Unearthing inter-job dependencies for better scheduling

Andrew Chung*
Subru Krishnan*, Konstantinos Karanasos*, Carlo Curino*, Greg Ganger+

* Carnegie Mellon University
+ Azure Data Gray Systems Lab
Wing summary

Shared cluster

Jobs

Scheduler
Understanding previously-ignored inter-job dependencies is important.
Outline

• Background: Clusters, scheduling, inter-job dependence
• Inter-job dependencies and the problems they bring
• The Wing inter-job dependency profiler
• Cluster resource scheduling with Wing
• Conclusion: Inter-job dependencies are important!
Outline

• Background: Clusters, scheduling, inter-job dependence
• Inter-job dependencies and the problems they bring
• The Wing inter-job dependency profiler
• Cluster resource scheduling with Wing
• Conclusion: Inter-job dependencies are important!
Shared data lakes

Org 1

Org 2

Org 3

Scheduler

Compute

Running jobs

Shared data lake
Shared data lakes

Org 1

Org 2

Org 3

Scheduler

Compute

Running jobs

Shared data lake

Shared data

W_1

W_2

W_3

R_2

R_3

A

B

C

A depends on B depends on C

C depends on B depends on A
Outline

• Background: Clusters, scheduling, inter-job dependence
• Inter-job dependencies and the problems they bring
• The Wing inter-job dependency profiler
• Cluster resource scheduling with Wing
• Conclusion: Inter-job dependencies are important!
Shared data lakes

Org 1

A

Org 2

B

Org 3

C

Scheduler

Compute

A

B

C

Running jobs

Shared data lake

W₁

W₂

W₃

Shared data

Compute

R₂

R₃

Scheduler

Org 1

A

Org 2

B

Org 3

C

Running jobs

Shared data lake

W₁

W₂

W₃

Shared data
Problems when not considering deps

Inter-job dependencies pervade data lakes, but are ignored in resource management
Problems when not considering deps

Inter-job dependencies pervade data lakes, but are ignored in resource management

Missed deadlines, wasted resources, and untapped opportunities
Towards addressing inter-job deps

Wing
Discover + analyzes inter-job dependencies from data provenance

Scheduling with Wing guidance
Scheduling informed with historical inter-job dependencies
Outline

• Background: Clusters, scheduling, inter-job dependence
• Inter-job dependencies and the problems they bring
• The Wing inter-job dependency profiler
• Cluster resource scheduling with Wing
• Conclusion: Inter-job dependencies are important!
Data from a Cosmos cluster

40k+ daily jobs

50k+ servers

Millions of daily tasks

Hundreds of queues (organizations)

TBs of job + data prov logs daily
Data from a Cosmos cluster

- 40k+ daily jobs
- 50k+ servers
- Millions of daily tasks
- Hundreds of queues (organizations)
- 160k+ daily inter-job dependencies
- 95% of queues inter-dependent
- 80% jobs depend on other jobs
- 68% jobs recurring
- TBs of job + data prov logs daily

Carnegie Mellon Parallel Data Laboratory
Microsoft
Scheduling and predicting the future

Better prediction of future jobs

Better planning for future jobs when scheduling

Better results

Recurring dependencies can help improve predictions
Job value & inter-job dependencies

- Failing/finishing jobs late can impact downstream jobs
- Wing analyzes the aggregate value (impact) of jobs
Job value & inter-job dependencies

- Failing/finishing jobs late can impact downstream jobs
- Wing analyzes the aggregate value (impact) of jobs
Job value & inter-job dependencies

- Failing/finishing jobs late can impact downstream jobs
- Wing analyzes the aggregate value (impact) of jobs

Utility function with data-driven deadlines

Carnegie Mellon Parallel Data Laboratory
Microsoft
Outline

• Background: Clusters, scheduling, inter-job dependence
• Inter-job dependencies and the problems they bring
• The Wing inter-job dependency profiler
• Cluster resource scheduling with Wing
• Conclusion: Inter-job dependencies are important!
YARN, Cosmos, and value scheduling

• **YARN**: A resource management framework
  • Back-end of Cosmos resource management
  • Default scheduler: Resource decisions based on priorities

• **Value scheduling**
  • Complete jobs in a timely manner to achieve value
  • State-of-the-art: Considers each job independently

**Inter-job dependencies to achieve more value**
Wing-Agg: Wing-guided scheduling

- Exploit job + dependency recurrence to attain value

- **Wing-Agg**: YARN’s prio-based sched + Wing-guidance
  - Prioritize recurring jobs with high aggregate value efficiency
Wing-Agg

Value

Value(A)

Time

A → B → C → D → E

Carnegie Mellon
Parallel Data Laboratory
Microsoft
Experimental setup

- Trace-driven simulations on real cluster traces
  - Preserves inter-job dependencies and properties
- Goal: Attain more value from the same workload
  - Value metric: Total file output downloads attained
- Experiments at various cluster sizes (capacities)
  - To simulate resource-constrained clusters
Value-attainment

- **Wing-Agg**: Prio as historical \textit{agg} value / \textit{agg} compute

\begin{figure}
\centering
\begin{tikzpicture}
\begin{axis}[
    title={},
    xlabel={$\%$ Cosmos cluster capacity},
    ylabel=Normalized value attained,
    xmin=0, xmax=100,
    ymin=0, ymax=1,
    xtick={20,40,60,80,100},
    ytick={0,0.2,0.4,0.6,0.8,1},
    xticklabels={20\%,40\%,60\%,80\%,100\%},
    yticklabels={0,0.2,0.4,0.6,0.8,1},
]
\end{axis}
\end{tikzpicture}
\end{figure}
Value-attainment

- **Wing-Agg**: Prio as historical `agg` value / `agg` compute

![](image.png)

% Cosmos cluster capacity
Value-attainment

• **Wing-Agg**: Prio as historical `agg` value / `agg` compute

% Cosmos cluster capacity

- Global (20%)
- 100%
- 80%
- 60%
- 40%
- 20%

Normalized value attained

- 1
- 0.8
- 0.6
- 0.4
- 0.2
- 0

- Default YARN
- Wing-Agg

Merge queues to eliminate bottlenecks
Takeaways

• Inter-job dependencies prevalent in real clusters
  • But, can be predictable with recurrence
• Inter-job dependencies need to be addressed
  • To ensure jobs meet their deadlines, reduce resource wastage, and improve value attained in shared clusters

Thank you!