Lessons Learned while Migrating Data Pipelines from Enterprise Schedulers to Airflow

Shivnath Babu  
CTO/Cofounder @ Unravel  
Adjunct Professor @ Duke University

Hari Nair  
Senior Software Engineer @ Unravel
About the speakers

**Shivnath Babu**
Cofounder/CTO at Unravel
Adjunct Professor of Computer Science at Duke University
Focusing on manageability of data pipelines & modern data stack
Recipient of US National Science Foundation CAREER Award, IBM Faculty Award, HP Labs Innovation Research Award

**Hari Nair**
Senior Software Engineer @ Unravel
Team Lead, Customer Success and Innovation
Focusing on Data Science and Insights
Unravel radically simplifies DataOps & has strong adoption across platforms & industries

**uncover**
- Brings together information about all your apps, clusters, resource utilization, users, & datasets in a single place

**understand**
- Creates end-to-end view of data pipelines to easily track & understand issues
- Tracks & reports on usage across environments
- Checks for & alerts on anomalous behavior

**unravel**
- Uses AI/ML to troubleshoot & optimize apps to meet desired performance & cost needs
- Spots & fixes inefficient usage
- Ensures efficiency, quality, & performance of all apps in development & production
Many enterprises are modernizing their data stack and pipelines
Many enterprises are modernizing their data stacks and pipelines

Large clusters supporting multiple apps and tenants
Less agile
Harder to scale

Smaller, decentralized, app-level clusters
Very agile
Easier to scale
Goals of modernization

- Improve agility
- Resources no longer become the constraint
- Reduce cost

Why Airflow gets picked as part of modernization:
- Well suited for agile development
- Better suited for cloud-native architectures than traditional schedulers
- Available as a service
Two main phases of modernization

Phase 1: Assess and Plan

Phase 2: Migrate, Validate, and Optimize
Assess and Plan: Lessons Learned
Assess & plan phase of modernization

- Pipeline discovery
- Resource usage analysis
- Dependency analysis
- Complexity analysis
- Mapping to target environment
- Cost estimation for target environment
- Migration effort estimation
Assess & plan phase: Lessons learned

Pipeline discovery itself can be challenging
- Multiple enterprise schedulers may be in use, e.g., Autosys, Informatica, Oozie, Pentaho, Tidal, etc.
- No common pattern may exist

Fine-grained tracking is needed for accurate resource usage and dependency analysis
Fine-grained tracking for accurate planning

Metadata

Input Data

KPIs

Output Data

- **DURATION**: 10m 47s
- **DATA I/O**: 101.60 GB
Fine-grained tracking for accurate planning
The annotated lineage graph
Picking the best migration execution strategy

Compute-heavy pipeline with bursty resource needs

Popular data asset
Migrate, Validate, and Optimize: Lessons Learned
Migrate, Validate, & Optimize phase of modernization

Criteria to Consider

Correctness

Performance

Cost

Undesired Behavior

Wrong results, Failing pipelines

Missed SLAs, Growing lag/backlog

Cost overruns, Going over budget
Guaranteeing pipeline correctness after migration

Ensure that the right checks are in place to validate correctness after the migration

Example checks:
• Daily partitions of Table “SignupsAndSubs” should have at least 1000 records
• “customerPinNumber” should not be NULL

Tools like Great Expectations make it easy to define checks
Guaranteeing pipeline performance after migration

Ensure that baselining is done and SLAs are defined to ensure performance needs are met after the migration

Example SLAs:
• Pipeline should finish by 6:00 AM PST
• Data in dashboard generated by the pipeline should not be older than 10 mins

SLAs can be defined in Airflow

Tools like Unravel help pinpoint bottlenecks and suggest performance fixes
Controlling pipeline costs after migration

Ensure that cost budget estimation & planning are done before the migration

Example budget specifications:
• Cost of any one run of the “BI-report” pipeline should not exceed $100
• Budget for the pipelines generating the “probable_churn” table is $1M/month

Tools like *Unravel* help with cost projection and also recommend fixes for cost inefficiencies
Demo
Sign up for a free trial!

https://unraveldata.com/saas-free-trial

shivnath@unraveldata.com
hari@unraveldata.com

Check out our next talk:

Data Pipeline HealthCheck for
Correctness, Performance, and Cost Efficiency