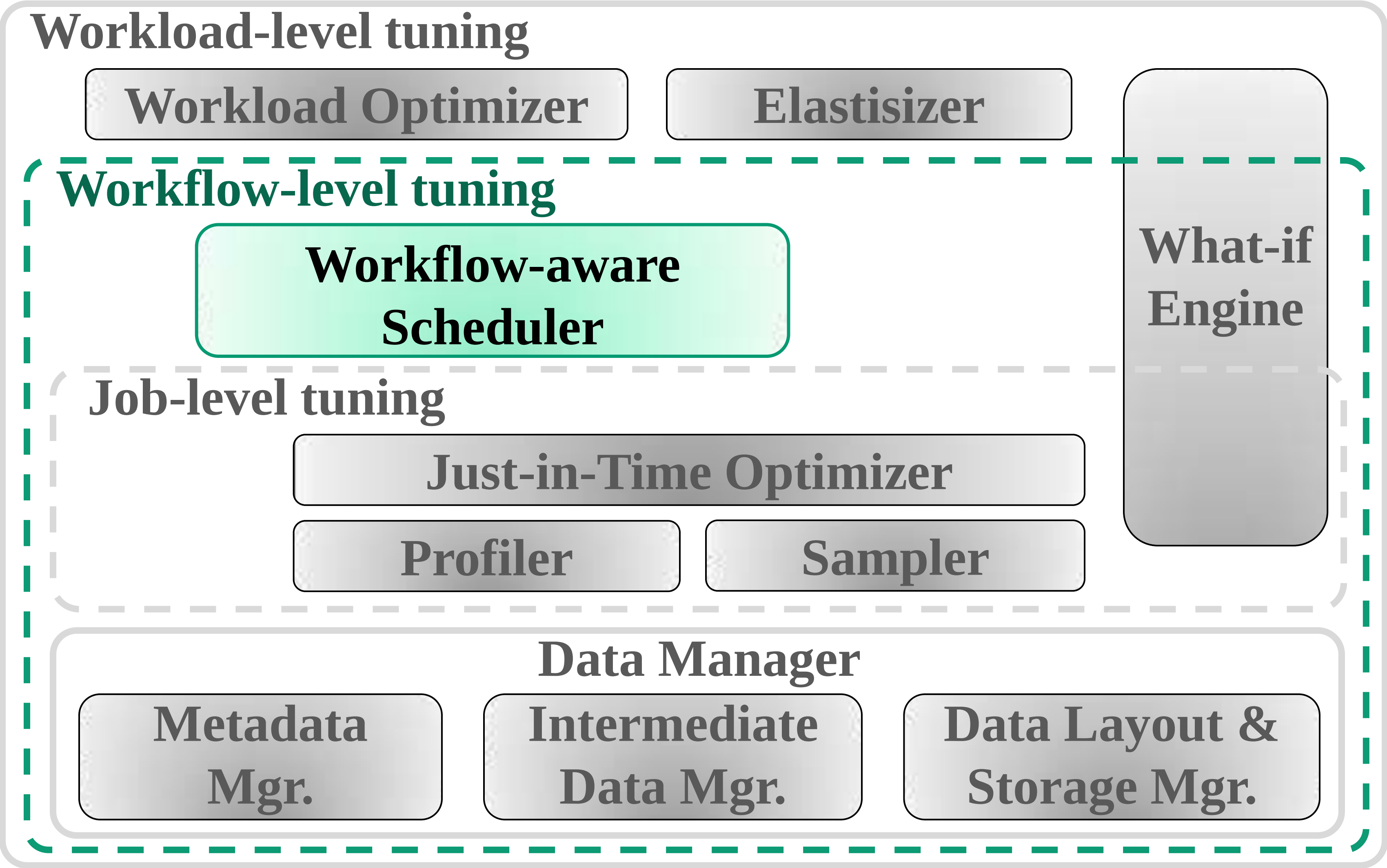
Just-in-Time Job Optimization

- Just-in-Time Optimizer
  - Searches through the high-dimensional space of parameter settings
- What-if Engine
  - Uses mix of simulation and model-based estimation
- Sampler
  - Collects statistics about input, intermediate, and output key-value spaces of MapReduce jobs
- Profiler
  - Collects information about MR job executions

Job Profiler

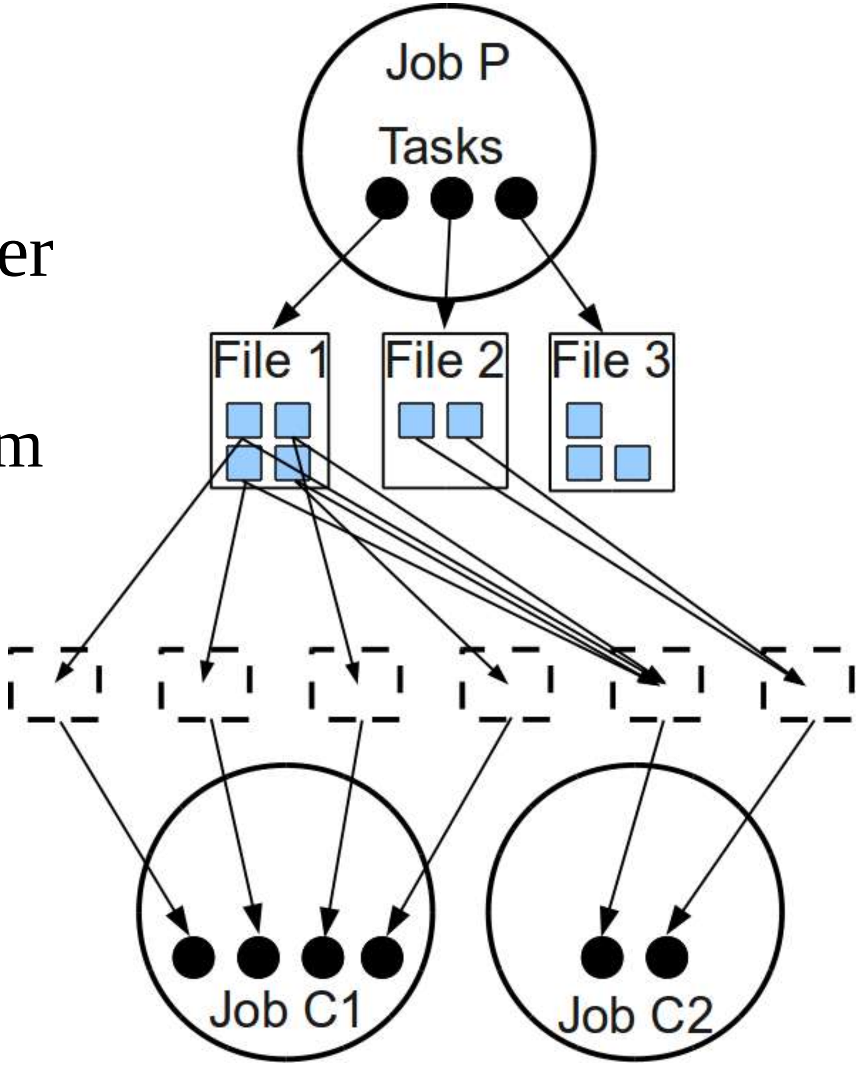
- Dynamic instrumentation
  - Monitor specific components in a system
  - Collect run-time information
- Benefits
  - Zero overhead when it is turned off
  - Works with unmodified MapReduce programs
- Used to construct a job profile
  - Concise representation of the job execution
  - Allows for in-depth analysis of the job behavior

Starfish Architecture



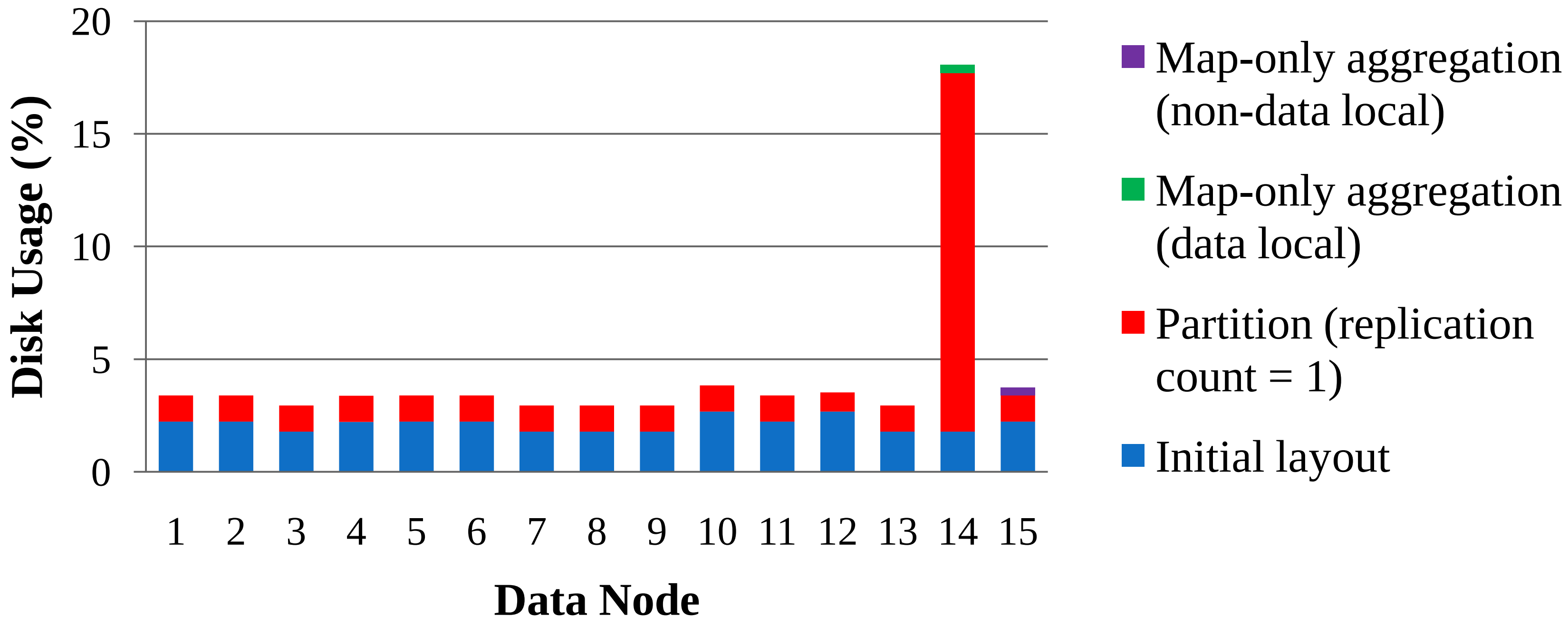
Job Workflows

- Producer-Consumer relationships among jobs
- Data Layout crucial for later jobs
  - Effective use of parallelism
  - Task scheduling
- Major Problem
  - Unbalanced data layouts



Unbalanced Data Layouts

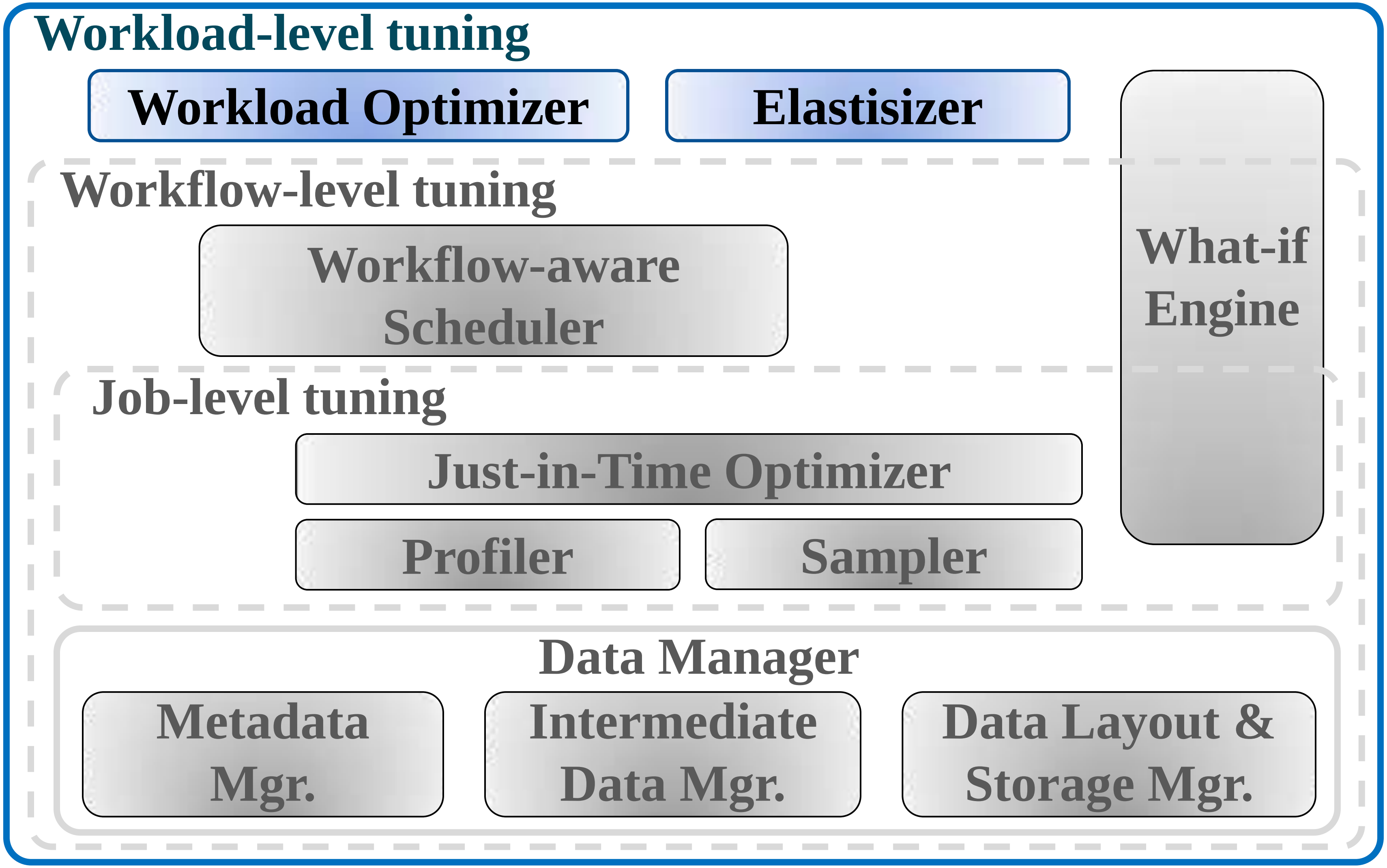
- Issues with data-locality-aware schedulers
  - Performance degradation due to reduced parallelism
  - Further unbalanced layout due to job outputs



Workflow-Aware Scheduler

- Goal: Optimize overall performance of workflow
  - Select best data layout + job parameters
- Space of options
  - Block placement policy
  - Replication factor
  - Block size
  - Output compression
- Approach
  - Simulate task scheduling and block placement policies
  - Perform cost-based search

Starfish Architecture



Workload-level tuning

Workload Optimizer

Elastisizer

What-if Engine

Workflow-level tuning

Workflow-aware Scheduler

Job-level tuning

Just-in-Time Optimizer

Profiler

Sampler

Data Manager

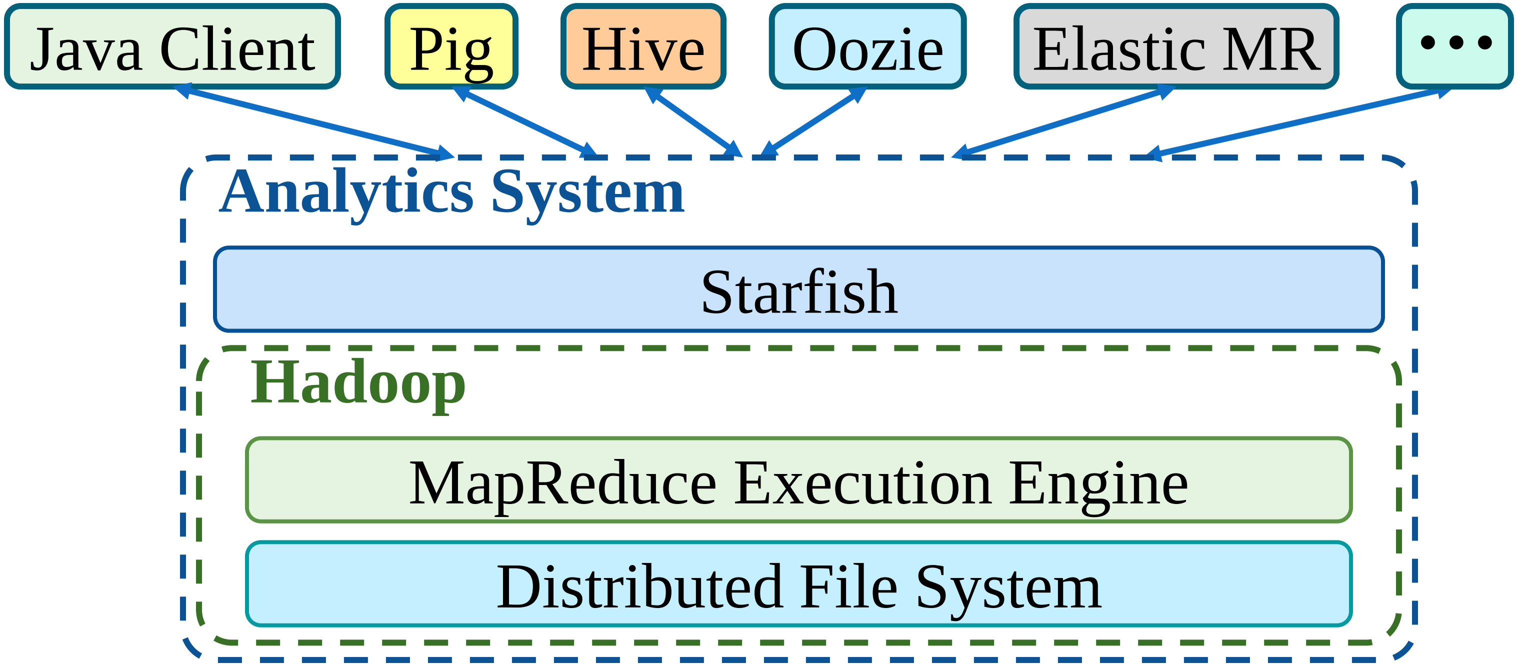
Metadata Mgr.

Intermediate Data Mgr.

Data Layout & Storage Mgr.

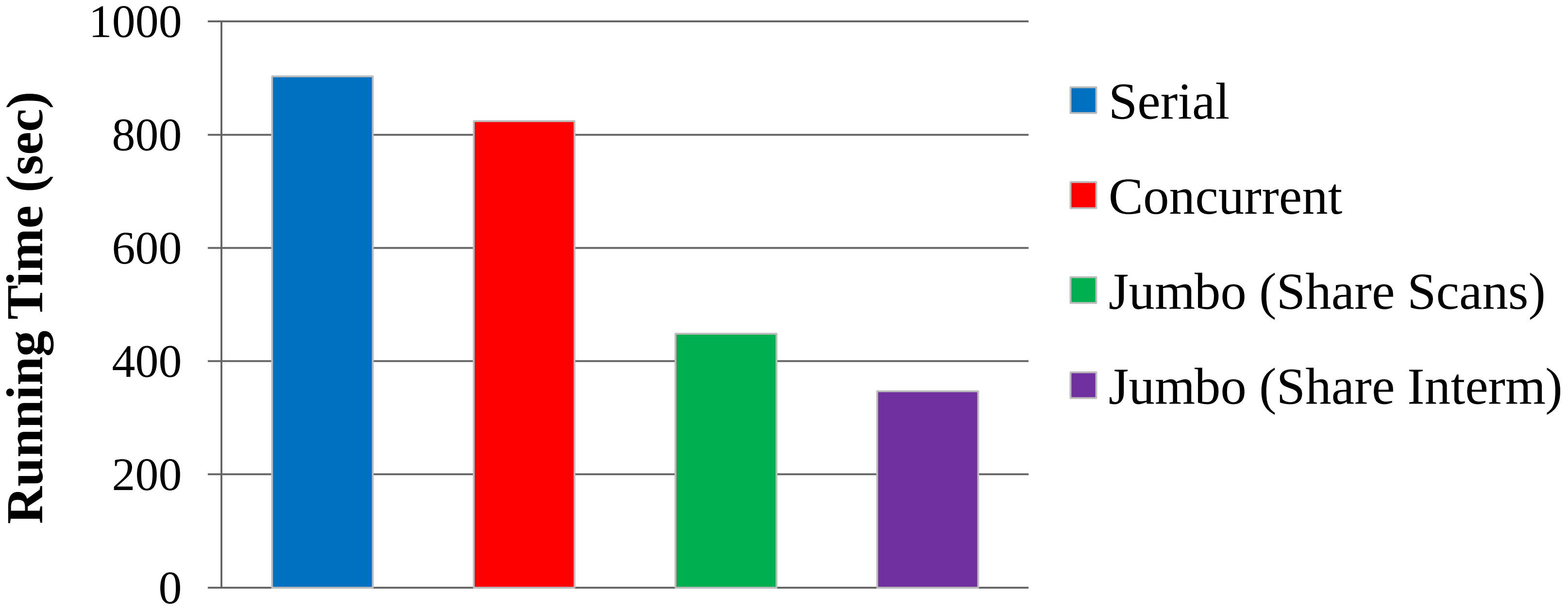
Optimizing Starfish Workloads

- Data-flow sharing
- Materialization
- Reorganization



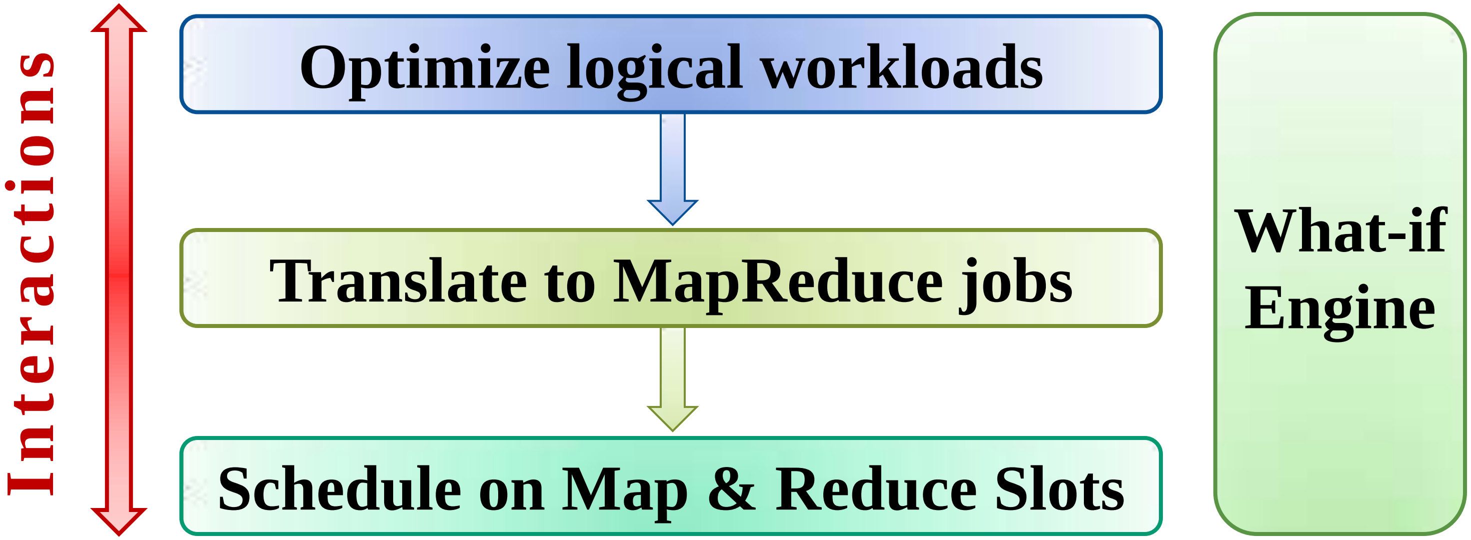
Jumbo Operator

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



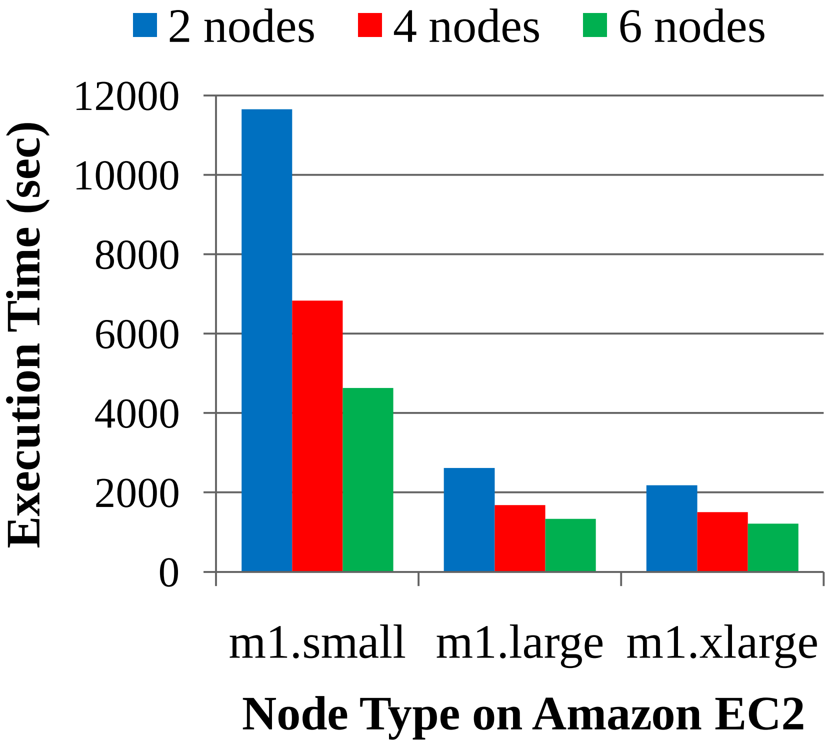
Challenges and Opportunities

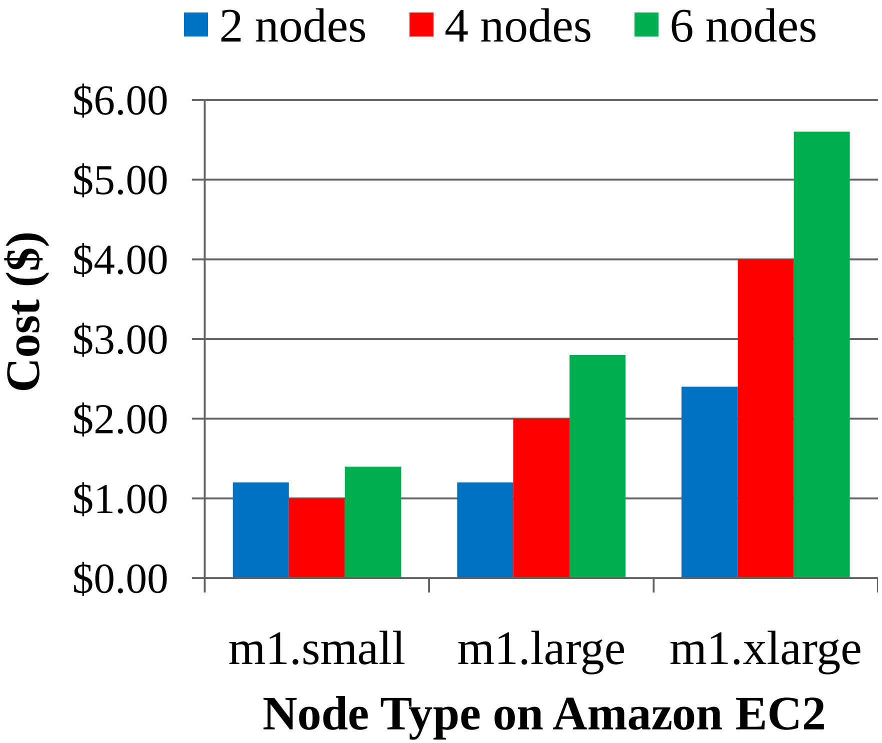
1. Study the interactions among optimizations across different levels
2. Construct a What-if Engine that can model these interactions


Interactions
Optimize logical workloads
Translate to MapReduce jobs
Schedule on Map & Reduce Slots
What-if Engine

Elastisizer – Hadoop Provisioning

- Goal: Make provisioning decisions based on workload requirements (e.g., completion time, cost)





Starfish: Self-Tuning System

Focus simultaneously on

- Different workload granularities
  - Workload
  - Workflows
  - Jobs (procedural and declarative)
- Across various decision points
  - Provisioning
  - Optimization
  - Scheduling
  - Data layout