Characterizing and Synthesizing Task Dependencies of Data-Parallel Jobs in Alibaba Cloud

Huangshi Tian, Yunchuan Zheng, Wei Wang @ HKUST

Nov. 21, 2019
Every job is born equal, but some are more complicated ...

- Hadoop

- Spark

- Ecosystem (MLlib, SQL, GraphX)

Iterative Job

Auto-generated Job
Do job DAGs have anything special?

An (unsatisfactory) answer from the literature:
Production DAGs are *large and complex* ...
A Glimpse into Production Clusters

In the year of 2018, Alibaba has released a trace that ...

- spans **8 days**,  
- records the activity of both **long-running containers** and **batch jobs** ...,  
- from a **cluster of 4034 machines**.
A Glimpse into Production Clusters

In the year of 2018, Alibaba has released a trace that spans 8 days, records the activity of both long-running containers and batch jobs...

- **Terminologies**
  - task
  - instance
  - dependency

- **Dataset Scale**
  - 4.2M jobs
  - 14.3M tasks
  - 1.4B instances

- **Applications**
  - SQL queries (90%)
  - data analytics (10%)

Zoom in on Batch Jobs

Job-A
Overview of DAG Jobs

- Temporal Distribution

| Dataset Scale | 4.2M jobs | 14.3M tasks | 1.4B instances |

- Applications
  - SQL queries (90%)
  - Data analytics (10%)

- Resource Consumption

![Temporal Distribution Graph](image1)

![Resource Consumption Graph](image2)
Overview of DAG Jobs

- Temporal Distribution

![Temporal Distribution Graph]

- Resource Consumption

![Resource Consumption Graph]

**Takeaway:** DAG jobs are prevalent and sometimes consume disproportionately many resources.
First Impression on Job DAGs: Trees Everywhere

- 78.54% of all jobs are **gatter jobs**;
- 36.03% are **scatter jobs**;
- Within **complex jobs**, 81.68% of tasks can be decomposed into scatter or gather jobs.
First Impression on Job DAGs: Trees Everywhere

- 78.54% of all jobs are **gatter jobs**;

- 36.03% are **scatter jobs**;

- Within **complex jobs**, 81.68% of tasks can be decomposed into scatter or gather jobs.

**Takeaway:** There are opportunities for algorithmic scheduling.
Commonality or Peculiarity?

We introduce four datasets of DAGs for comparison:

1. **Alibaba DAGs** extracted from the trace,

2. **Random DAGs** generated by a uniformly random algorithm,

3. **TPC-DS DAGs** from the namesake benchmark,

4. **TPC-H DAGs** similar as above.
Sparsity and Probable Cause

- **Edge density** defined as:

  \[
  \text{edge density} = \frac{\text{# dependencies}}{\text{# possible dependencies}}
  \]

![Box plot showing edge density comparison between Alibaba, Random, TPC-DS, and TPC-H datasets.](image)
Sparsity and Probable Cause

- **Edge density** defined as:
  \[
  \frac{\text{# dependencies}}{\text{# possible dependencies}}
  \]

- **Chain ratio** defined as:
  \[
  \frac{\text{# tasks with only one parent/child}}{\text{# all tasks}}
  \]
Sparsity and Probable Cause

- **Edge density** defined as:
  \[
  \frac{\text{# dependencies}}{\text{# possible dependencies}}
  \]

- **Chain ratio** defined as:
  \[
  \frac{\text{# tasks with only one parent/child}}{\text{# all tasks}}
  \]

**Takeaway**: Job DAGs are sparse and have many chains.
In- and Out-Degrees

- Edge density: defined as 
  \# dependencies / \# possible dependencies
- Chain ratio: defined as 
  \# tasks with only one parent/child / \# all tasks
- Sparsity and Probable Cause

Takeaway: Job DAGs are sparse and have many chains.

In- and Out-Degrees:

- CCDF (1 - CDF) graphs for Alibaba, Random, TPC-DS, and TPC-H.

- Random: 35.9% no less than 3
- Alibaba: 99% no more than 3

- In-Degree graph range: 0 to 200
- Out-Degree graph range: 0 to 100
In- and Out-Degrees

Takeaway: A task can have many dependencies, but typically a few children.
Shape of DAG

- **Maximum Parallelism**
- **Critical Path**
Shape of DAG

The diagrams depict the critical path length and max parallelism for different datasets:
- Alibaba
- Random
- TPC-DS
- TPC-H

The critical path length increases with the number of tasks, while max parallelism shows fluctuations.
Shape of DAG

**Takeaway:** Production DAGs grow "wider" instead of "longer".
Runtime Performance of DAG Jobs

- **Runtime Variability**: troublemaker for cluster schedulers
  - straggler tasks
  - resource fragmentation

- **Measuring Variation**
  - dependent pair: ratio between metrics
  - dependent set: geometric mean of all pairwise ratios
Does Dependency Constrain Runtime Variability?

Vary over 5x

<table>
<thead>
<tr>
<th>Metric</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance #</td>
<td>26.46%</td>
</tr>
<tr>
<td>Duration</td>
<td>20.77%</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>1.89%</td>
</tr>
<tr>
<td>Memory Usage</td>
<td>20.12%</td>
</tr>
</tbody>
</table>

Variation in Alibaba Trace

Geometric Mean of Pairwise Ratios in Dependent Sets
Does Dependency Constrain Runtime Variability?

**Takeaway:** Unfortunately, not that much.
Variability of "Recurrent" Jobs

We select "recurrent" jobs by the criteria of (1) **isomorphic structures**, (2) **periodic submission intervals** and (3) **identical resource requests**.

<table>
<thead>
<tr>
<th>... vary over 2x</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance #</td>
<td>69.25%</td>
</tr>
<tr>
<td>Duration</td>
<td>75.69%</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>54.15%</td>
</tr>
<tr>
<td>Memory Usage</td>
<td>57.61%</td>
</tr>
</tbody>
</table>

Geometric Mean of Pairwise Ratios between Recurring Tasks
Variability of "Recurrent" Jobs

We select "recurrent" jobs by the criteria of (1) **isomorphic structures**, (2) **periodic submission intervals** and (3) **identical resource requests**.

<table>
<thead>
<tr>
<th>... vary over 2x</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance #</td>
<td>69.25%</td>
</tr>
<tr>
<td>Duration</td>
<td>75.69%</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>54.15%</td>
</tr>
<tr>
<td>Memory Usage</td>
<td>57.61%</td>
</tr>
</tbody>
</table>

**Takeaway**: Recurrent tasks can have high variability.
How to Synthesize a DAG

**STEP I:** Randomly draw a critical path length from the distribution.

**STEP II:** Randomly decide how tasks are distributed along the path.

**STEP III:** Randomly connect tasks on adjacent levels.

(Please refer to the paper for the evaluation results.)
Trace Generator

- No need to manipulate 200GB+ of raw data.
- Flexibly control the duration, load and heterogeneity of the trace.
Summary

• Structural Properties of Job DAGs, ...
  ○ sparse
  ○ "bounded" critical path
  ○ increasing parallelism

• Runtime Performance, ...
  ○ salient variability
  ○ ... even among recurrent tasks

• and Trace Generator