Ciel: A Universal Execution Engine For Distributed Data-Flow Computing

Presented by: Tejas Kannan
Date: 17/10/2018
Outline

1. Introduction
2. Implementation and Contributions
3. Critique and Further Reading
4. Conclusion
Background Details

• Authors: Derek Murray, Malte Schwarzkopf, Christopher Smowton, Steven Smith, Anil Madhavapeddy and Steven Hand

• Product of the University of Cambridge Computer Laboratory

• Published in 2011 at the NSDI Conference
Limitations of Existing Platforms

MapReduce [1,4,9]  

Dryad [5,6]
Limitations of Existing Platforms

Issue: task graph is fixed, so iteration is difficult

- MapReduce
- Dryad
Adding Iteration to Hadoop with Mahout [2]

Problems:
1. Job overhead every iteration
2. No fault-tolerance between iterations
Ciel’s Dynamic Task Graph

Ciel enables a dynamic graph by allowing tasks to create follow-up tasks.

Diagram recreated from [8]
Preventing Cycles

A child task can depend only on:
1. Concrete references
2. Future references from already running tasks
Preventing Cycles

A child task can depend only on:
1. Concrete references
2. Future references from already running tasks
System Architecture

Object Table: Maintains references to objects stored on workers

Worker Table: Holds worker nodes and used to track their health

Task Table: Contains references to spawned tasks, as well as their dependencies

Diagram recreated from [8]
Scheduling Tasks

Scheduling is done using lazy evaluation:
1. Evaluate starting from the root task

2. For each subsequent task:
   a. If the task has concrete dependencies, evaluate it
   b. Otherwise, recursively evaluate tasks needed to resolve dependencies and unblock this task

Tasks are dispatched to workers who are nearest to the data
Performance Optimization: Memoization

• Tasks are deterministic

• Objects are given unique names using properties of the parent task

• Object name and reference stored in master’s object table

• If an object already exists, it is reused instead of recomputed

• Reduces runtime during computations which involve repetitive tasks
Performance Optimization: Streaming

Some tasks do not need the entire input object to start making progress.
Recovering From Failures

Worker failures are detected using periodic heartbeat messages
Recovering From Failures

Worker failures are detected using periodic heartbeat messages

1. Master invalidates object references at the failed worker
2. Master schedules the recomputation of any lost object according to the lazy policy
Recovering From Failures

Master failures are also detected using periodic heartbeat messages.

On recovery, a master node can rebuild its object table using the workers’ object stores.
Creating Ciel Jobs

Skywriting is an interpreted language created to run Ciel jobs

To boost performance, Skywriting tasks can make calls to procedures written in other languages.
Performance Evaluation

Ciel outperforms Hadoop when running both Grep and K-Means

Execution Time on Grep

Iteration Length on K-Means

Cluster Utilization on K-Means

Graphs taken from [8]
Performance Evaluation

Binomial Operations Pricing

Failed Master During Iteration

Speedup Using Streaming

Cluster Utilization

Graphs taken from [8]
Criticism

1. Ciel’s execution is never compared to a more optimized iterative platform such as HaLoop [3]
2. Number of trials during testing never specified
3. Streaming optimization demonstrated but never compared to another system
4. Ciel does not use multiple cores on worker nodes while scheduling
Selection of Related Work

1. Hive enables SQL-like queries to be executed on large datasets using Hadoop [10]

2. Spark allows for iterative tasks and derives its efficiency from in-memory computation [11, 12]

3. Naiad uses cycles in its execution graph to enable low latency processing of streams, as well as iterative and incremental tasks [7]
Conclusion

1. Distributed data processing engine meant for general purpose tasks
2. Dynamic task allocation enables efficient iterative computations
3. Fault-tolerant design with automatic recovery
4. Scripting language Skywriting used to construct Ciel jobs
5. Empirically outperforms Hadoop on iterative tasks
References


Questions?