An On-Demand Airflow Service for Internet Scale Gameplay Pipelines

Yuanmeng Zeng & Nitish Victor
Yuanmeng Zeng
Software Engineer II

EA Digital Platform (EADP) DATA & AI

yzeng@ea.com
EA Digital Platform (EADP)

- EADP Data & AI
  - Data capture
  - Data storage and processing
  - Data science and analysis

- Game Telemetry adhering to a common taxonomy

- Centralized data platform with unified access to the data
About 40 game studios under EA’s umbrella

Ever-changing data landscape with new acquisitions such as Codemasters and Glu Mobile

Terabyte-scale data generated daily and petabyte-scale data access

Thousands of ETL jobs

Different needs in terms of data from different studios
Airflow at EADP

- Airflow as a Service
- Support multiple game analyst teams and EADP ETL team
- DevOps job for monitoring and scaling
- Experimentation job for AI and machine learning
- Data processing job using our compute cluster

Game Analyst Team

ETL Team

Airflow Job

Ops/ Monitoring

Compute Cluster

AI / ML/ Experimentation

Data Lake
Before

Single Airflow Cluster

- Monolithic design
- Multi-tenant shared environment
- Variables and connections managed through single Airflow instance.
Motivation for On-Demand Airflow Service

Lack of Isolation between teams
Need of team-level ACL on sensitive datasets, variables and connections

Implementation is monolithic
Any platform level change impacts all production workloads

Difficult to support diverse use-cases across teams
Teams have very different use cases and ways they use Airflow

Lack of self-serve management capabilities
Teams are unable to manage user access on their own and scale up/down according to their needs

Ever increasing DAG count and workload
High stress on a single scheduler

Variance resource requirement of dags
Workload of different dags has different requirements for hardware resources
On-Demand Airflow Design

Multiple Airflow clusters

- Each “team” or entity gets their own isolated Airflow cluster(s)
- Route to their cluster using subdomains under airflow.data.ea.com
- For example, FIFA team could have a cluster at fifa.airflow.data.ea.com
- Deploy new dags through CI/CD without redeployment of airflow
On-Demand Airflow Design

Airflow using K8s Operator

- Use cloud-native Kubernetes operator to deploy/scale/shutdown clusters
- Customize or create an operator for EADP use-case and expose functionality via an API.
- REST API layer in front of K8s Operator to allow external calls
## Cluster Sizes

<table>
<thead>
<tr>
<th>Cluster Size</th>
<th>XS</th>
<th>S</th>
<th>M</th>
<th>L</th>
<th>XL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Concurrency</td>
<td>4</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>Worker Concurrency</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Worker Count</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>12</td>
<td>16</td>
</tr>
</tbody>
</table>
## Self-Serve Management UI

### Schedulers

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Scheduler Size</th>
<th>Operations</th>
<th>Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>ai</td>
<td>Running</td>
<td>X-Small (1 credit/hour)</td>
<td>Edit</td>
<td>Restart</td>
</tr>
<tr>
<td>a</td>
<td>Running</td>
<td>X-Small (1 credit/hour)</td>
<td>Edit</td>
<td>Restart</td>
</tr>
<tr>
<td>a</td>
<td>Running</td>
<td>X-Small (1 credit/hour)</td>
<td>Edit</td>
<td>Restart</td>
</tr>
<tr>
<td>central-eadp</td>
<td>Running</td>
<td>X-Small (1 credit/hour)</td>
<td>Edit</td>
<td>Restart</td>
</tr>
<tr>
<td>d</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dayofdata</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>devops</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>etl</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>etl-t</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Edit Scheduler: devops**

- **Size**
  - X-Small (1 credit/hour)
  - Medium (4 credits/hour)
  - Large (8 credits/hour)
  - X-Large (16 credits/hour)

- **Config S3 DAG Location**
  - t | Delete | dashboard
## Improvements

<table>
<thead>
<tr>
<th>Fully self-serve</th>
<th>Minimum impact of downtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster flexibility &amp; Diversity</td>
<td>Homogeneous jobs in the same cluster</td>
</tr>
<tr>
<td>Easy troubleshooting</td>
<td>Cost saving</td>
</tr>
</tbody>
</table>
Future work

- AI-based cluster scaling
- Various hardware support
- More granular access control on cluster operation