Execution Time Prediction of Jobs Using SWAT

Motivation

- Data and compute intensive workload in cloud infrastructure.
- Different hardware configurations.
- Many distributed systems frameworks are using GPUs.
- Best configurations for lengthy jobs.
- Jobs execution time using machine learning models.

Outline

1. **Introduction**
2. Background
   a. GPUs
   b. Spark.
   c. Swat.
3. System Architecture
   a. Feature Extraction.
   b. Machine learning models.
4. Evaluation
5. Conclusion
6. Future Work

Introduction

- Frameworks like Tensorflow, SWAT, HadoopCL and SparkCL uses GPU.
- These Frameworks runs Lengthy jobs.
- Hardware configuration has large impact on performance.
- Predictive models need to evaluate jobs against different hardware configurations.
- Previous approaches concentrated on treating the job as a black box or using statistical models.

Outline

1. Introduction
2. **Background**
    a. GPUs
    b. Spark.
    c. Swat.
3. System Architecture
    a. Feature Extraction.
    b. Machine learning models.
4. Evaluation
5. Conclusion
6. Future Work

GPUs

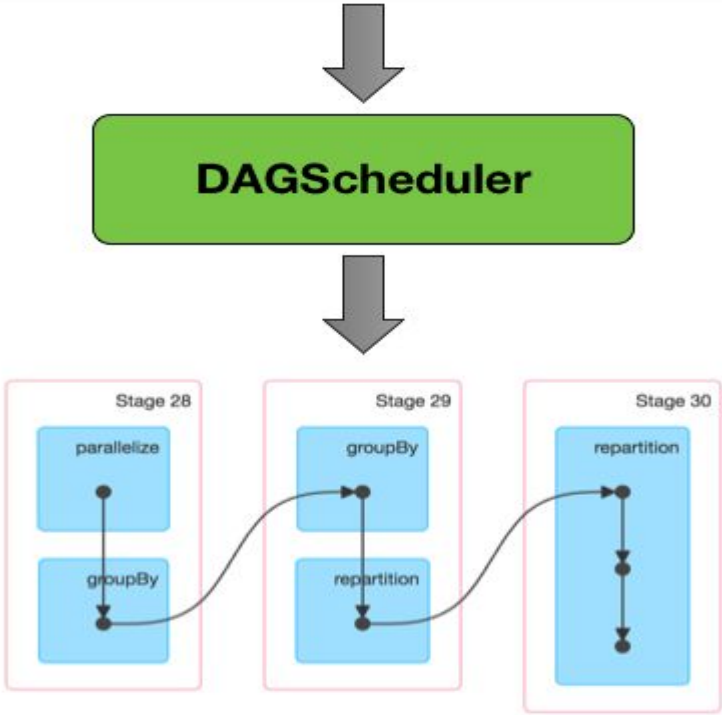
- High computation power.
- Parallel processing.
- Suitable for processing common workload executed on spark.
- Different model such as CUDA and OpenCL.
- Low level-programming and difficult concepts.
- SWAT solves that by running spark job without modifications.

Spark

- An RDD represents a distributed vector of elements. Elements in an RDD can be of any serializable type.
- A single RDD is split into multiple partitions. All elements in the same partition are stored on the same machine.
- DAGScheduler is the scheduling layer of Apache Spark that implements stage-oriented scheduling.
- It transforms a logical execution plan (i.e. RDD lineage of dependencies built using RDD transformations) to a physical execution plan (using stages).

DAG Scheduler

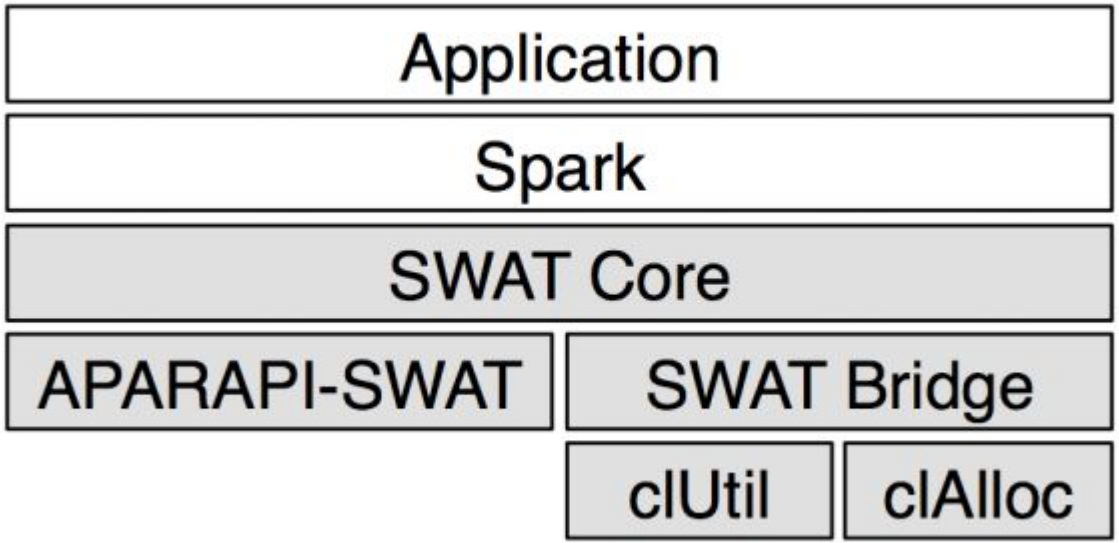
```
(2) MapPartitionsRDD[62] at repartition at <console>:27
 |  CoalescedRDD[61] at repartition at <console>:27
 |  ShuffledRDD[60] at repartition at <console>:27
 +-(8) MapPartitionsRDD[59] at repartition at <console>:27
    |  ShuffledRDD[58] at groupBy at <console>:27
    +-(8) MapPartitionsRDD[57] at groupBy at <console>:27
       |  ParallelCollectionRDD[0] at parallelize at <console>:24
```



SWAT

- SWAT is an accelerated data analytics (ADA) framework.
- Enables programmers to natively execute Spark applications on high performance hardware.
- Write their applications in a JVMbased language like Java or Scala.
- Runtime code generation creates OpenCL kernels from JVM bytecode.
- Open Source found in h□ttps:// github.com/agrippa/spark-swat.

SWAT Software Stack



Outline

1. Introduction
2. Background
   a. GPUs
   b. Spark.
   c. Swat.
3. **System Architecture**
   a. Feature Extraction.
   b. Machine learning models.
4. Evaluation
5. Conclusion
6. Future Work

System Architecture

- Two main phases training phase and test phase.
- we train our machine learning model on features extracted from jobs or stages.
- we extract these features and predict the execution time of the job or stage itself.
- The full predictive model vs. The extrapolator model.

Job and Stages Features

- Data and job (or stage) features: such as Number of RDD, GC Time, Ge tting Result Time, etc.
- Hardware configurations: Number of machine, Memory Size, Number of GPUs.
- These kinds of features spans all what we need to predict the job execution time if they were extracted right.

Machine learning Models

- SVR (SVM Regressor): a regression version of famous SVM algorithm.
- Gradient Boosting Regression: ensemble prediction model using multiple decision trees.
- For each algorithm, there are two models:
  - Stage Execution time predictor.
  - Job Execution time predictor.

Outline

1. Introduction
2. Background
   a. GPUs
   b. Spark.
   c. Swat.
3. System Architecture
   a. Feature Extraction.
   b. Machine learning models.
4. **Evaluation**
5. Conclusion
6. Future Work

Environment & Datasets used

- Processor: Intel Core i7-3612QM CPU @ 2.10GHz x8.
- Memory Size: 8GB
- ATI Radeon HD 5450 2GB memory.
- Datasets:
  - TPCH dataset (with no gpu).
  - SWAT tests with (41 machine learning algorithms mainly).
  -

Metrics Evaluation

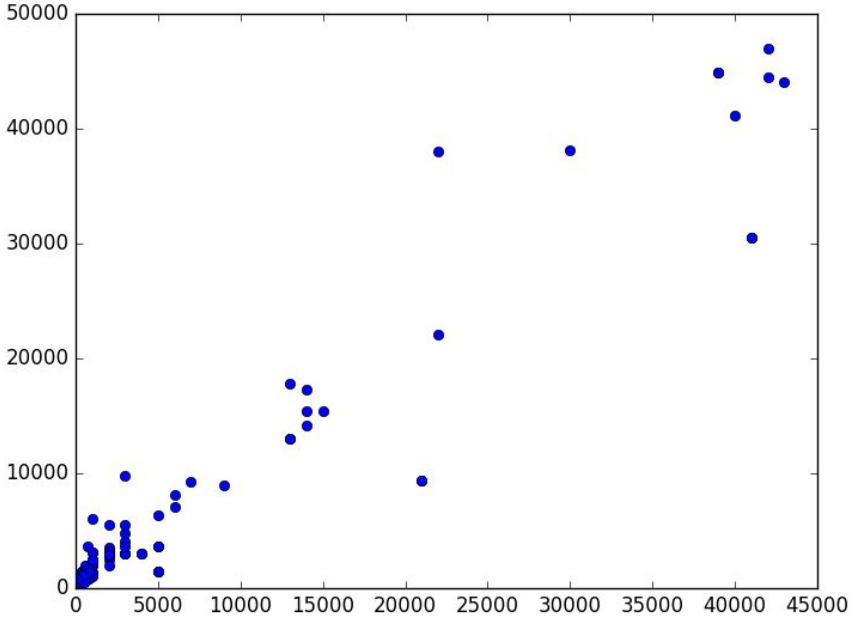
Metric	Gradient Boosting	SVR
R-Squared	0.76	0.33
MSE	3346.78689743548	11799.042329045044

Table 2: Metric for prediction on stages

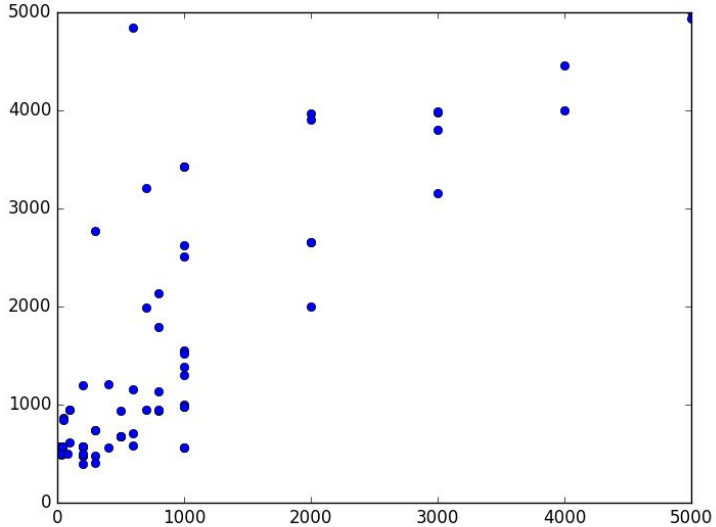
Metric	Gradient Boosting	SVR
R-Squared	-4.14147924142	-0.581533211492
MSE	44068.955	25809.03

Table 3: Metric for prediction on jobs

Scatter Plots for stages



Gradient Boosting



SVR

Scatter Plots for Jobs

Figure 5: Scatter Plot of Gradient Boosting for Jobs

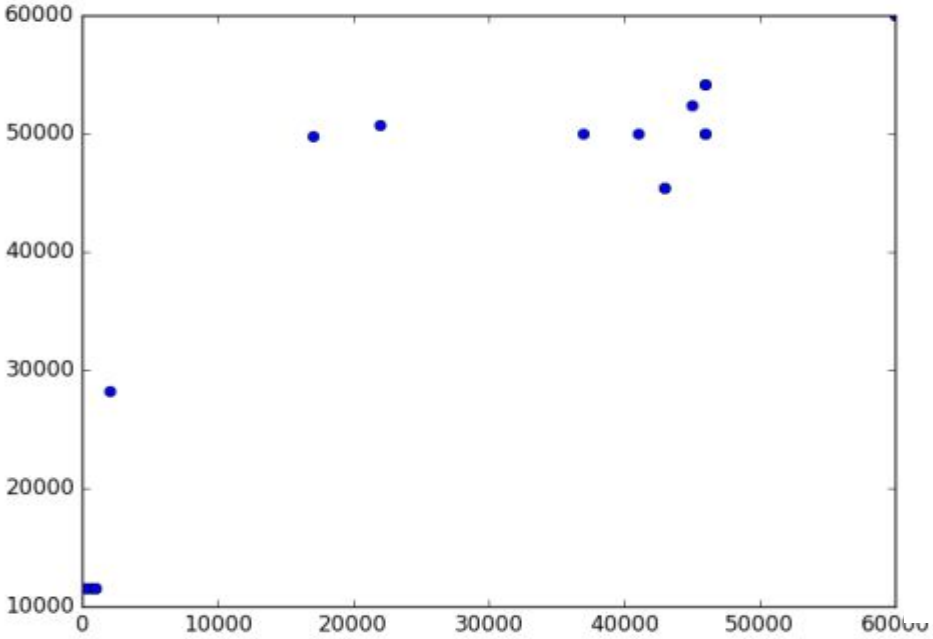
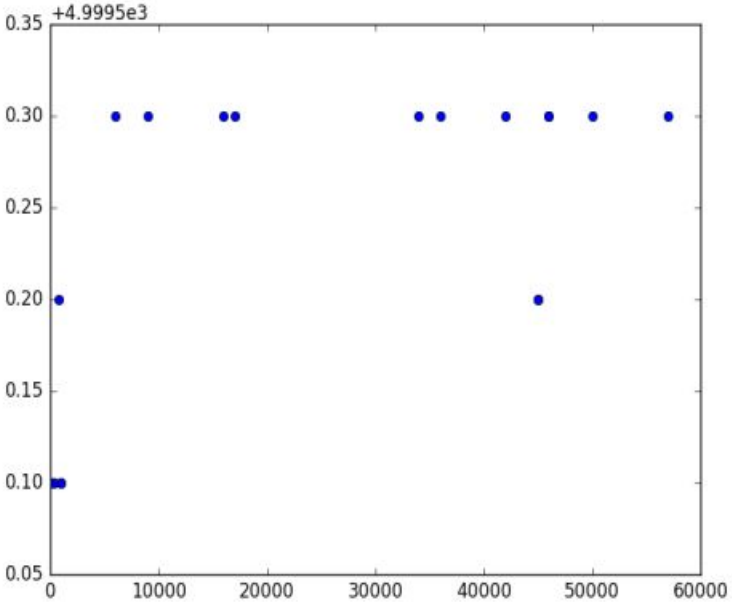


Figure 6: Scatter Plot of SVR for Jobs



Conclusion & Future Work

- Current Gradient Boosting model for stages is certainly the best.
- Need more data for jobs and trying different features.
- Need huge datasets for thorough testing and apply complex machine learning algorithms.
- Implementing the extrapolator when the prediction with full model is better.