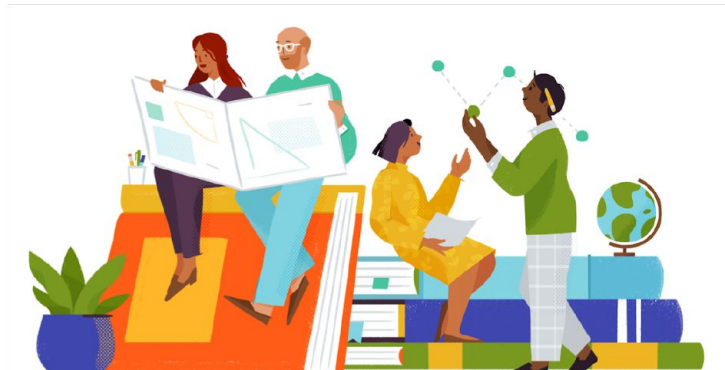


Operating Data Pipeline with Airflow

Ananth Packkildurai



About Slack



Public launch: 2014



**1000+ employees across
7 countries worldwide**
HQ in San Francisco



Diverse set of industries

including software/technology, retail, media,
telecom and professional services.

March 2016



5

Data Engineers



350+

Slack employees



2M

Active users

April 2018

10

Data Engineers

1000+

Slack employees

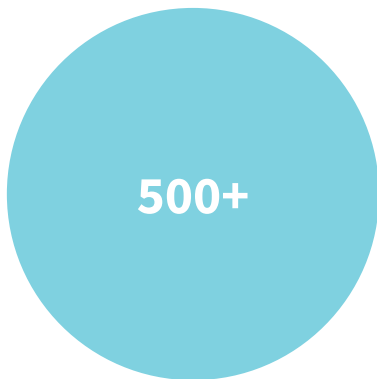
6M

Active users

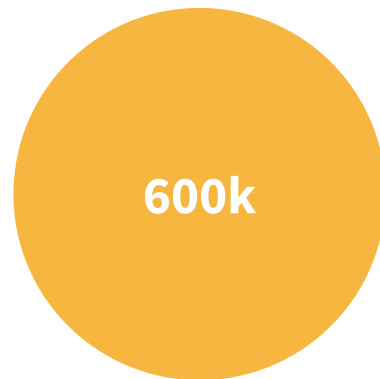
Data usage



access data
warehouse



Tables



Events per sec at
peak

Airflow stats

240+

Active Dags

5400+

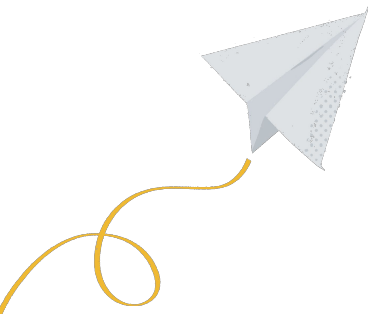
Tasks Per Day

68

Contributors

Agenda

1. Airflow Infrastructure
2. Scale Airflow Executor
3. Pipeline Operations
4. Alerting and monitoring





Airflow infrastructure



Airflow infrastructure

- Local Executor
- Tarball code deployment
- Continuous deployment with Jenkins
- Flake8, yapf & pytest
- `airflow.sh` shell utility to ensure consistent development environment for all the users.



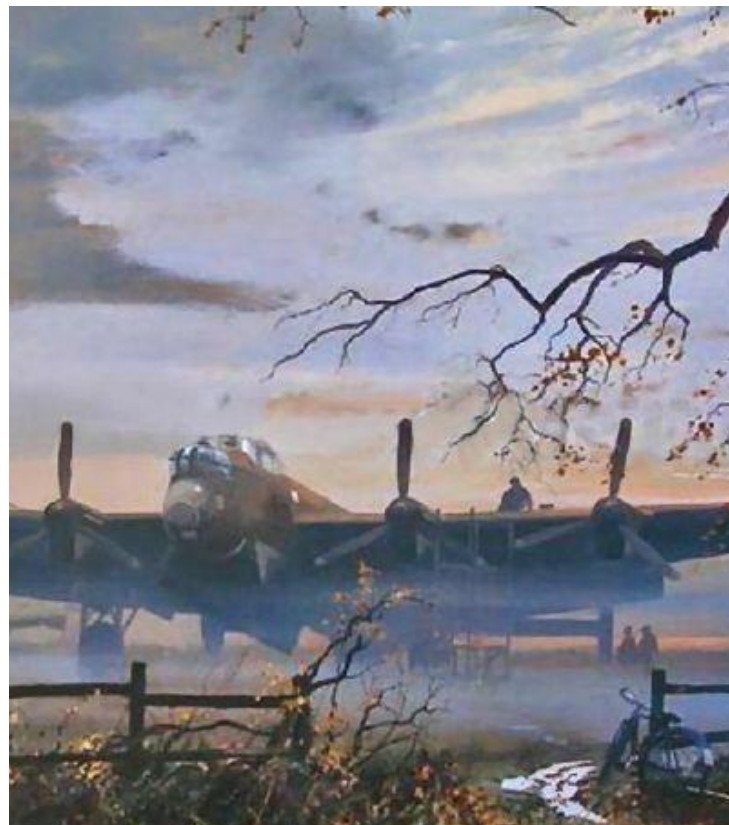
Scale Airflow Executor



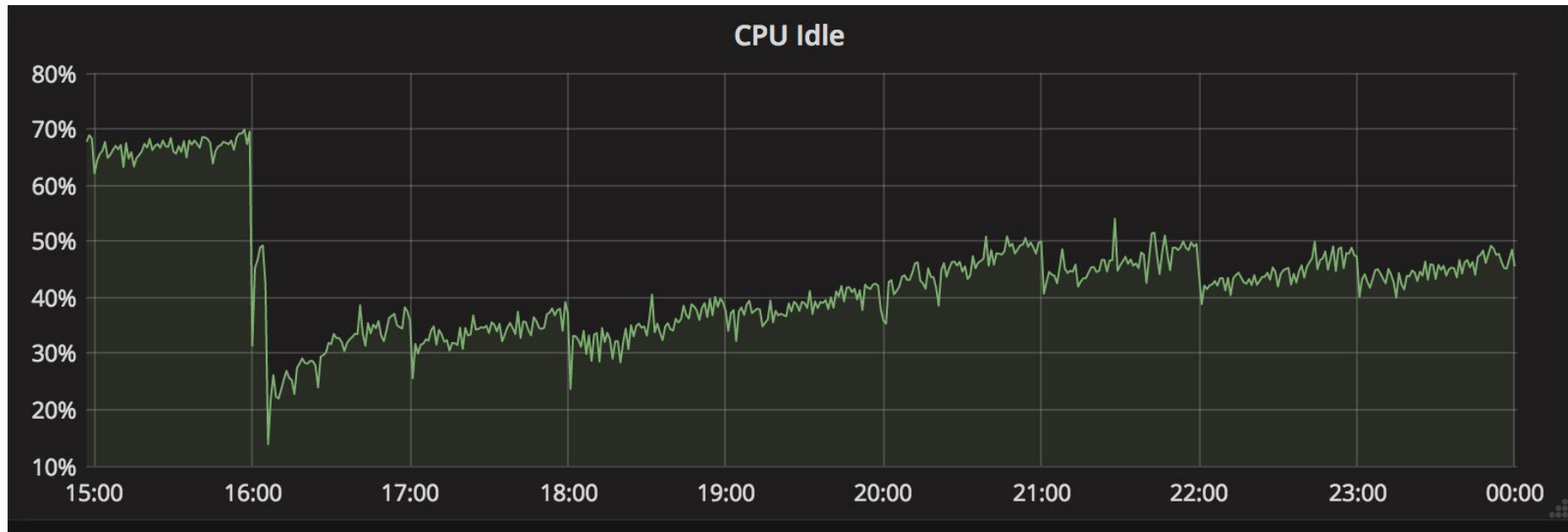
Scale Airflow Executor

It's just Airflow being Airflow

- Why my task is not running
- Airflow deadlock again
- Airflow not scheduling any tasks



Airflow CPU usage



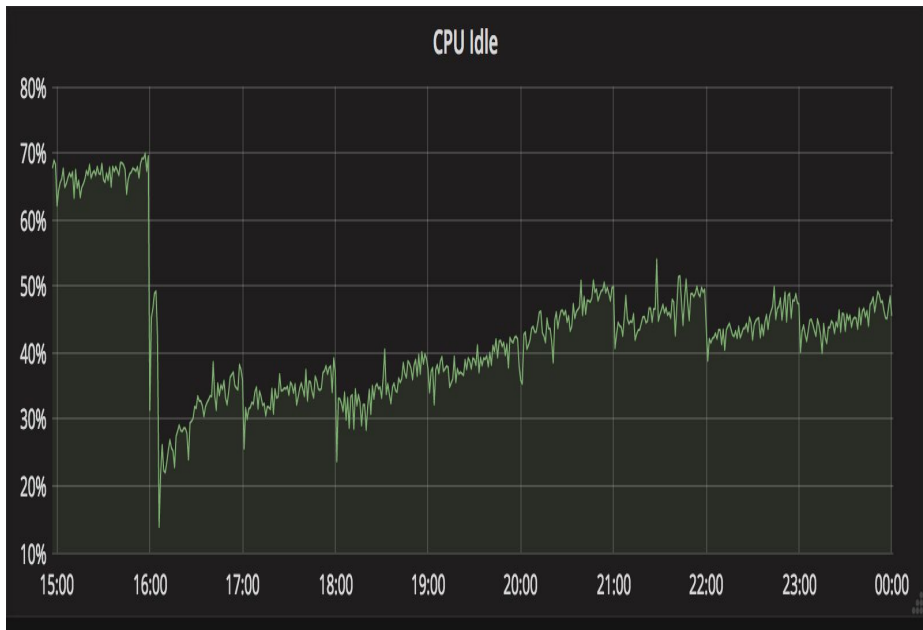
Airflow Multi Retryable Sensors

Airflow local executor launches a new python interpreter per-task, which has been observed to use significant system resources. To minimize the local machine cost of checking external tasks, here we check multiple external tasks in a single task.

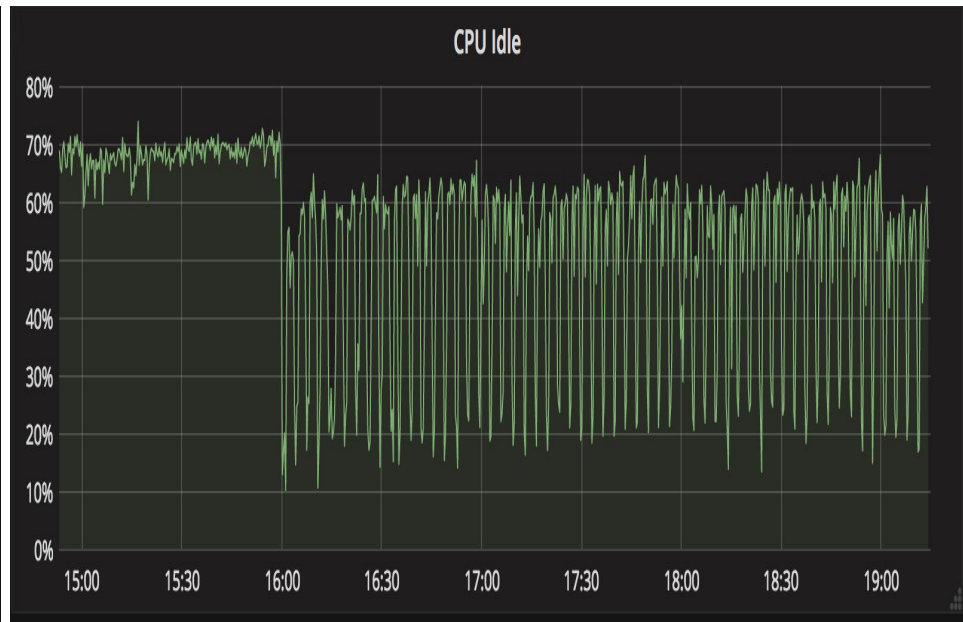
To simplify depending on multiple external tasks, it's recommended using the builder to create new instances:

```
wf = (  
    MultiRetryableExternalTaskSensor.Builder(task_id='my_task', dag=dag)  
        .waitfor('dag1', 'task1', ...)  
        .waitfor('dag2', 'task2', ...)  
        .build()  
)
```

Retryable Sensors CPU usage



Non-Retryable Sensors Load



Retryable Sensors Load



Pipeline Operations



Airflow fallacies

- The upstream task success is reliable.
- The task remain static after the success state.
- The DAG structure is static.
- The data quality not part of a task life cycle.

Mario: Global DAG operator

```
~/w/d/bin >>> ./mario --help
```

```
Usage: mario [OPTIONS] COMMAND1 [ARGS]... [COMMAND2 [ARGS]...]...
```

Options:

```
--help  Show this message and exit.
```

Commands:

clear-downstream	Print Airflow commands to clear all...
dependencies	Export a .graphml representation of airflow...
downstream	Find all tasks (across DAGs) that are...
export_graphml	Export a .graphml representation of airflow...
render	Render Airflow tasks with local params.
upstream	Find all tasks (across DAGs) that are...
waitfor-counts	Prints the number of waitfors that are...
why	Por que?

Airflow operations

*Hive Partition
Sensor Operator*

1. Check task success state
2. Check Hive metastore for partition
3. Check S3 path for the `_SUCCESS` file

DQ Check

```
```\nfrom slack.airflow.data_quality import DQCheck\n\nMyTaskOperator(\n    task_id='my_task',\n    dag=dag,\n    dq_sql=DQCheck('dw.my_table', unique='my_primary_key').is_positive("count(1)", "row_count", over=1).build()\n)\n```\n
```

*DAG cleanup*

**delete\_dag <dag name>**

# DAG Policy Validator

---

*test\_external\_tasks*

Check if external tasks point to valid DAGs and tasks.

*test\_circular\_dependencies*

Check if tasks have circular dependencies *\*across\** DAGs.

*test\_priority\_weight*

Check that production tasks do not depend on a lower priority task.

*test\_on\_failure*

Require that high-priority DAGs have an on-failure alert.

# DAAG Policy Validator

---

*test\_sla*

Require that high-priority DAGs have an SLA.

*test\_sla\_timing*

SLAs timing should make sense. No job should depend on a task that has an equal or longer SLA than it does.

*test\_has\_retry\_and\_success\_callbacks*

Require an on\_success\_callback for tasks with an on\_retry\_callback.

*test\_require\_dq\_for\_prod*

Require SQ check for all the high priority tasks.



# Alerting and Monitoring



# Alerting and Monitoring

---


- Alerting should be reliable
- Alerts should be actionable.
- Alert when it really matters.
- Suppress repeatable alerts.

# OnCall Alert callback

```
from slack.airflow.webhook import Channels, OncallAlertCallback
dag = SlackDAG(
 alerts=OncallAlertCallback(
 channels=Channels.MY_CHANNEL,
 priority='high',
 escalation_chain=['Foo', 'Bar'],
 notes='This task upstream of billing pipeline'
 resources={
 'My runbook': 'https://slack-github.com/path/to/runbook',
 },
 pagerduty_email='data@slack.com',
),
)
```

# OnCall Alert callback

---

 Airflow

DAGs

Data Profiling ▾

Browse ▾

Admin ▾

Docs ▾

About ▾

Data Etl ▾

On Call ▾

DAGs

On Call Dashboard

Production Dashboard

Runbook




# Sample Alerts



**Airflow Alerts** APP 5:00 PM



 SLA miss for DAG `team_company_segment` task  
`team_company_segment_all_done` on 2018-04-04T00:00:00.

## Notes:

This is a core pipeline. No downstream tasks will run until this task succeeds.

**Escalation Chain:** atl, lj (*These people should be contacted only if the on-call Analyst/DE cannot resolve the issue without additional context.*)


High Priority | On-Call Playbook | Sent by airflow on airflow9

# Sample Alerts



## Data Quality Warning APP 2:43 AM



 Task `teams_aux_v2_dq_warn` of DAG `dim_aux` failed a data quality warning on 2018-03-05T00:00:00 (attempt 1 of 4).

### Notes:

This is not a blocking task and failures can happen due to fluctuations in values that are outside normal levels. Investigate this and post an update in a thread, but you don't have to clear the Airflow task.

### Failed Columns:

The following columns failed their data quality checks:

- email\_domain\_check

**Escalation Chain:** @, j, e, s, s, k, i (*These people should be contacted only if the on-call Analyst/DE cannot resolve the issue without additional context.*)


High Priority | On-Call Playbook | Sent by airflow (running with sudo) on airflow9

# Sample Alerts



**Data Quality Failure** APP 10:43 AM



 Task [enterprise\\_stats](#) of DAG [customer\\_stats](#) failed a data quality check on 2018-03-05T00:00:00 (attempt 1 of 4).

## Notes:

This is a core pipeline. No downstream tasks will run until this task succeeds.

## Failed Columns:

The following columns failed their data quality checks:

- row\_count

## Triage Resources:

- [Customer Stats runbook](#)

[Show less](#)

High Priority | On-Call Playbook | Sent by airflow (running with sudo) on airflow9

# A little too quiet

---



**Les Jones** 📱 2:59 PM

Quiet in here.



**Les Jones** 📱 2:59 PM

A little too quiet.





# What is next?



# What is next?

---

- Increase pipeline visibility (user action, task life cycle etc)
- Data Lineage
- Airflow Kubernetes Executors



# Thank You!

