Managing Scalable Data Workflows on HPC Clusters

Erfan Nourbakhsh (UC Davis)

Astroinformatics 2019

June 26th, Caltech

Outline

- Time-based job schedulers
- Smart scheduling and scalability
- Workflow management systems and big data pipelines
- Introducing Apache Airflow
- Airflow components
- UI walk-through
- Airflow and HPC

Time-based job schedulers

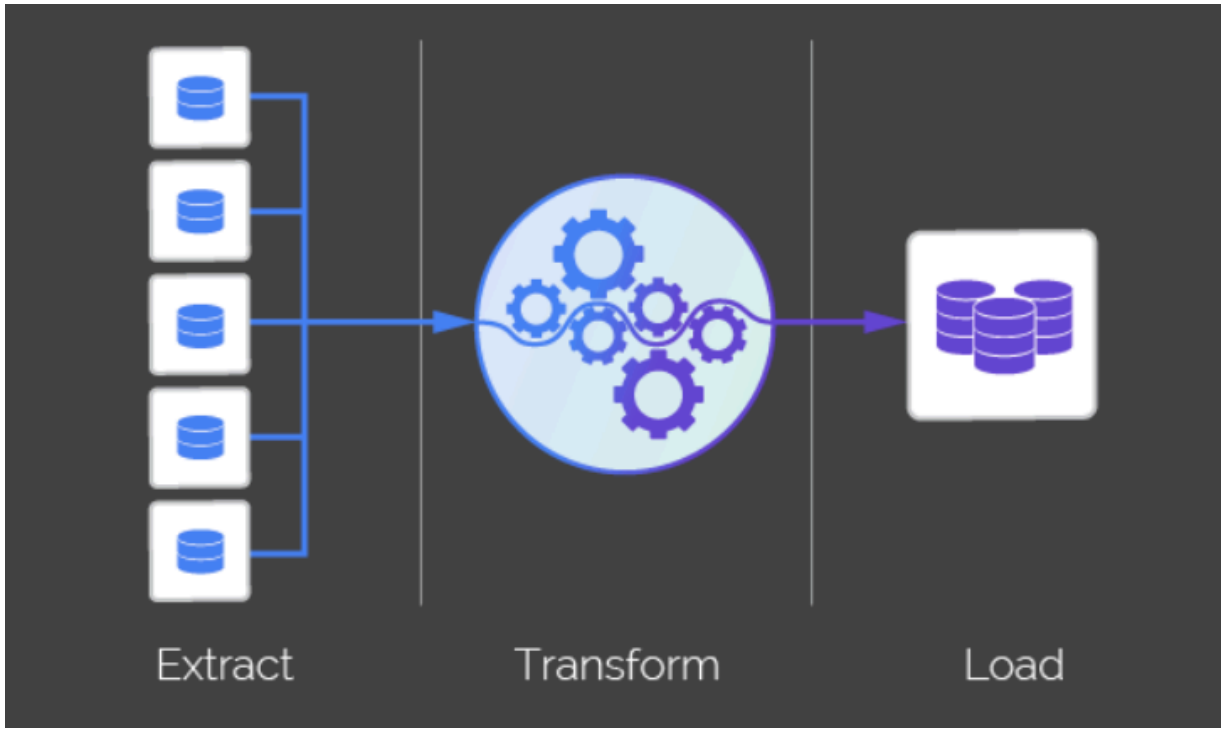
CRON

Smart scheduling and scalability

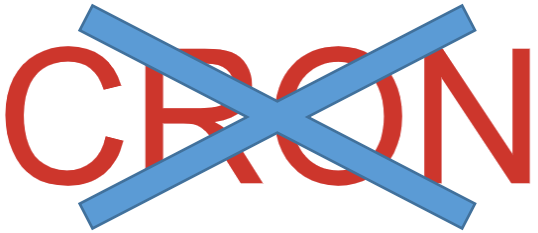
- **Smart Scheduling**
  - Scheduling task execution minimal criterion for a WMS
  - We want task execution to be more “data-aware”

- **Scalability**
  - Not only address our current data processing needs, but also scale with our future growth
  - This should be without requiring substantial engineering time and infrastructure changes

Workflow management systems and big data pipelines



Workflow management systems and big data pipelines



Apache Airflow

- Airflow is a platform to programmatically author, schedule and monitor workflows.
- Developed at Airbnb in 2014
- Joined the Apache Software Foundation's incubation program in 2016
- It has been built to scale
- Python script (configuration as code)
- Active development on GitHub (835 contributors, 6,534 commits)
- Rich web UI
- Officially used by about 300 companies (Tesla, Twitter, Yahoo!, PayPal, United Airlines etc.)

Airflow Components

Airflow

Run your pipelines in style
https://airflow.apache.org/

Directed Acyclic Graph – DAG

The DAG defines the blueprint of your analysis . It is an instructional format to tell airflow which tasks are executed in what order.

Operators

Your task blueprint identifying the top level instructions to execute your task . Is your task a bash command, python function, sql query, spark operation, etc.

Airflow Components

Sensors

Sensors are operators that wait for some event to occur, such as a file on appearing or an http request to occur, before executing a task.

Tasks

The actual execution instructions. Your python function to execute, bash command to run, spark pipeline ,etc.

<sources>

Airflow Docs - https://airflow.apache.org/
Airflow Github - https://github.com/apache/airflow
Some Icons made by "https://www.freepik.com/"

</visit https://dabble-of-devops.com for more information>

time_splitter.py

```python
from airflow import DAG
from airflow.operators import BashOperator
from datetime import datetime, timedelta


default_args = {
    'owner': 'mils',
    'start_date': datetime(2017, 6, 13),
    'retries': 1,
    'retry_delay': timedelta(minutes=1),
}

dag = DAG('Time_splitter',
    schedule_interval='*/30 * * * *',
    default_args=default_args,
    catchup=False)


t1 = BashOperator(
    task_id='task1',
    bash_command='date > /home/ubuntu/airflow/utils/data.txt',
    dag=dag
)

t2 = BashOperator(
    task_id='task_2_split_time',
    bash_command='python /home/ubuntu/airflow/utils/split_into_time.py',
    dag=dag)

t3 = BashOperator(
    task_id='task_3_split_into_mins',
    bash_command='python /home/ubuntu/airflow/utils/split_into_min.py',
    dag=dag)

t1.set_downstream(t2)
t2.set_downstream(t3)
```

Importing modules

Default Arguments

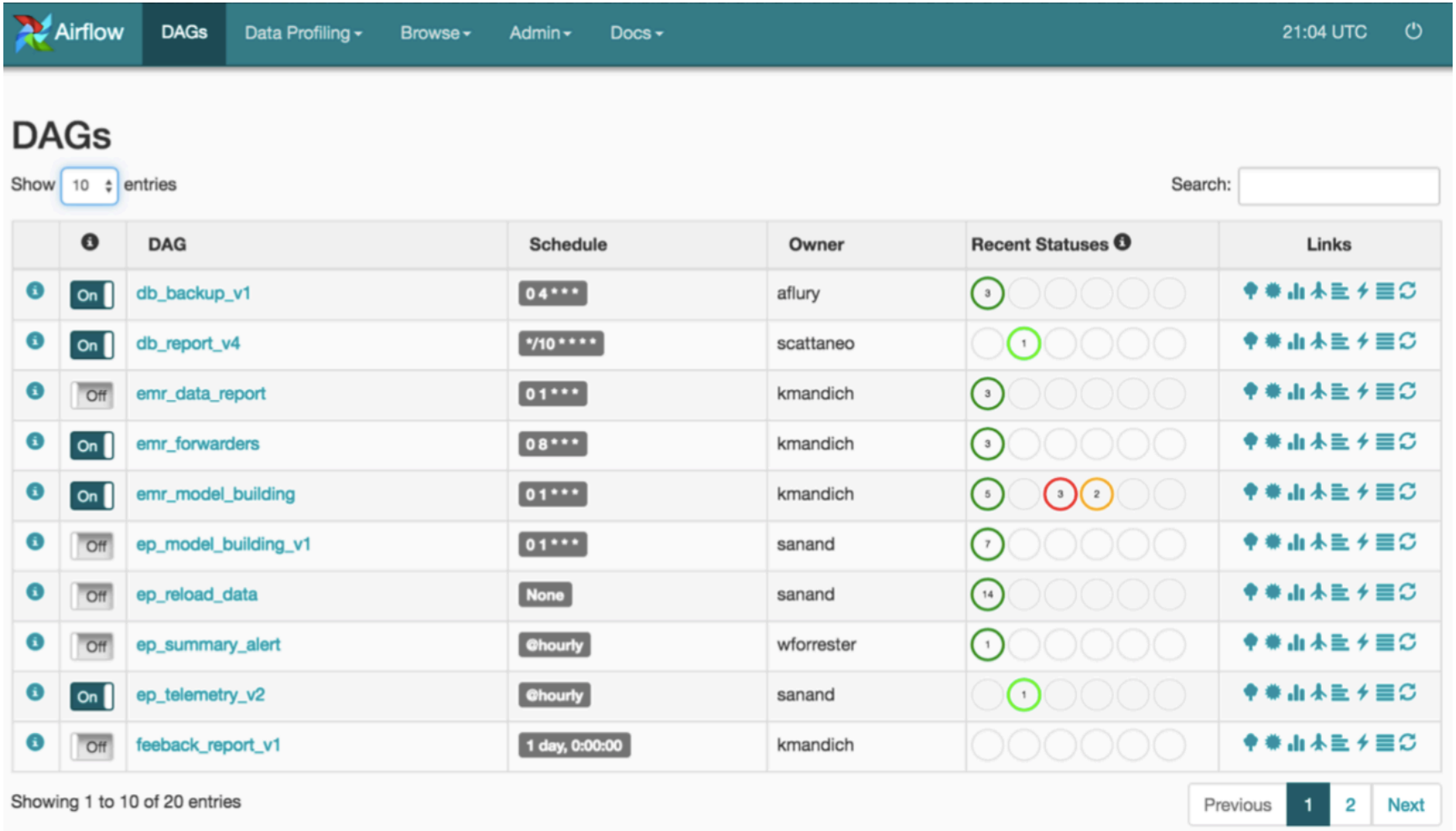
DAG

tasks

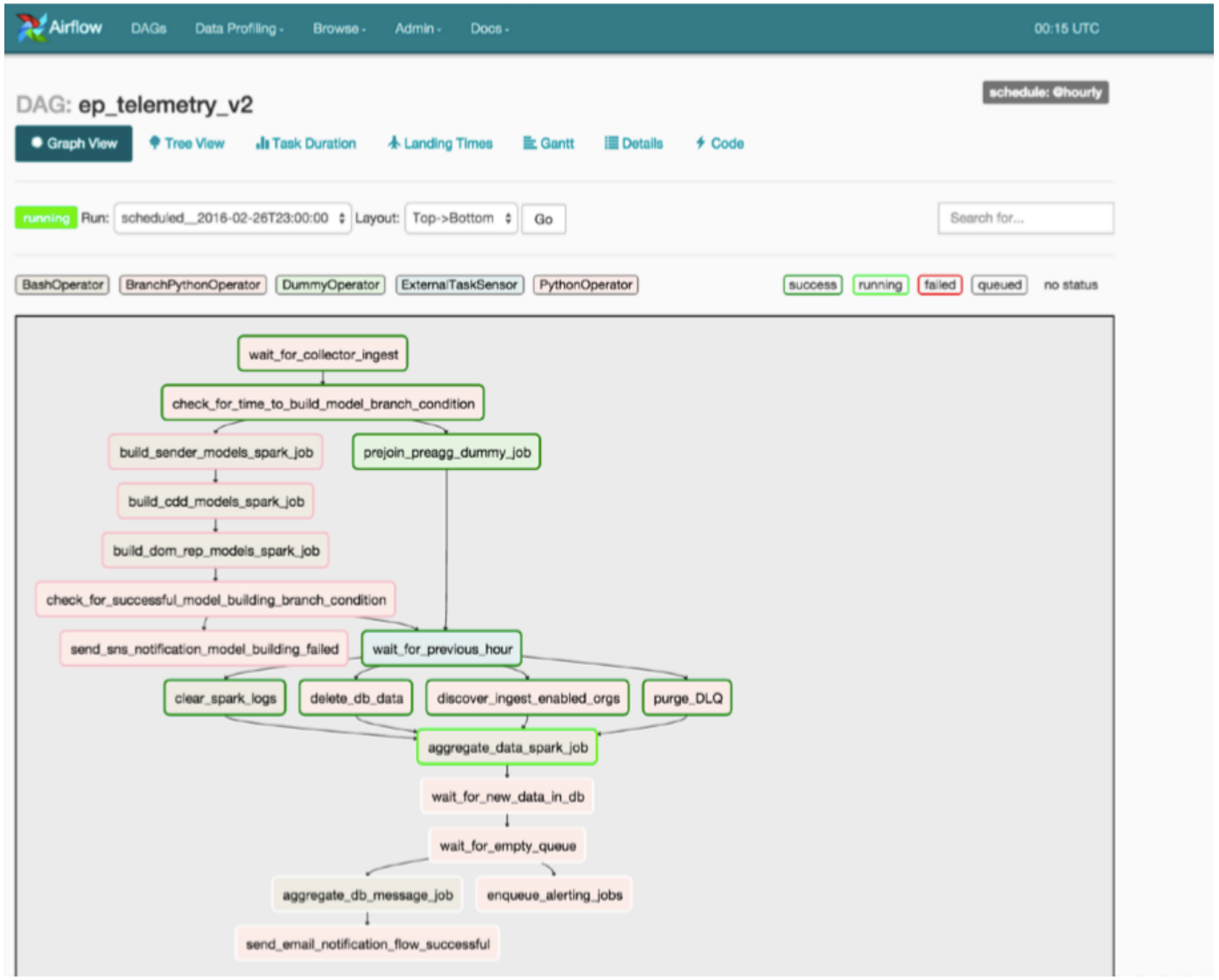
dependencies

DAG File

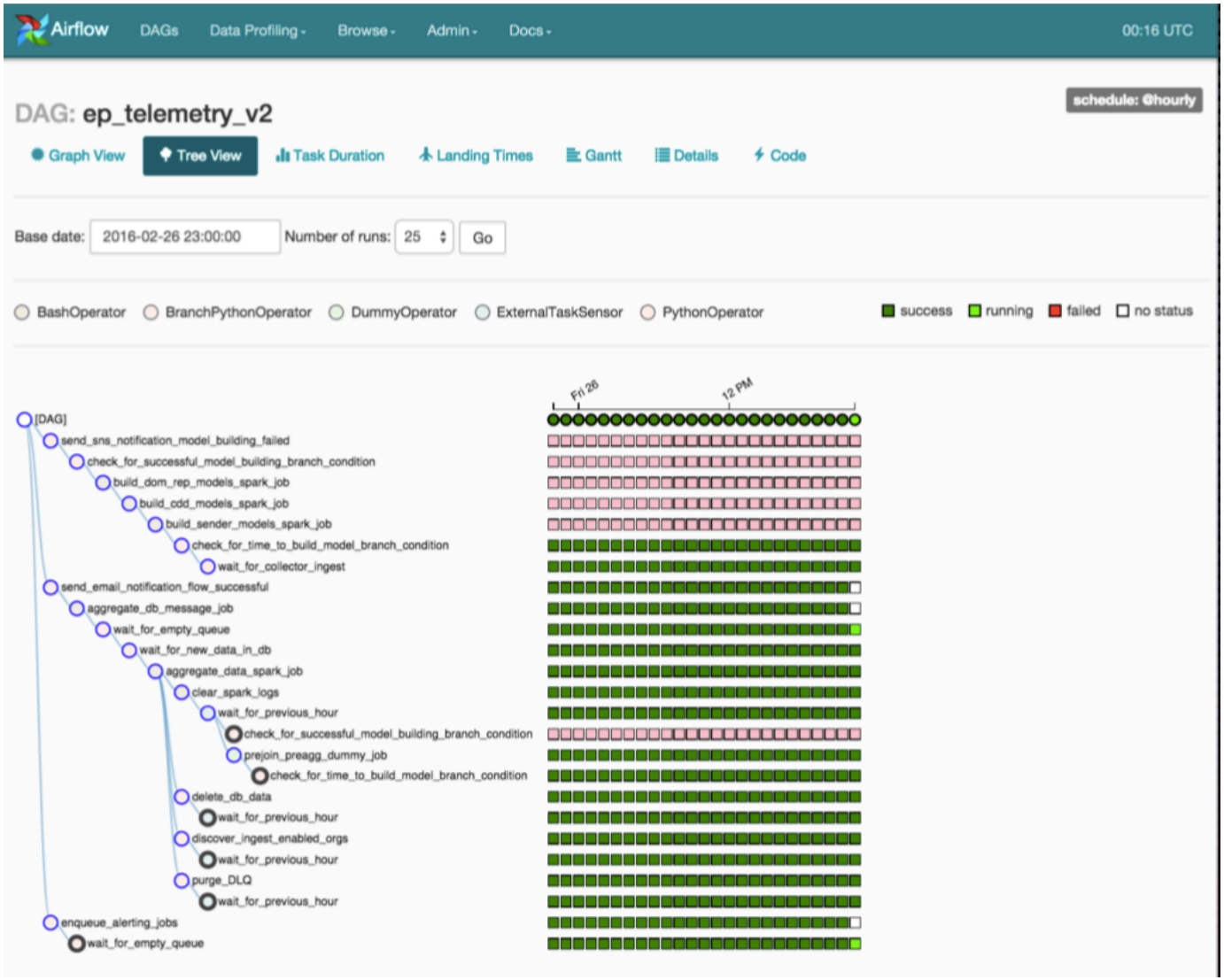
UI Walk-through



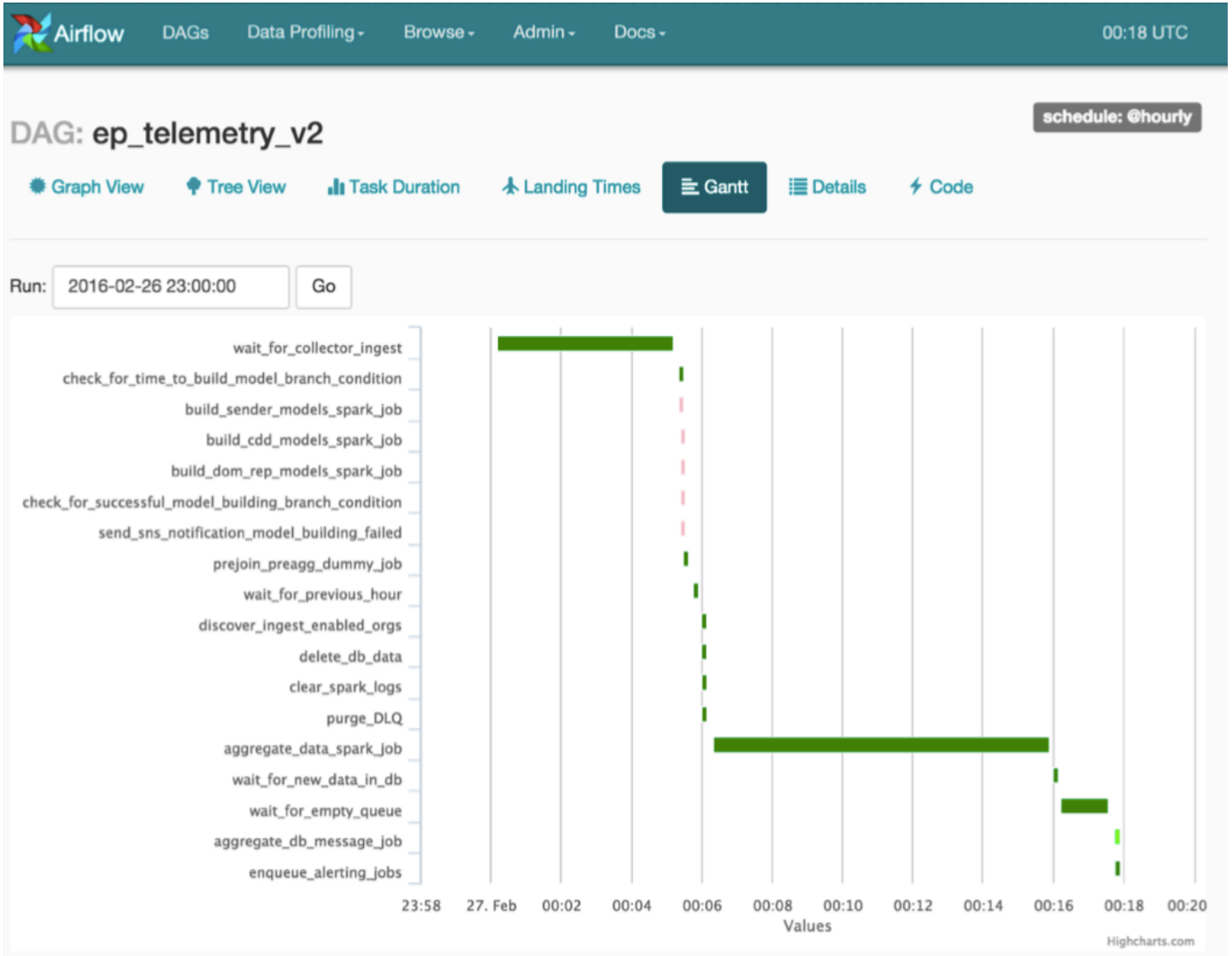
UI Walk-through



UI Walk-through



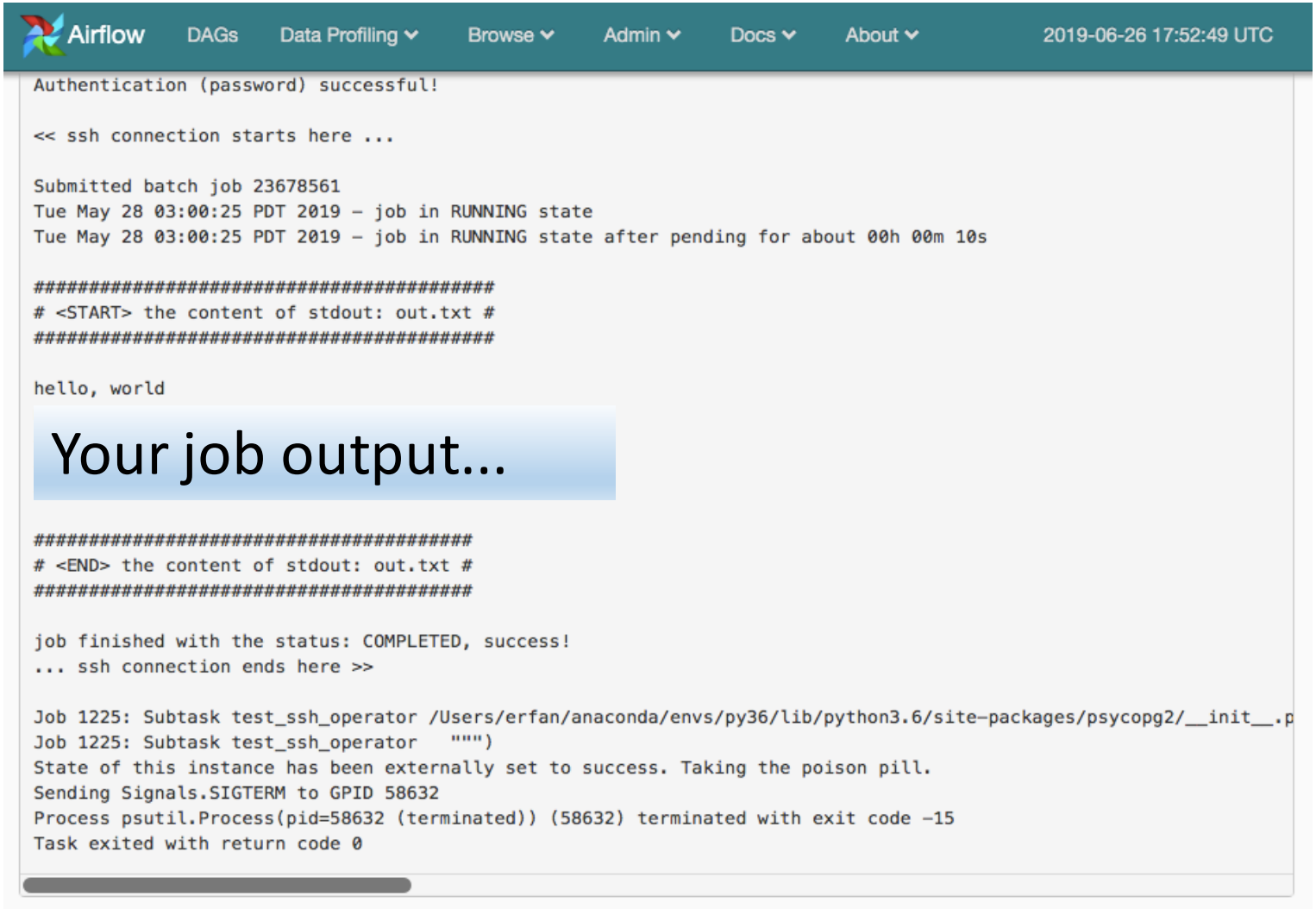
UI Walk-through



Airflow and HPC

- Apache Spark Integration
- MPI with Slurm Integration (not natively supported)

Slurm Output



Conclusion

Try Airflow and apply
it to your work

Thank you!

Sources

https://github.com/apache/airflow

https://www.alooma.com/answers/what-is-apache-airflow

https://sonra.io/2018/01/01/using-apache-airflow-to-build-a-data-pipeline-on-aws/

https://www.slideshare.net/mesodiar/intro-to-airflow-good-bye-cron-welcome-scheduled-workflow-management

https://qcon.ai/system/files/presentation-slides/sid_anand_qcon_ai_2018_v2.pdf

https://dabble-of-devops.com/learn-airflow-by-example-part-1-introduction/

https://towardsdatascience.com/why-quizlet-chose-apache-airflow-for-executing-data-workflows-3f97d40e9571