# Unravel Newsletter: February 2020

Greetings from Unravel,

Hope you have had a great start to the year so far! On our end, we have been keeping busy with several interesting feature developments.

In this month's newsletter, we start with discussing about Unravel REST APIs and use cases that our customers are utilizing these for. Then, we will discuss work in progress features for achieving price-performance trade-off for Azure Databricks. We will continue with some community highlights in the area of Data and Analytics and include links to recent recordings and upcoming events that you may find interesting.

## Unravel APIs

While the intuitive and easy to use Unravel user interface serves many of the Application Performance Management (APM) and Operational Intelligence use cases, several of Unravel customers have also expressed a need for Unravel REST APIs to enable use cases that are beyond those that can be met by the Unravel user interface alone. For example,

- **Custom Dashboards/Widgets**: Unravel APIs can be used to incorporate information available from Unravel in existing dashboards that consolidate information from multiple sources. In some cases, customers have also created custom dashboards to display reports that achieve their organization's specific reporting needs.
- **Custom Reports**: Some of our customers have wanted to create various types of custom reports published in their common reporting portal, for example a chargeback report that has the ability to group by Cluster or a custom reports that combines YARN and Impala Chargeback etc.
- **Sending Custom Notifications**: While Unravel Auto-Actions can be used for sending email alerts (among other things), in some cases, there may be a need to customize the content of those alerts specific to the business requirements of a given organization. For example, incorporate team specific guidelines in the alert message. Unravel APIs (e.g. /autoactions) can be used to retrieve the active auto-actions, organization specific content can be added to the retrieved information and the combined information can be sent out as a custom notification.
- **Customized Analysis relevant for the Organization**: While Unravel provides a wide variety of valuable insights and recommendations to run

your applications more efficiently and with better performance, there are times customers have requested for incorporating team/organization-specific rules as well. So one way to achieve this is to pull the various application details as well as insights and recommendations using Unravel REST APIs (e.g. /common/app/{app_id}/summary, /common/app/{app_id}/errors, /common/app/{app_id}/extendedsummary, /common/app/{app_id}/logs, /common/app/{app_id}/recommendation) and applying organization specific rules and surface the new set of insights and recommendations.

- **Using Unravel Recommendations for CI/CD**: Platform owners want to make sure that the applications are configured/tuneed to run optimally before propagating pipelines from lower environments like Dev/Test to Production. To achieve this, customers are using Unravel APIs to retrieve insights and recommendations for applications run in lower environments (/common/app/{app_id}/recommendation), if there are none, the application is propagated to production. If there are available recommendations, an automated email is sent to the application's owner to first apply those and then attempt to propagate the application to production.
- **Building chatbots for various functionality**: Unravel APIs are also being used to build a variety of chatbots. Here's a presentation and video demonstration of one such chatbot: An AI Powered Chatbot to Simplify Apache Spark Performance Management

Unravel REST APIs have been available for a while now and we are seeing a continued increase in their use and also in the variety of use cases they are being used for. Easy and intuitive to use, Unravel REST APIs are a great way for achieving net new custom use cases and customizing ones served by Unravel UI. Please check out the documentation here and do reach out to us if you would like to learn more about possible use cases or consult on how you can achieve your custom ones using Unravel APIs.

## Achieving Price-Performance Trade-off for Azure Databricks Jobs

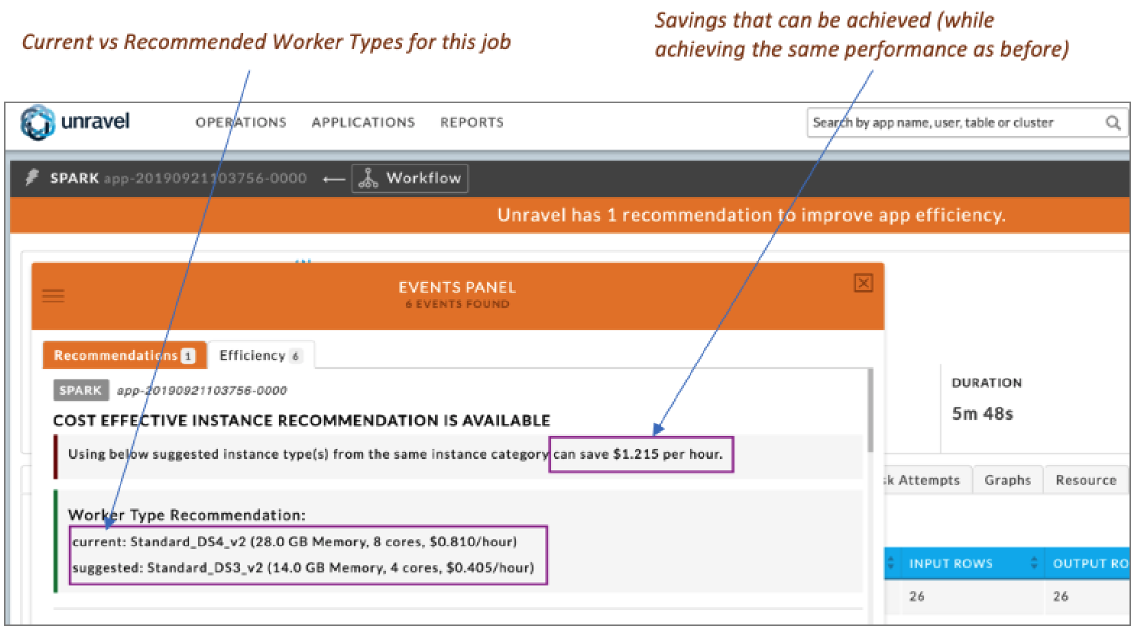
Mission-critical Data Engineering tasks such as Extract Transform Load (ETL), preparing training models, Data Cleansing etc. to enable critical business use cases are easy to automate and schedule using Databricks Automated (Job) Clusters. It is important for such job runs to complete within the stipulated duration so that business SLAs can be met. However, at the same time, it is also important for companies to keep costs under control and have the ability to exercise custom price-performance trade-off for various jobs.

One of the challenges faced by various companies we have spoken to is to determine the Optimal Cluster Topology for a given Databricks Job. This includes figuring out the following:

- What are the optimal instance types to use for Driver and Worker nodes?
- Whether or not to enable auto-scaling for the Automated cluster?
- What is the optimal number of worker nodes to set (and in the case when auto-scaling is enabled, what is the optimal min and max number of worker nodes to set)?

Since in the case of Data Engineering use cases, the same job run is executed periodically, there is a wealth of historical information to understand the resource usage profile of these job runs and how various factors affecting the resources' needs (e.g. amount of input data processed) are changing over time.

Unravel has developed a non-intrusive and extremely light data collection tool to collect granular data about each job run and a powerful algorithm for analyzing that data (and additional information related to instance types, their costs etc.) in near real-time to determine the optimal cluster topology for a given Databricks job. As an example, Unravel now provides recommendations for Driver and Worker VM types which when used result in the same performance as was achieved with the previous choice of VM types, but for a lesser cost.



Current vs Recommended Worker Types for this job
Savings that can be achieved (while achieving the same performance as before)
unravel
OPERATIONS
APPLICATIONS
REPORTS
Search by app name, user, table or cluster
SPARK app-20190921103756-0000
Workflow
EVENTS PANEL
6 EVENTS FOUND
Recommendations 1
Efficiency 6
DURATION
5m 48s
k Attempts
Graphs
Resource
INPUT ROWS
OUTPUT RO
26
26

Unravel has 1 recommendation to improve app efficiency.

SPARK app-20190921103756-0000

COST EFFECTIVE INSTANCE RECOMMENDATION IS AVAILABLE

Using below suggested instance type(s) from the same instance category can save $1.215 per hour.

Worker Type Recommendation:
current: Standard_DS4_v2 (28.0 GB Memory, 8 cores, $0.810/hour)
suggested: Standard_DS3_v2 (14.0 GB Memory, 4 cores, $0.405/hour)

*Unravel VM Type Recommendations*

We are now working towards a feature set that will allow the user to choose the desired price-performance trade-off and answers the questions listed earlier, thereby enabling customers like you to meet their set SLAs in the most cost-optimal way. Stay tuned for more developments in this area!

Keep subscribing to these newsletters for upcoming exciting Unravel features and functionality or get in touch with us directly to get your feature requests in (hello@unraveldata.com)!

## Conference Sessions and Recordings

- Check out this webinar recording: Best Practices: Monitoring End-to-End HBase Performance with Unravel
- Check out this webinar recording: Best Practices: Troubleshoot and Optimize Spark Data Pipelines with Unravel
- Check out this webinar recording: 5 Ways to Slash Your On-Premises Hadoop Costs
- Check out How to get started with Unravel for Amazon EMR from the AWS Marketplace
- Check out How to get started with Unravel for Azure HDInsight from the Azure Marketplace

---

## Upcoming Events & Webinars

- Join us at Fogo de Chao in Atlanta for Atlanta IT Leaders & "Big Data" Professionals Luncheon (sponsored by Microsoft Azure & Unravel Data), in Atlanta, Georgia, Feb 26, 2020.
- Check out our next webinar on Feb 20: Best Practices: Troubleshoot and Optimize Spark Data Pipelines with Unravel

---

## Community Highlights

- Engineering SQL Support on Apache Pinot at Uber: Uber writes about how they've integrated Pinot, their real-time analytics system, with Presto for SQL queries. The article describes the architecture, and how they improved the connectors performance with predicate/limit/aggregate/more pushdown, and how it performs in practice.
- Streams and Monk – How Yelp is Approaching Kafka in 2020: Yelp writes about their Kafka infrastructure, which has several components. The post describes two of them in detail, the Stream Discovery and Allocation service (which enforces schemas and defines a stream as either fire and forget or acked) and "Monk Leaf" which is a service that runs locally and proxies to Kafka (implementing either the acked or fire and forget semantics). These components provide a platform that make it easy for developers to deploy applications and get data into the Kafka data pipeline.
- Spark UDAF could be an option!: A post from Teads describes both how to speed up a Spark application using a Spark User-Defined Aggregate Functions (UDAF) as well as how to optimize Spark applications in general. Their article walks through how they sped up one of their applications from 28 mins to 9 mins. Several of the optimizations are informed by the Spark execution DAG, several of which they analyze in the post.
- Presto in 2019: Year in Review: Presto's 2019 year in review—covering new syntax (e.g. adding comments and fetching just the first N rows), query

optimizations (e.g. improvements to the cost based optimizer and lazy materialization), new connectors (elasticsearch, google sheets), and much more. The post also looks at what's next for Presto in 2020.

---

## Resources

- Learn more about Unravel.
- Online Product Demo
- Unravel Partners
- Unravel Product Releases and Documentation
- Unravel Datasheet
- More Unravel News

Contact Us. Sign Up for 30-day Trial. Take Unravel for a Test Drive.