AP 13: Ausnutzung von Software-Defined Networking zur Datenverarbeitung

Niklas Semmler, Robert Krösche, Damien Foucard, Prof. Anja Feldmann, PhD
MOTIVATION

Big Data Applications are expected to adhere to static runtimes, independently of the data volume.
MOTIVATION

The Cloud offers flexible allocation of computing units in the form of containers.
PROBLEM

• Network is shared by all clients
• Location of containers is decided by Cloud operator
• Different applications might run on the same machine

Available Bandwidth is unpredictable!
SOLUTION

Reserve bandwidth for high-priority applications.

I need to transfer data from A to B!

I estimate possible bandwidth.

I enforce the bandwidth!
SOLUTION

We implemented bandwidth guarantees using software-defined networking.
BANDWIDTH GUARANTEES

Incoming Traffic → OF Table Match Flows → Meter Table Match Tag → Queue Priority, Queue Best-Effort → Outgoing Traffic

 Src to Dest → Bandwidth → Meter traffic add tag for PRIO or BE → OpenFlow
**BANDWIDTH GUARANTEES**

BDA Flow, A → B
7 Mbit/s

CT Flow1, C → B
9.5 Mbit/s

Incoming Traffic

**OF Table Match Flows**

**Meter Table**
Meter traffic add tag for PRIO or BE

**OF Table Match Tag**

**Queue Priority**
**Queue Best-Effort**

Outgoing Traffic

BDA – Big Data Application
CT – Cross Traffic
BANDWIDTH GUARANTEES

**BDA Flow, A → B**
- 7 Mbit/s

**CT Flow1, C → B**
- 9.5 Mbit/s

**A → B**
- 6 Mbit/s

**Incoming Traffic**

**OF Table Match Flows**

**Meter Table**
- Meter traffic add tag for PRIO or BE

**OF Table Match Tag**

**Queue Priority**

**Queue Best-Effort**

**Outgoing Traffic**

**BDA – Big Data Application**

**CT – Cross Traffic**
BANDWIDTH GUARANTEES

BDA Flow, A → B
7 Mbit/s

CT Flow1, C → B
9.5 Mbit/s

OF Table
Match Flows

A → B
6 Mbit/s

Meter Table
Meter traffic
add tag for
PRIO or BE

OF Table
Match Tag

Queue
Priority

Queue
Best-Effort

Outgoing Traffic

BDA – Big Data Application
CT – Cross Traffic

BDA Flow, A → B
PRIO, 6 Mbit/s

BE

BDA Flow, A → B
1 Mbit/s

CT Flow1, C → B
9.5 Mbit/s

BE

BDA Flow, A → B
7 (=6+1) Mbit/s

CT Flow1, C → B
3 Mbit/s
NETWORK SETUP

Data Analytics
Cross-Traffic
Scheduler

Job Manager
Scheduler

Mapper

Receiver
Reducer

Reducer

Sender

Mapper

OpenFlow
EXPERIMENT SETUP

**Testbed**
- Mininet Virtual Machine

**Switch**
- OpenFlow 1.3 Software Switch
- Links have 10 Mbit/s Bandwidth

**Cross-Traffic**
- 9.5 Mb/s UDP static traffic (iperf)

**Condition**
- 6 Mbit/s Guarantee or None
EXPERIMENT SETUP

Workloads

1. wordcount: Compute word frequency
   - Read text
   - Extract words
   - Group by word
   - Sum counts
   - Write

2. k-Means: Estimate cluster centers for points
   - Read points
   - Closest center?
   - Group by center
   - Combine points
   - Create new centers

Iterate
RESULTS

- 807 MB Copies of Hamlet
- 16 GB German Wikipedia
- 13 MB Points

**Median improvement**

- **wc-hamlet**: 1.7
- **wc-dewiki**: 4.7
- **kmeans**: 3.5
Improvement through guarantees depends on
- Output-to-input ratio of Operators
  - k-means: The same points are used in the iterations → ratio equal to 1
  - wordcount: Words are grouped. Ratio depends on data redundancy.
    - Hamlet: Multiple versions of the same piece → high redundancy, lower ratio
    - Wikipedia: XML File with tags → low redundancy, ratio closer to 1
- CPU-heaviness
  - k-means requires more computation than wordcount
CONCLUSIONS & FUTURE WORK

Mechanism for Guarantees work.

How can the scheduling be extended to larger networks (more switches) and more applications?

What happens when the network is spread out over more than a single data center?