March 29th,, 2023
Data Council, Austin TX

Sync

Workload optimized Apache Spark with Sync

The easiest way to optimize Apache Spark

Raffle winners announced at the end!

Jeffrey Chou, PhD
CEO
jeff.chou@synccomputing.com
MIT Post-Doc

About the team - Join us!

Leadership

Jeff Chou, PhD

CEO, Co-founder

Suraj Bramhavar, PhD

CTO, Co-founder

Casey Doran

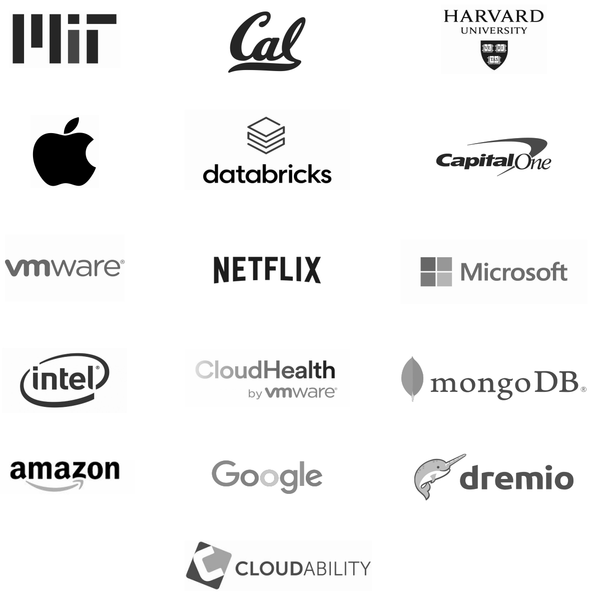
VP of Product

Malino Oda

VP of Engineering

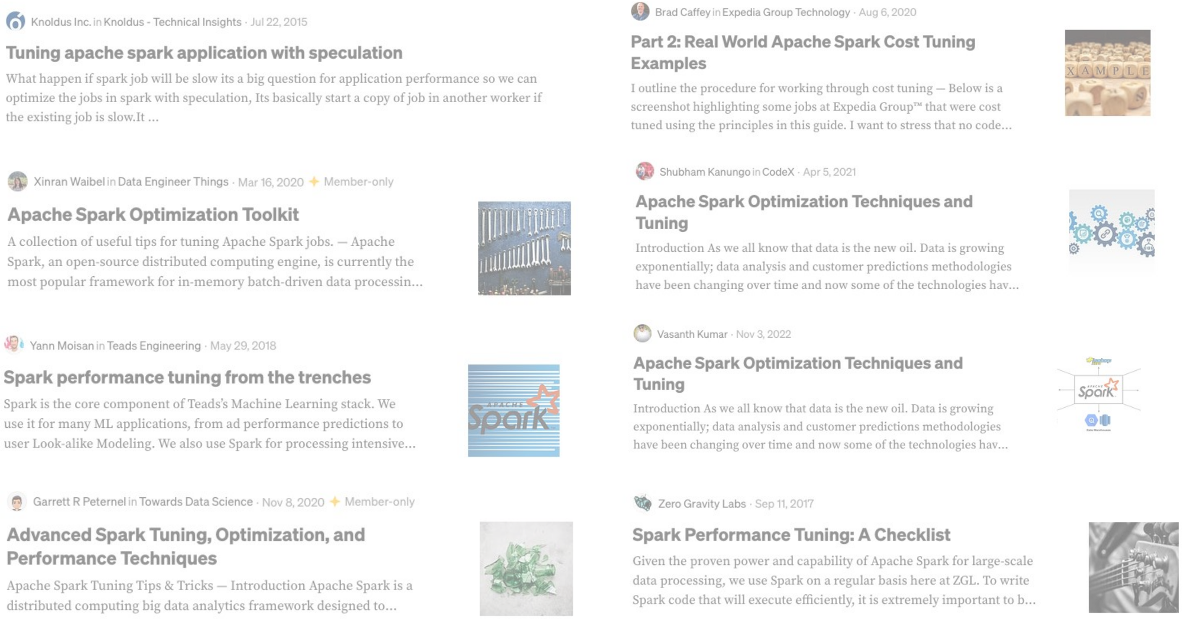
https://www.synccomputing.com/careers

Staff & Advisors From



The annoying problem of tuning Spark

100’s of posts on Medium… and counting



Knoldus Inc. in Knoldus - Technical Insights · Jul 22, 2015

Tuning apache spark application with speculation

What happen if spark job will be slow its a big question for application performance so we can optimize the jobs in spark with speculation, Its basically start a copy of job in another worker if the existing job is slow.It …

Xinran Waibel in Data Engineer Things · Mar 16, 2020 ✦ Member-only

Apache Spark Optimization Toolkit

A collection of useful tips for tuning Apache Spark jobs. — Apache Spark, an open-source distributed computing engine, is currently the most popular framework for in-memory batch-driven data processin…

Yann Moisan in Teads Engineering · May 29, 2018

Spark performance tuning from the trenches

Spark is the core component of Teads’s Machine Learning stack. We use it for many ML applications, from ad performance predictions to user Look-alike Modeling. We also use Spark for processing intensive…

Garrett R Peternel in Towards Data Science · Nov 8, 2020 ✦ Member-only

Advanced Spark Tuning, Optimization, and Performance Techniques

Apache Spark Tuning Tips & Tricks — Introduction Apache Spark is a distributed computing big data analytics framework designed to…

Brad Caffey in Expedia Group Technology · Aug 6, 2020

Part 2: Real World Apache Spark Cost Tuning Examples

I outline the procedure for working through cost tuning — Below is a screenshot highlighting some jobs at Expedia Group™ that were cost tuned using the principles in this guide. I want to stress that no code…

Shubham Kanungo in CodeX · Apr 5, 2021

Apache Spark Optimization Techniques and Tuning

Introduction As we all know that data is the new oil. Data is growing exponentially; data analysis and customer predictions methodologies have been changing over time and now some of the technologies hav…

Vasanth Kumar · Nov 3, 2022

Apache Spark Optimization Techniques and Tuning

Introduction As we all know that data is the new oil. Data is growing exponentially; data analysis and customer predictions methodologies have been changing over time and now some of the technologies hav…

Zero Gravity Labs · Sep 11, 2017

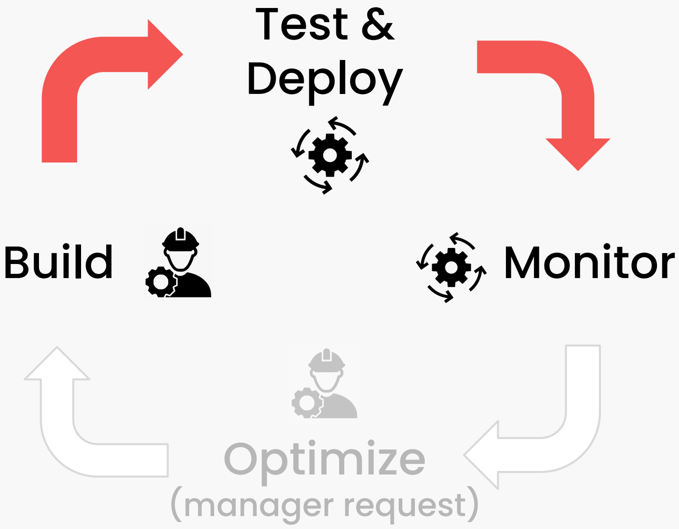
Spark Performance Tuning: A Checklist

Given the proven power and capability of Apache Spark for large-scale data processing, we use Spark on a regular basis here at ZGL. To write Spark code that will execute efficiently, it is extremely important to b…

- Spark configurations
- AWS configurations
- Spot variability
- Code optimization
- Data skew
- Memory errors

Optimization Problem for Developers

Before Sync:


Test & Deploy

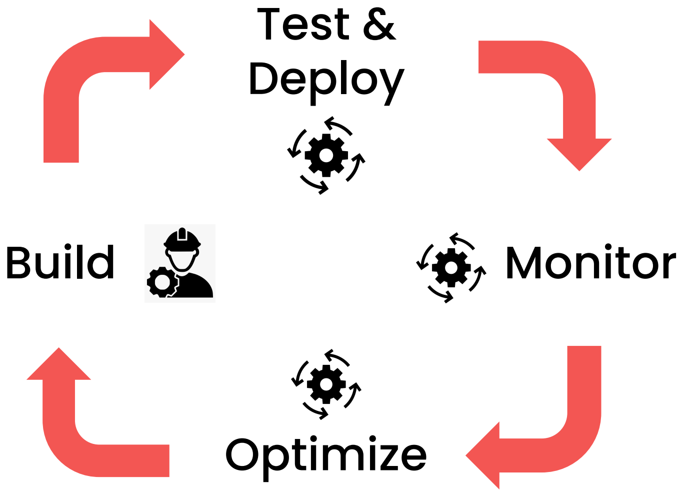
Build

Monitor

Optimize

(manager request)

After Sync:


Test & Deploy

Build

Monitor

Optimize

The 3 Sync Value Propositions

01

Significantly Reduce Total Cost of Data Workloads

Reduce waste from overprovisioning and align infrastructure with business value.

02

Dramatically Improve Data Engineering Productivity

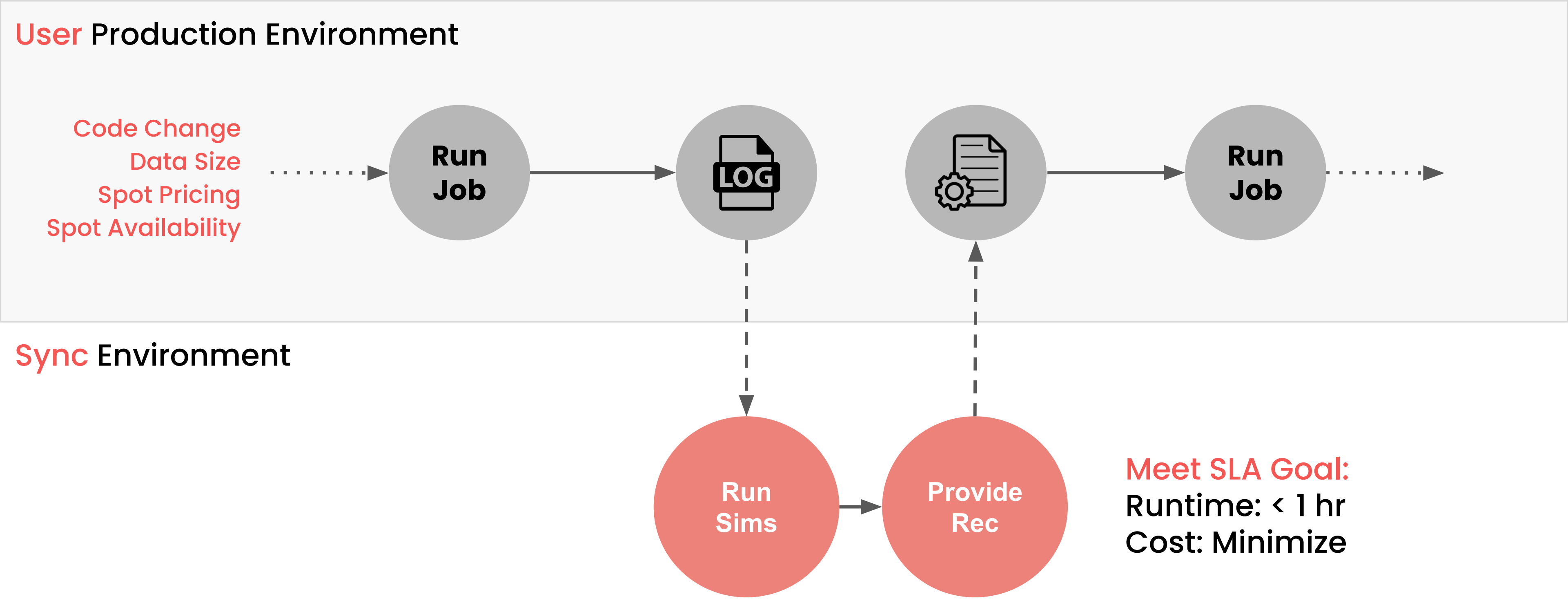
Increase the velocity of your Data Engineering teams and align priorities with customer use cases.

03

Reduce Risk for Mission Critical Data Workloads

While delivering value for your end users, consistently meet performance SLAs, reduce customer churn, and improve brand perception.

Data workflow



User Production Environment

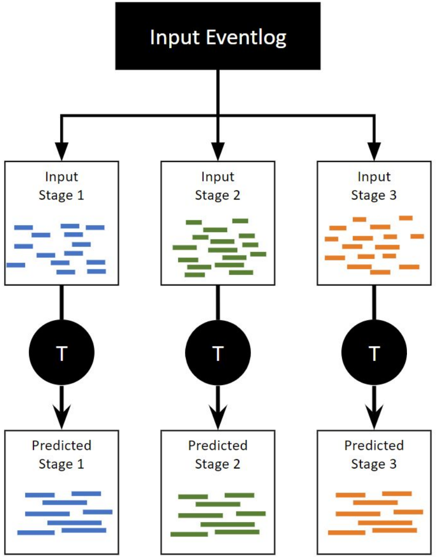
Code Change
Data Size
Spot Pricing
Spot Availability

Sync Environment

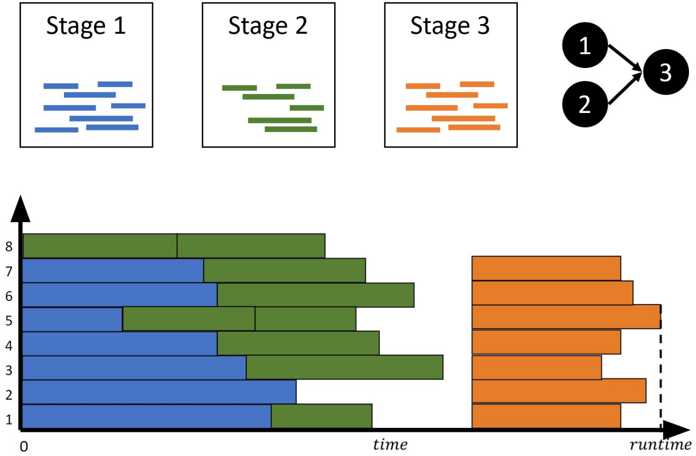
Meet SLA Goal:
Runtime: < 1 hr
Cost: Minimize

Under the hood of predicting Spark

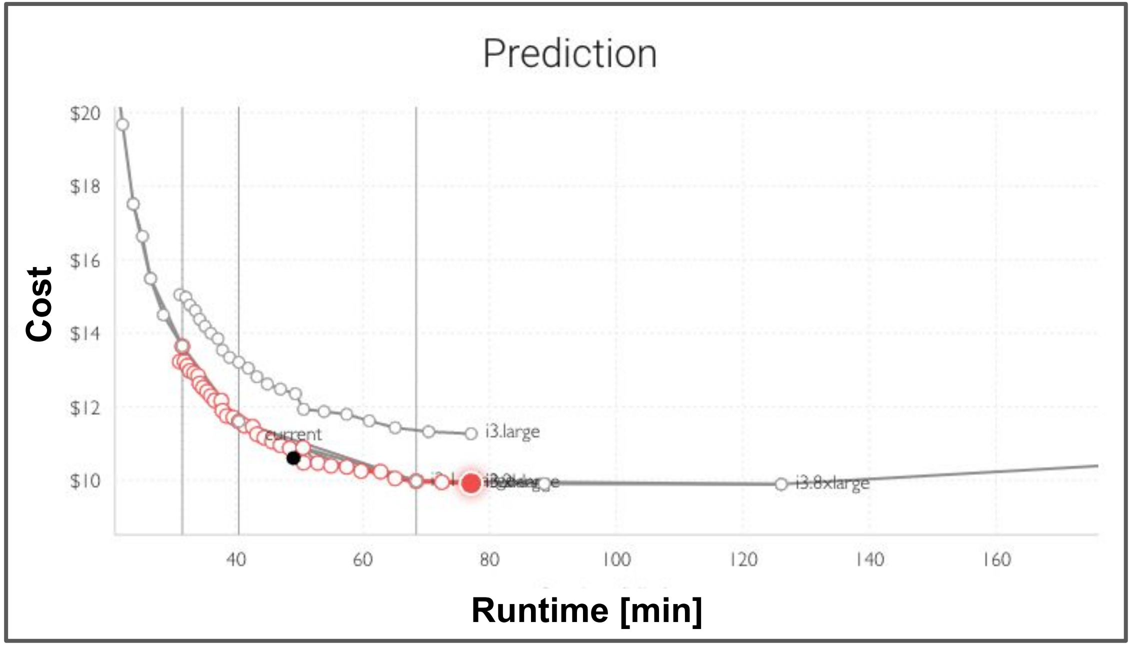
Predict how tasks transform to different infrastructure



Simulate task placement

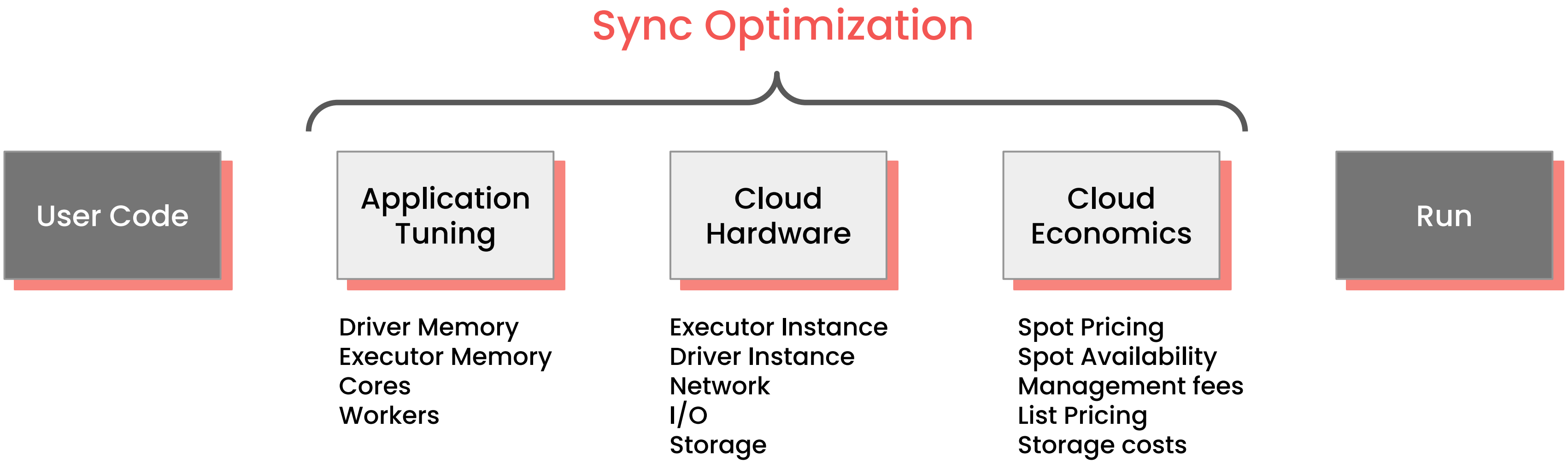


End Result – making it easy



All users have to do is select their cost and runtime

What is optimized?



Sync Optimization

User Code

Application Tuning

Cloud Hardware

Cloud Economics

Run

Driver Memory
Executor Memory
Cores
Workers

Executor Instance
Driver Instance
Network
I/O
Storage

Spot Pricing
Spot Availability
Management fees
List Pricing
Storage costs

- ✓ No application code changes
- ✓ Integrates with infrastructure
- ✓ Fully reversible / No risks

User Results in Production

01

Global Streaming Company

80%

Faster & cheaper

amazon EMR

02

Data Startup

47%

Faster & Same cost

databricks

03

Public Online Learning Company

55%

Cheaper & slower

amazon EMR

04

Global Digital Media Company

03

71%

Cheaper & 31% faster

databricks

05

Large Automotive Manufacturer

33%

Faster & 25% cheaper

amazon EMR

Sync

Live Demo

Kartik Nagappa
Staff Product Manager

Pete Tamisin
Technical CS Lead

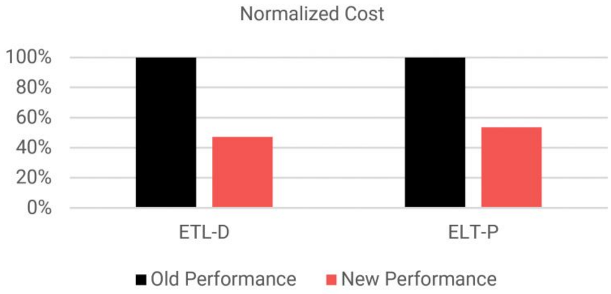
Tuning Spark on EMR with duolingo

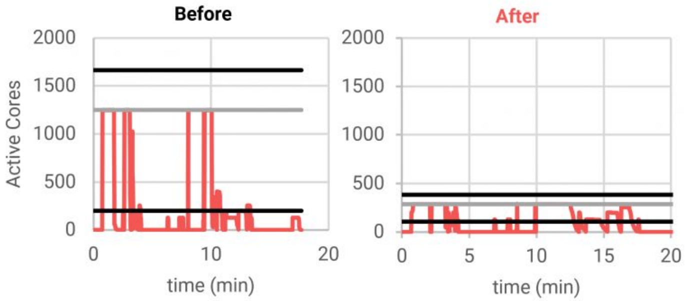
55% Reduction in Cost

4x reduction in cluster size
1664 vCPUs → 384 vCPUs

30% Increase in Runtime

5 minute increase in runtime
17 minutes → 22 minutes





Source: https://synccomputing.com/automatically-optimize-apache-spark-on-emr-aws/

Audience Poll

01

Have worked with / tuned / optimized Spark on EMR jobs?

02

Found it a major pain to tune / optimize Spark on EMR jobs?

03

Would like to be able to easily tune / optimize Spark on EMR jobs?

Live Demo Overview

01

Prerequisites

What you need to run this demo on your machine.

02

Setup

Create a virtual environment, download test logs, and install the Sync Client Library/CLI.

03

Run

Use the Sync Client Library to tune a Spark on EMR job.

Developer Interfaces

01

REST API

https://developers.synccomputing.com/reference

02

Sync Client Library/CLI

https://github.com/synccomputingcode/syncsparkpy

Live Demo Prerequisites

01

API Key

You'll need to sign up for an account with Sync to create an API Key.

02

*nix OS

The Sync Client Library/CLI has been tested to with Linux like systems.

03

Python 3.10

The Sync Client Library/CLI has been tested to run on Python 3.10.

Live Demo Prerequisites API Key

01

Sign up on https://app.synccomputing.com

02

Click Account in the left nav bar

03

Click Create Key in the API Keys section

Live Demo Setup Virtual Environment

01

# Clone our git repo which contains the library / CLI, and test logs

```
git clone https://github.com/synccomputingcode/syncsparkpy.git
```

02

# Source the install script to activate a virtual environment for the CLI

```
cd syncsparkpy
source demo/install_cli.sh
```

Live Demo Setup Sync Client Library/CLI

01

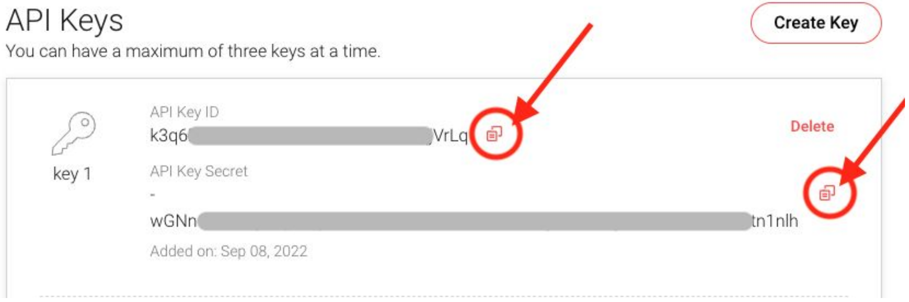
# Configure the Sync Client Library/CLI

```
sync-cli configure
```

02

# Verify configuration

```
sync-cli predictions platforms
```



API Keys

You can have a maximum of three keys at a time.

key 1

API Key ID

k3q6[redacted]VrLq

API Key Secret

-

wGNn[redacted]tn1nlh

Added on: Sep 08, 2022

Live Demo Run Context

01

[Input] Cluster config

The input cluster config tells us the kind of cluster you used to run your Spark job.

Cluster configs contain information about the nodes in the cluster and what Spark parameters are configured.

02

[Input] Spark event log

The input Spark event log tells us how your Spark job was executed in the cluster.

Spark event logs contain information on DAG execution and resource utilization.

03

[Output] Tuned cluster configs

The Sync Autotuner returns a list of tuned cluster configs. This list can be filtered to a single recommendation that can best meet your business needs in terms of job cost and/or runtime.

The Sync Autotuner uses the input cluster config and Spark event log to generate a list of tuned cluster configs. These tuned configs can help save on cost and/or runtime.

Live Demo Run

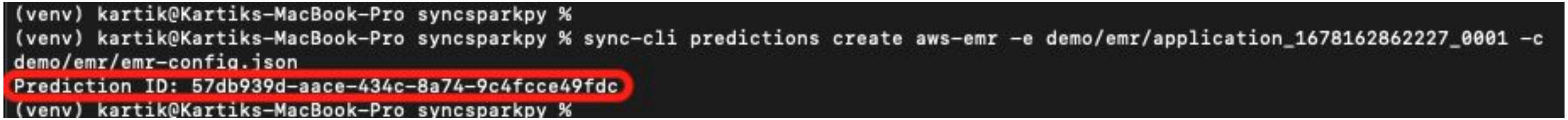
01

# Initiate a prediction run for Spark on EMR

```
sync-cli predictions create aws-emr
-e demo/emr/application_1678162862227_0001 -c demo/emr/emr-config.json
```

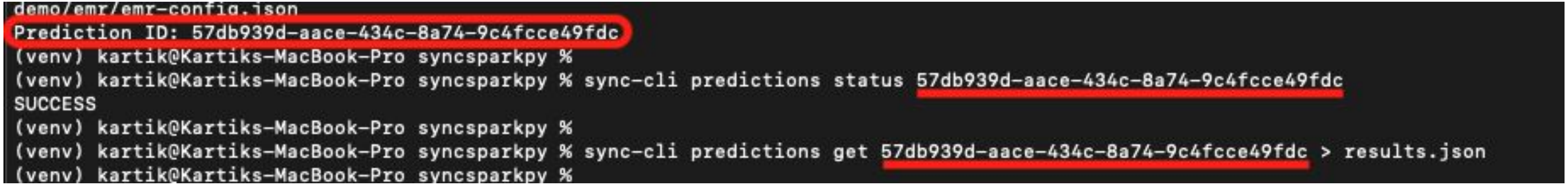
Eventlog

Cluster Config



```
(venv) kartik@Kartiks-MacBook-Pro syncsparkpy %
(venv) kartik@Kartiks-MacBook-Pro syncsparkpy % sync-cli predictions create aws-emr -e demo/emr/application_1678162862227_0001 -c
demo/emr/emr-config.json
Prediction ID: 57db939d-aace-434c-8a74-9c4fcce49fdc
(venv) kartik@Kartiks-MacBook-Pro syncsparkpy %
```

Live Demo Run



```
demo/emr/emr-config.json
Prediction ID: 57db939d-aace-434c-8a74-9c4fcce49fdc
(venv) kartik@Kartiks-MacBook-Pro syncsparkpy %
(venv) kartik@Kartiks-MacBook-Pro syncsparkpy % sync-cli predictions status 57db939d-aace-434c-8a74-9c4fcce49fdc
SUCCESS
(venv) kartik@Kartiks-MacBook-Pro syncsparkpy %
(venv) kartik@Kartiks-MacBook-Pro syncsparkpy % sync-cli predictions get 57db939d-aace-434c-8a74-9c4fcce49fdc > results.json
(venv) kartik@Kartiks-MacBook-Pro syncsparkpy %
```

02

# Get prediction Status

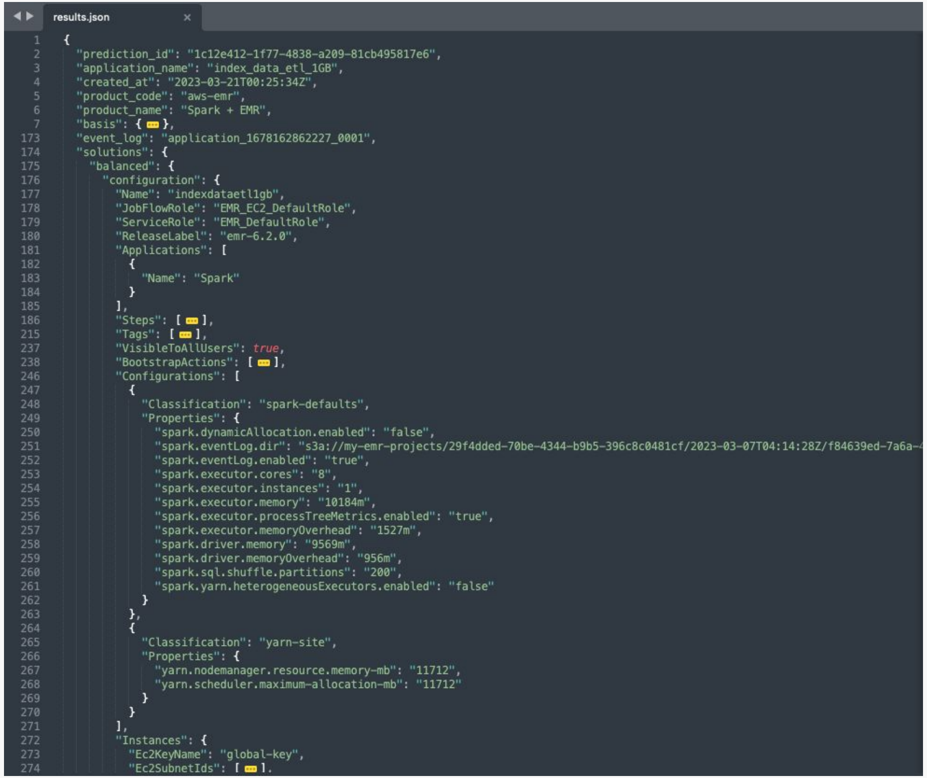
```
sync-cli predictions status prediction_id
```

03

# Get prediction results

```
sync-cli predictions get prediction_id > result.json
```

Live Demo Run Results



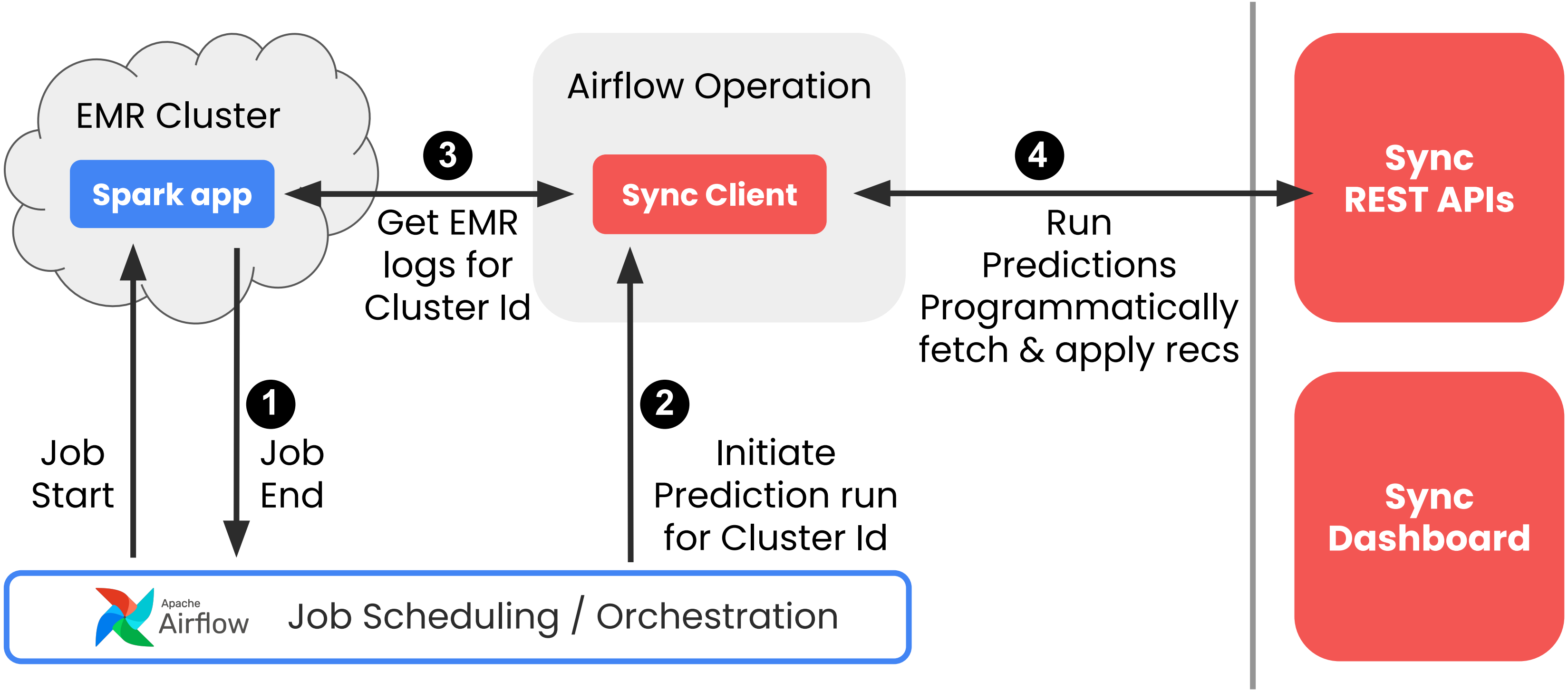
```
{
  "prediction_id": "1c12e412-1f77-4838-a209-81cb495817e6",
  "application_name": "index_data_etl_1GB",
  "created_at": "2023-03-21T00:25:34Z",
  "product_code": "aws-emr",
  "product_name": "Spark + EMR",
  "basis": { ⋯ },
  "event_log": "application_1678162862227_0001",
  "solutions": {
    "balanced": {
      "configuration": {
        "Name": "indexdataetl1gb",
        "JobFlowRole": "EMR_EC2_DefaultRole",
        "ServiceRole": "EMR_DefaultRole",
        "ReleaseLabel": "emr-6.2.0",
        "Applications": [
          {
            "Name": "Spark"
          }
        ],
        "Steps": [ ⋯ ],
        "Tags": [ ⋯ ],
        "VisibleToAllUsers": true,
        "BootstrapActions": [ ⋯ ],
        "Configurations": [
          {
            "Classification": "spark-defaults",
            "Properties": {
              "spark.dynamicAllocation.enabled": "false",
              "spark.eventLog.dir": "s3a://my-emr-projects/29f4dded-70be-4344-b9b5-396c8c0481cf/2023-03-07T04:14:28Z/f84639ed-7a6a-
              "spark.eventLog.enabled": "true",
              "spark.executor.cores": "8",
              "spark.executor.instances": "1",
              "spark.executor.memory": "10184m",
              "spark.executor.processTreeMetrics.enabled": "true",
              "spark.executor.memoryOverhead": "1527m",
              "spark.driver.memory": "9569m",
              "spark.driver.memoryOverhead": "956m",
              "spark.sql.shuffle.partitions": "200",
              "spark.yarn.heterogeneousExecutors.enabled": "false"
            }
          },
          {
            "Classification": "yarn-site",
            "Properties": {
              "yarn.nodemanager.resource.memory-mb": "11712",
              "yarn.scheduler.maximum-allocation-mb": "11712"
            }
          }
        ],
        "Instances": {
          "Ec2KeyName": "global-key",
          "Ec2SubnetIds": [ ⋯ ],
```

Key highlights

- cluster configs use RunJobFlow
- tuned configs are “plug and play”

- response is in JSON
- input cluster config under basis
- tuned configs are under solutions

A Customer’s Solution


EMR Cluster
Spark app
3
Get EMR logs for Cluster Id
Airflow Operation
Sync Client
4
Run Predictions Programmatically fetch & apply recs
Sync REST APIs
Sync Dashboard
Job Start
1
Job End
2
Initiate Prediction run for Cluster Id
Apache Airflow
Job Scheduling / Orchestration

Q&A - Follow up

More Questions/Feedback after the workshop?

- Visit our booth to chat!
- Email: support@synccomputing.com

Want to conduct a formal proof of concept?

- Reach out to schedule a meeting with Pete next week (support@synccomputing.com)

Linkedin

Medium

www

Raffle Time!