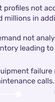
unravel

Managing costs on Azure Databricks

Unexpected costs are eating into cloud budgets and lack of visibility to root cause and general inefficiency is costing organizations thousands, if not millions in operating their Azure Databricks environment.

Controlling IT staff productivity costs (quick troubleshooting, meeting SLAs) are just as important as controlling cloud infrastructure costs (compute, storage, networking) in Azure Databricks.

- **Poorly performing or failed jobs**
- **Getting to root cause analysis**
- **Missed service level agreements**
- **DevOps/IT Ops blame game**
- **Chargeback/showback**



True cost go beyond data infra cost

The issues in DataOps contribute to direct organizational costs attributed to developing and operating an Azure Databricks 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.

Azure Databricks pricing quick guide

WORKLOAD	DBU PRICES STANDARD TIER	DBU PRICES STANDARD TIER
Data analytics	$0.40/DBU-hour	$0.55/DBU-hour
Data engineering	$0.15/DBU-hour	$0.30/DBU-hour
Data engineering light	$0.07/DBU-hour	$0.22/DBU-hour

Data analytics / Interactive workloads to analyze data collaboratively with notebooks

Data engineering / Automated workloads to run fast and robust jobs via API or UI

Data engineering light / Automated workloads to run robust jobs via API or UI

Tuning Azure Databricks is largely a manual effort

Choose proper Azure VM topology / Properly assigning the correct Azure VM 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 (standard, premium, ADLS).

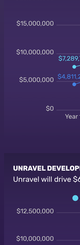
Code optimization / Inefficient Spark code in your Azure Databricks application can kill your performance or cause failures - tight collaboration between DevOps and ITOps is essential to provision right level of resources.

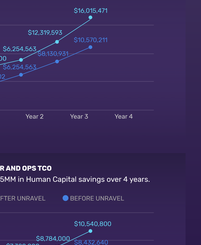
Workload identification / Not all Azure Databricks jobs are created equal. Data Analytics will have more persistent resource requirements (and cost) than Data Engineering. Anticipate these needs and adjust accordingly.

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 / Automated termination of clusters are included as standard features in Azure Databricks, but the user must understand when auto termination is warranted.

Unravel's solution





WITH UNRAVEL	WITHOUT UNRAVEL
Time to resolution: Hours to minutes	**Time to resolution:** Days to weeks
Required infrastructure: ADLS - $100K/year Azure Linux VMs - $200K/year DBU - $200K/year	**Required infrastructure:** ADLS - $150K/year Azure Linux VMs - $500K/year DBU - $400K/year
Required team: Big data engineers - 10 hours/week DevOps - 5 hours/week	**Required team:** Big data engineers - 40 hrs/week DevOps - 10 hours/week

Unravel techniques for Azure Databricks tuning

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

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.

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

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

Chargeback/showback / Per application costs: identify resource wasters by application or by user to drive accountability for responsible Databricks resource use.

Example: Unravel banking customer TCO

UNRAVEL PERFORMANCE TCO

Unravel will drive $9MM in infrastructure savings over 3 years.

UNRAVEL DEVELOPER AND OPS TCO

Unravel will drive $6.5MM in Human Capital savings over 4 years.

Interested in learning more?

Contact us at **hello@unraveldata.com**

unravel