Flowtune

Jonathan Perry
Joint work with Hari Balakrishnan and Devavrat Shah.
Flowtune is..

Allocate network resources
  • Quickly
  • Explicitly (maximize utility)
  • Flexibly (in software)
Traditional approach is packet-centric

Switch Algorithms

Server Algorithms

Implicit Allocation

Several RTT to converge

Changes many components
Flowtune’s approach

1. **Flowlet control**
   
   Allocation changes *only* when:
   
   - Flowlets arrive
   - Flowlets terminate

2. Logically centralized
   
   - Reduce RTT dependence
Example

A → Allocator

Allocator

Allocator → A

“Hadoop on A has data for B”

Assign rates

“Send at 10Gbps”
Example

C → Allocator

Allocator

Allocator → A

“Send at 1Gbps”

Allocator → C

“Send at 9Gbps”

“ads_update on C has data for B”

Assign rates
Why is this hard?

Need to choose rates given active flowlets

1. Updates cascade!
2. What is the goal? To act like TCP?
NUM Iterative Optimizer

1. Each link $\ell$ chooses price $p_\ell$

$$\sum_{s \in S(\ell)} x_s - c_\ell$$

2. Each flow $s$ chooses rate $x_s$

3. Goto 1

---

Kelly et al., Journal of the Operational Research Society, 1998
How to reduce latency?

Solution 1:
Update inputs → Run 100 iterations → Output rates

But: too slow!

Solution 2:
Update inputs → Run 1 iteration → Output rates

But: links are over-allocated!

Solution 3:
Update inputs → Run 1 iteration → Normalize rates → Output rates
Flowtune normalizes rates between iterations

• For each flow:
  • Find link $\ell$ on path with largest $r_\ell = \frac{\sum \text{flow rates}}{\text{link capacity}}$
  • Normalize: $x_s \leftarrow x_s / r_\ell$
Architecture

Optimizer

Normalizer

Allocator

Endpoints

rates

flowlet start/end

normalized rates
Flowtune converges quickly to fair allocation

1. Every 10 milliseconds add sender, up to 5 senders
2. Every 10 milliseconds remove sender
Flowtune scales to 49K flows on 64 cores

<table>
<thead>
<tr>
<th>Cores</th>
<th>Nodes</th>
<th>Flows</th>
<th>Cycles</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>384</td>
<td>3072</td>
<td>19896.6</td>
<td>8.29 µs</td>
</tr>
<tr>
<td>16</td>
<td>768</td>
<td>6144</td>
<td>21267.8</td>
<td>8.86 µs</td>
</tr>
<tr>
<td>64</td>
<td>1536</td>
<td>12288</td>
<td>30317.6</td>
<td>12.63 µs</td>
</tr>
<tr>
<td>64</td>
<td>1536</td>
<td>24576</td>
<td>33576.2</td>
<td>13.99 µs</td>
</tr>
<tr>
<td>64</td>
<td>1536</td>
<td>49152</td>
<td>40628.5</td>
<td>16.93 µs</td>
</tr>
<tr>
<td>64</td>
<td>3072</td>
<td>49152</td>
<td>57035.9</td>
<td>23.76 µs</td>
</tr>
<tr>
<td>64</td>
<td>4608</td>
<td>49152</td>
<td>73703.2</td>
<td>30.71 µs</td>
</tr>
</tbody>
</table>
Flowtune

Allocate network resources

• Quickly (centralized)
• Explicitly (maximize utility)
• Flexibly (in software)