Networking & Scheduling

Quick Poll

Topics people are interested in next

- Scheduling low latency jobs (web requests etc.)?
- Scheduling machine learning jobs?
  - Collective communication
  - Other types of jobs (RL), etc.
- Scheduling for servers vs at the edge
- Other topics?
Networks. What is the same & what changes

- Still about allocating capacity
  - Who goes next

- Goals are still similar
  - Fairness / minimize JCT / maximize tput /...
  - Pareto efficiency / minimize cost
  - ...

- The thing being allocated is a bit strange
  - path?
- Task ↔ Packet
  - Non-preemptive: cannot send ½ a packet
- Generally want reliability
  - Retx packet if dropped
- Minimize cost ⇒ Avoiding Retx
  - Coordination?

**Problem:**
How much buffering is enough?
Worse case \( kn \) pkts arrive per second
\( n \) pkts leave

Consequence: Loss is inevitable in the worse case

How this is dealt with:
congestion control at end host
[infrequent coordination]

Roughly:
Reduce sending rate when packets are lost

Why?
(and probe to see if rate should be increased otherwise)

Pros:
Mostly just works!
Do not need information about who else is using the network, how much data they have to send, etc.
Cons: Adjustment is slow

Time to use full capacity?

Especially for 'Short flows'

Q: How to detect packet loss?
WHERE WE ARE

- **Performance**: Send packets at the right rate
- **Pareto Efficiency**: Path has links shared with others

⇒ The 'right rate' depends on
  - Utility function (latency vs. throughput)
  - Behavior of others

- **Scalability/Management Ease**: Do not coordinate with other senders

Might not be good enough for some uses

RELAXING CONSTRAINTS

- **Pareto Efficiency** & no coordination

Moving towards coordination/approaches to cheap coordination
Concerns around coordination:

1. Do not know (a-priori) when a host has data to send
2. Do not know (a-priori) how much data a host wants to send
3. Coordination can require communication

Some approaches:

TDMA (Time Division Multiplexed Access)

Trade-offs?
- Some flows/pkt are still delayed
  - No data \implies Empty net
  - Scaling? \implies Not parralel efficient

  Delay scales with \# of servers

  \[ + \text{Simple & clean} \]
  \[ + \text{Less CC} \]

\textsc{Centralized Coordination (Fastpass NSDI'17)}

\textbf{Trade-Offs}

\[ + \text{Try all schedules/ find best} \]
\[ - \text{Overhead from latency} \]
Silo

VMS Promise Max Utilization

Burst allowance ← VMS

\[ \text{Output lab, burst}\]

VM placement

\[ \text{Space Partition}\]

Trade-offs
VARYS

- Delay Traffic to make space for others
WHAT DATA CENTER NETWORKS LOOK LIKE

"FULL BISECTION BANDWIDTH"

'Fat-Tree' Topology / Folded Clos Network

Over Subscription

Top-Of-Rack Switch (ToR)