Siphon: Expediting Inter-Datacenter Coflows in Wide-Area Data Analytics

Shuhao Liu, Li Chen, Baochun Li
University of Toronto
July 12, 2018
What is a **Coflow**? 

One stage in a data analytic job

- Map 1 → Reduce 1
- Map 2 → Reduce 2
- Map 3 → Reduce 3
- Map 4 → Reduce 4
What is a **Coflow**?

One stage in a data analytic job

Map tasks

Map 1 → Reduce 1
Map 2 → Reduce 2
Map 3 → Reduce 3
Map 4 → Reduce 4
What is a **Coflow**?

One stage in a data analytic job

Map tasks

Map 1 → Reduce 1
Map 2 → Reduce 2
Map 3 → Reduce 3
Map 4 → Reduce 4

Reduce tasks
What is a **Coflow**?

One stage in a data analytic job

Map 1 \rightarrow \text{Reduce 1}

Map 2 \rightarrow \text{Reduce 2}

Map 3 \rightarrow \text{Reduce 3}

Map 4 \rightarrow \text{Reduce 4}

all-to-all shuffle
What is a **Coflow**?

One stage in a data analytic job

Coflow: considered done only when all flows finish
Coflow Scheduling

- Objective: minimizing average coflow completion time
- Network model: datacenter networking
  - Big switch abstraction
    - network core is congestion-free
Coflow Scheduling

- Objective: minimizing average coflow completion time
- Network model: datacenter networking
  - Big switch abstraction
    - network core is congestion-free
Coflow Scheduling

- Objective: minimizing average coflow completion time
- Network model: datacenter networking
  - Big switch abstraction
    - network core is congestion-free
Coflow Scheduling

- Objective: minimizing average coflow completion time
- Network model: datacenter networking
  - Big switch abstraction
    - network core is congestion-free
Coflow Scheduling

- Objective: minimizing average coflow completion time
- Network model: datacenter networking
  - Big switch abstraction
    - network core is congestion-free
Coflow Scheduling

- Objective: minimizing average coflow completion time
Coflow Scheduling

- Objective: minimizing average coflow completion time
Coflow Scheduling

- Objective: minimizing average coflow completion time

![Diagram of coflow scheduling with non-blocking switch and jobs-coflows connections]
Wide-Area Data Analytics
Wide-Area Data Analytics

Datacenter 1
- Map 1
- Reduce 1
- Data 1

Datacenter 2
- Data 2

Datacenter 3
- Data 3

Datacenter 4
- Data 4

Wide Area Network
Wide-Area Data Analytics
With tasks placed in different datacenter, what about their generated **inter-datacenter coflows**?
Challenges

- Dumb bell network model: inter-datacenter links are the only bottleneck
Challenges

- Constantly **changing** available bandwidth

![Graph showing measured bandwidth (Mbps) in a 100s interval with data points for CA-EU and US-EU connections.](image-url)
Can existing heuristics work?

Estimated Flow Completion Time

Link 1

Link 2
Coflow scheduling should consider the distribution of available bandwidth.
Monte Carlo Simulation
Monte Carlo Simulation

Scheduling Decision Tree
Monte Carlo Simulation

Scheduling Decision Tree

A [0/0]  B [0/0]  C [0/0]
Monte Carlo Simulation

Scheduling Decision Tree

A [0/0]  B [0/0]  C [0/0]

B  C  A  C  A  B
Monte Carlo Simulation

Scheduling Decision Tree

A [0/0]  B [0/0]  C [0/0]

B  C  A  C  A  B
16.2 9.3 12.5 15.5 20.2 13.1
Monte Carlo Simulation

Scheduling Decision Tree

A [0/0]  B [0/0]  C [0/0]

B  C  A  C  A  B

16.2 9.3 12.5 15.5 20.2 13.1
Monte Carlo Simulation

Scheduling Decision Tree

- A [1/1]
  - B
    - 16.2
  - C
    - 9.3
- B [0/1]
  - A
    - 12.5
  - C
    - 15.5
- C [0/1]
  - A
    - 20.2
  - B
    - 13.1
Monte Carlo Simulation

Scheduling Decision Tree

Monte Carlo Simulation

Scheduling Decision Tree


B  C  A  C  A  B

Complexity? 100 * O(n!)
Reduced Simulation Complexity
Reduced Simulation Complexity

\[ \Theta(t \times n^d) \]
Reduced Simulation Complexity

Bounded Search Depth

$\Theta(t \times n^d)$

Reduced Search Breath
(Early termination)
Reduced Simulation Complexity

- Bounded Search Depth
  \[ \Theta(t \times n^d) \]

- Reduced Search Breath (Early termination)
  \[ O(t \times n^d) \]
Reduced Simulation Complexity

- Bounded Search Depth: $\Theta(t \times n^d)$
- Reduced Search Breath (Early termination): $O(t \times n^d)$
- Online Incremental Search
Reduced Simulation Complexity

<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded Search Depth</td>
<td>( \Theta(t \times n^d) )</td>
</tr>
<tr>
<td>Reduced Search Breath (Early termination)</td>
<td>( O(t \times n^d) )</td>
</tr>
<tr>
<td>Online Incremental Search</td>
<td>( O(t \times n^{d-1}) )</td>
</tr>
</tbody>
</table>
How to enforce the scheduling decisions?
Siphon: System Overview

- Form a software-defined overlay network
  - Aggregators: measure bandwidth, schedule coflows based on priority assignment
  - Controller: compute priority based on Monte Carlo simulation
Performance: Coflow Scheduling

Available Bandwidth (Mbps)

Normalized CCT (%)

<table>
<thead>
<tr>
<th></th>
<th>&lt;25%</th>
<th>25-49%</th>
<th>50-74%</th>
<th>≥75%</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-V</td>
<td>95.7</td>
<td>98.7</td>
<td>98.3</td>
<td>89.6</td>
<td>89.7</td>
</tr>
<tr>
<td>T-M</td>
<td>94.9</td>
<td>99.3</td>
<td>96.7</td>
<td>85.9</td>
<td></td>
</tr>
<tr>
<td>V-T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V-M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M-T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M-V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
With Siphon, we can do more...
Intra-Coflow Scheduling

- Largest Flow Group First
Coflow Multipath Routing

No text content available for transcription.
Performance: Intra-Coflow Scheduling

![Graph showing performance comparison between Spark, Naive, Multipath, and Siphon]

- **Task Execution**
  - Spark: 72.2s
  - Naive: 56.3s
  - Multipath: 49.4s
  - Siphon: 48.7s

- **Shuffle Read**
  - Spark: 186.2s
  - Naive: 155.3s
  - Multipath: 130.3s
  - Siphon: 130.3s
Performance: Benchmark Workloads

![Graph showing application run times for ALS, PCA, BMM, Pearson, W2V, and FG, comparing Siphon and Spark performance.]
Takeaway

- Siphon is a software-defined inter-datacenter overlay that realizes:
  - Coflow scheduling in wide-area data analytics: Monte Carlo Simulation
  - Intra-coflow scheduling: Largest Flow Group First
  - Coflow multipath rerouting
- Shorter coflow completion time leads to better job-level performance
Thank you!