unravel

Whitepaper

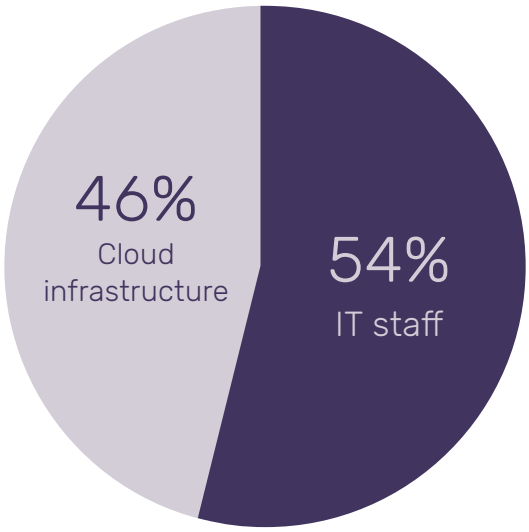
Manage costs on Amazon EMR

Operationalizing big data to optimize and automate modern applications and infrastructure.

Main cost drivers

Amazon EMR is a service that makes it easy to process large amounts of data efficiently. It uses Hadoop processing combined with several AWS products to do such tasks as data mining, log file analysis, machine learning and scientific simulation through a managed service. It has become very popular as a cloud alternative to physical hosting of complex infrastructure. However, customers are finding unexpected costs eating into their cloud budget. Furthermore, lack of visibility to root cause and general inefficiency is costing organizations thousands, if not millions in operating their Amazon EMR environment.

The following chart shows a breakdown of typical Total Cost of Ownership for Amazon EMR project costs:



Controlling IT Staff productivity costs (quick troubleshooting, meeting SLAs) are just as important as controlling cloud infrastructure costs (compute, storage, networking) in Amazon EMR. To address both human capital and infrastructure costs, the following issues are the largest contributors to cost overruns in Amazon EMR:

1. **Poorly performing or failed jobs -** EMR environments often have hundreds, if not thousands of jobs running simultaneously. Identifying poorly performing jobs often occur after resources are consumed.
2. **Getting to root cause analysis -** EMR administrators sift through thousands of lines of logs to identify and remediate the root cause of failed or underperforming jobs.
3. **Missed service level agreements -** Missed SLAs for business applications cost an organization real dollars in missed opportunity or financial penalties.
4. **DevOps/IT Ops blame game -** Failure to quickly identify root cause will lead to reactionary finger-pointing between the app developers ("it's the cluster") and IT Ops ("it's the code).
5. **Chargeback/showback -** Inadequate tracking of which department or user may be abusing cluster resources perpetuate cost overruns.

The issues in DataOps have been present for some time, but we are now better assessing the costs of these inefficiencies in big data projects. These issues contribute to direct organizational costs attributed to developing and operating an Amazon EMR environment. As a result, the true costs of missed SLAs could have vast financial impacts on your business:

	Consequences of missed SLAs
Banking	Fraudulent transactions in banking not discovered in time, leading to millions in theft and fines.
Healthcare	Patient profiles not accurately described, leading to lapses in care and millions in additional healthcare costs.
Retail	Customer demand not analyzed for a particular product, overestimating inventory leading to millions in waste.
Manufacturing	Equipment failure not detected with accuracy, leading to costly maintenance calls.

Tuning Amazon EMR requires awareness of resources

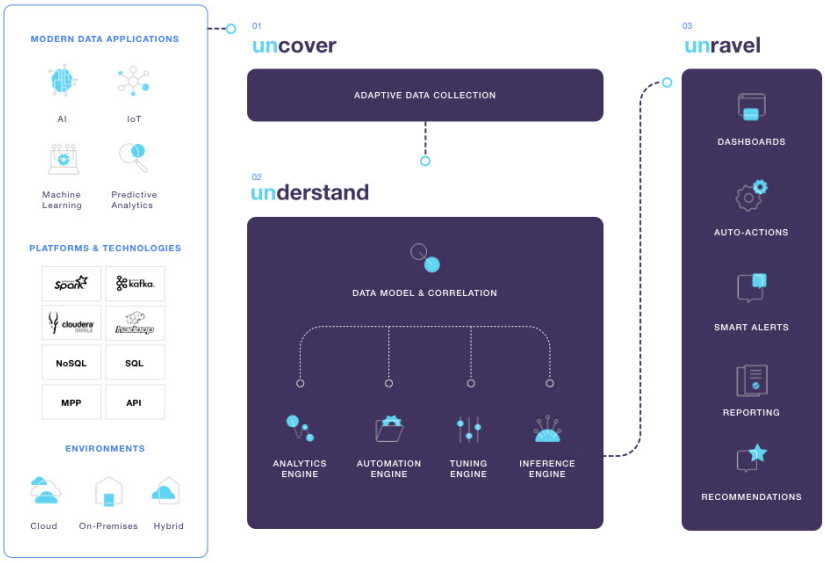
There are a few money saving techniques which will help avoid cost overruns:

- **Choose Proper AWS EC2 topology -** Properly assigning the correct AWS EC2 instance types and number of nodes is essential to control spending.
- **Choose proper storage and networking -** Decide on the networking configuration (is public IP necessary?) as well as anticipating optimal storage (S3 types).
- **Code optimization -** Inefficient Spark or Hive code in your Amazon EMR application can kill your performance or cause failures - tight collaboration between DevOps and ITOps is essential to provision right level of resources.
- **Workload identification -** Not all Amazon EMR jobs are created equal. Data warehousing will have more persistent resource requirements (and cost) than ephemeral data engineering jobs, for example. Anticipate these needs and adjust accordingly.
- **Manual instance matching -** developers are able to choose from a variety of workloads (data analytics, data engineering, data engineering light) as well as Premium/Standard levels within each workload.
- **Unused compute elimination -** Unchecked rogue resources can be a large contributor to waste. Users must understand when auto termination is warranted.

These techniques require manual data collection, but can be effective for managers versed in log collection techniques for Spark, Hive, MapReduce and Kafka.

Unravel's solution

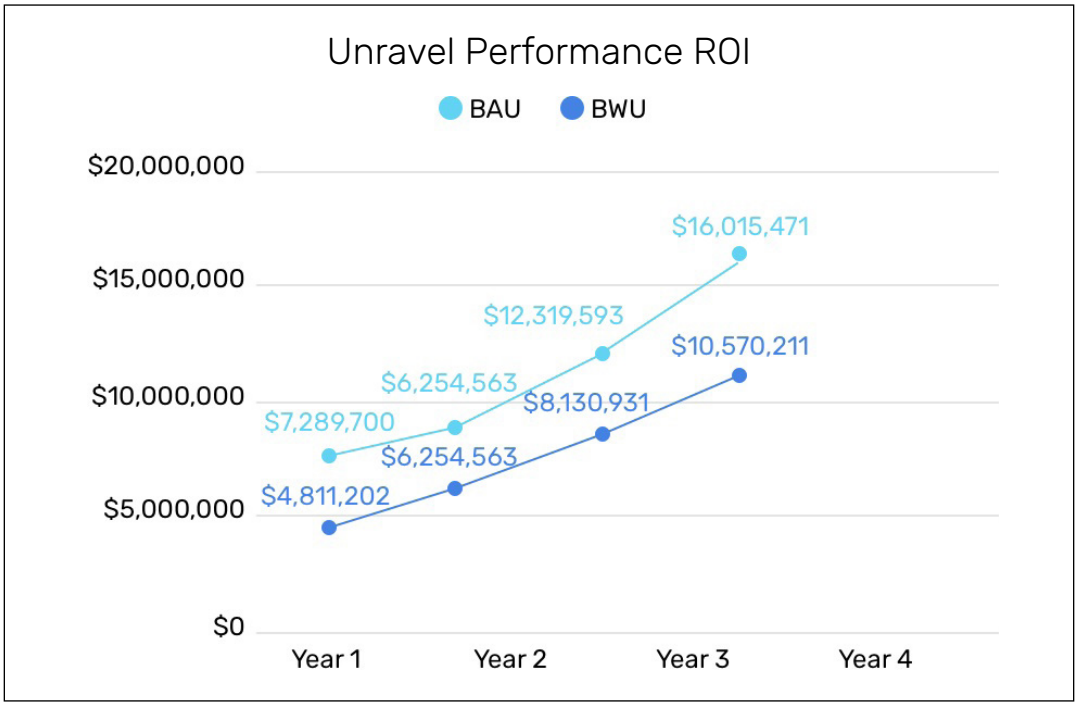
Alternatively, Unravel provides a turn-key solution for cost assurance.



With Unravel	Without Unravel
Time to resolution: Hours to minutes	**Time to resolution:** Days to weeks
Required Infrastructure: S3 - $100K/year EC2 - $200K/year EMR - $100K/year	**Required Infrastructure:** S3 - $150K/year EC2 - $500K/year EMR - $200K/year
Required Team: Big Data Engineers - 10 hours/week	**Required Team:** Big Data Engineers - 40 hrs/ week
DevOps: 5 hours/week	**DevOps:** 10 hours/week

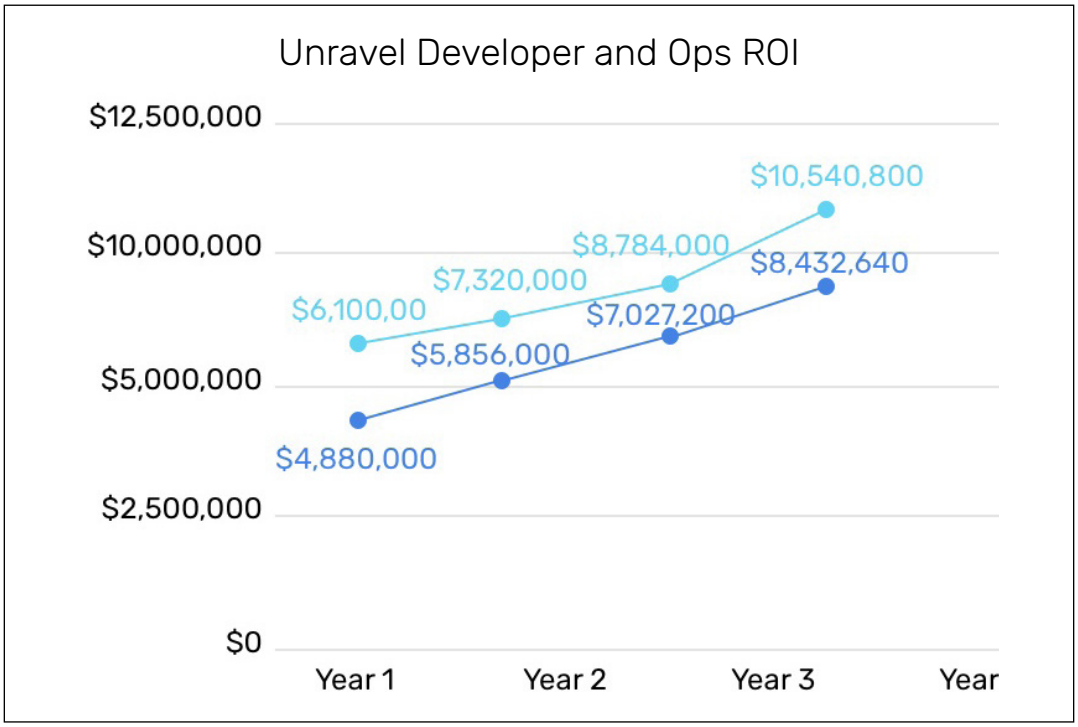
Example: Unravel banking customer TCO

The following diagram illustrates a real life banking customer's infrastructure savings using Unravel:



Unravel will drive $9MM in infrastructure savings over 3 years.
Assumes 20% annual growth rate.

The following diagram illustrates a real life banking customer's labor savings using Unravel:



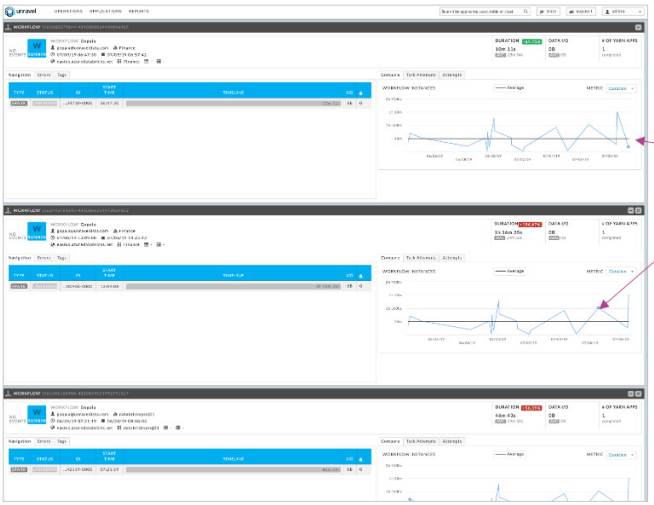
Unravel will drive $6.5MM in human capital savings over 4 years.
Assumes 20% employee growth.

Given the growth rate of data, and the appetite for enterprise organizations to analyze this data in real time, companies are facing increasing costs in infrastructure (both on-premises and cloud) and human capital to maintain analytics projects.

Unravel presents an elegant solution to provide cost assurance in Amazon EMR environments, automatically collecting and correlating jobs data in context, helping companies pinpoint issues, often before they happen.

Unravel techniques for Amazon EMR tuning

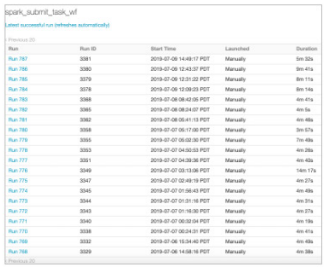
1. **Poorly performing or failed jobs**
   Visualize jobs and job runs: track individual jobs to assess performance improvements.



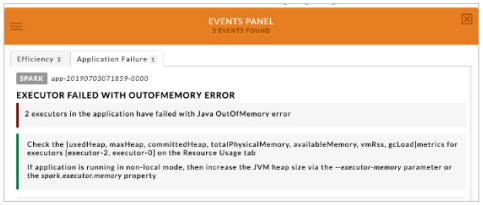
- Easily compare runs of a job and their KPIs
- Find out the reasons for difference in perf

Various runs of a job

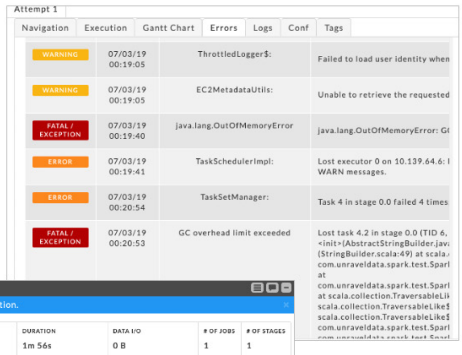
Easy to compare various attributes – Duration, I/IIO across runs of the same job etc.



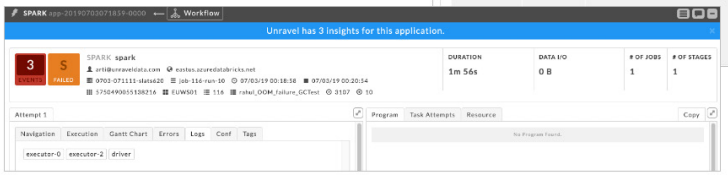
2. **Getting to root cause analysis**
   Root cause analysis: if we see a resource having issues we can use point of time KPIs to identify the state of the applications at the point of failure.



Specific reason for failure extracted and presented - RCA is fast and easy. No need to dig through logs



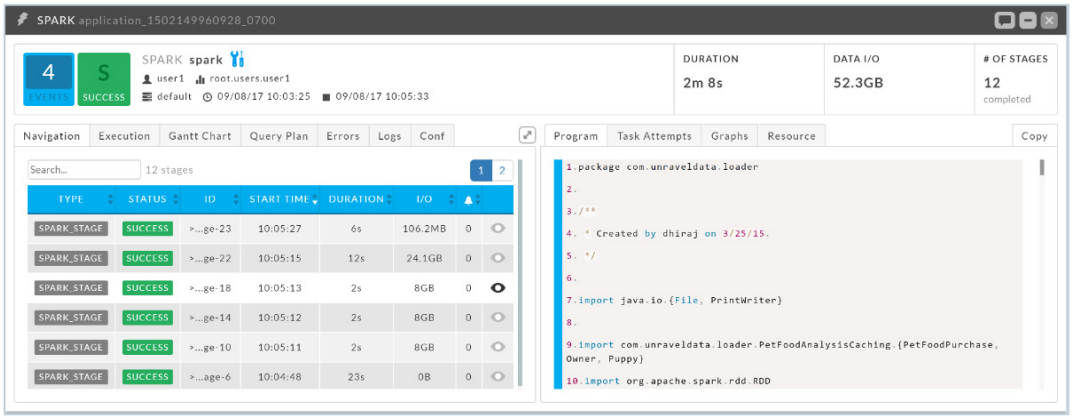
List of errors and relevant information extracted and surfaced



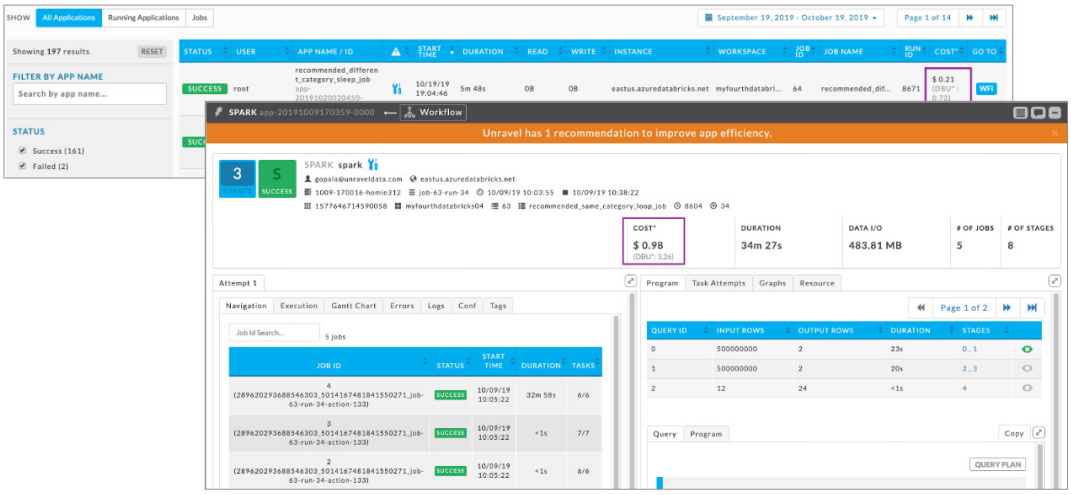
Relevant logs retrieved and made easily available

3. **Missed service level agreements**
   Single pane of glass view: ensure maximum SLAs by pinpointing possible failures before they happen.

4. **DevOps/ITOps blame game**
   AI-Driven recommendations: pinpoint whether the issue resides in the code or the cluster configuration via automatic recommendations.



5. **Chargeback/showback**
   Per application costs: i Identify resource wasters by application or by user to drive accountability for responsible EMR resource use.



Interested in learning more? Contact us at hello@unraveldata.com

About Unravel

Unravel radically simplifies the way businesses understand and optimize the performance of their modern data applications – and the complex pipelines that power those applications. Providing a unified view across the entire stack, Unravel's AI-powered data operations platform leverages AI, machine learning, and advanced analytics to offer actionable recommendations and automation for tuning, troubleshooting, and improving performance – both today and tomorrow.

By operationalizing how you do data, Unravel's solutions support modern big data leaders, including Kaiser Permanente, TIAA, Adobe, Deutsche Bank, Wayfair, and Neustar. The company is headquartered in Palo Alto, California, and is backed by Menlo Ventures, GGV Capital, M12, Data Elite Ventures, and Jyoti Bansal. To learn more, visit unraveldata.com.

unravel