Mahout: Low-Overhead Datacenter Traffic Management using End-Host-Based Elephant Detection

Vasileios Dimitrakis
Datacenters have enormous demands for bandwidth.

Large fraction of data center traffic is carried in small number of flows.

Management of large flows could lead to better utilization of datacenter fabric.
Introduction - Motivation (1)

- Datacenters have enormous demands for bandwidth
Introduction - Motivation (1)

- Datacenters have enormous demands for bandwidth
- Large fraction of data center traffic is carried in small number of flows
Introduction - Motivation (1)

- Datacenters have enormous demands for bandwidth
- Large fraction of data center traffic is carried in small number of flows

Management of large flows could lead to better utilization of datacenter fabric
Introduction - Motivation (2)

Current elephant detection methods suffer from limitations
Current elephant detection methods suffer from limitations

Solution-Mahout: A traffic management method for Elephant Flows
Current Approaches
Current Approaches

- Applications identify their flows as elephant
Current Approaches

- Applications identify their flows as elephant
- Maintain per-flow statistics (Hedera Approach)
Current Approaches

- Applications identify their flows as elephant
- Maintain per-flow statistics (Hedera Approach)
- Sampling Method
Proposed Solution: Mahout

• A shim layer on each end host monitors flows
• It detects elephant flows and marks their packets
• Switches forward marked packets to controller
• Mahout controller computes the best path only for the elephant flows
Proposed Solution: Mahout

- A shim layer on each end host monitors flows
Proposed Solution: Mahout

- A shim layer on each end host monitors flows
- It detects elephant flows and marks their packets
Proposed Solution: Mahout

- A shim layer on each end host monitors flows
- It detects elephant flows and marks their packets
- Switches forward marked packets to controller
Proposed Solution: Mahout

- A shim layer on each end host monitors flows
- It detects elephant flows and marks their packets
- Switches forward marked packets to controller
- Mahout controller computes the best path only for the elephant flows
Detection of elephant flows in end host: Mahout approach

Advantages of elephant flow detection in end-hosts:
Detection of elephant flows in end host: Mahout approach

Advantages of elephant flow detection in end-hosts:

• End host OS has better visibility into the applications’ behavior
Detection of elephant flows in end host: Mahout approach

Advantages of elephant flow detection in end-hosts:

• End host OS has better visibility into the applications’ behavior
• Feasible deployment on end-hosts
Detection of elephant flows in end host: Mahout approach

Advantages of elephant flow detection in end-hosts:

- End host OS has better visibility into the applications’ behavior
- Feasible deployment on end-hosts
- Very low overhead on commodity servers
Mahout Architecture
In-band Signaling

When an elephant flow is detected, controller is informed!
In-band Signaling

When an elephant flow is detected, controller is informed!

The packets are marked using the **Differentiated Services (DS)** field
Mahout Controller

- The controller computes the best path for the packet marked as elephant
Mahout Controller

- The controller computes the best path for the packet marked as elephant

An example flow table setup at a switch by Mahout controller:
Analytical Evaluation

Hedera
Analytical Evaluation

Hedera

- Table entries need to be maintained for all flows

Sampling

- Adds low overhead
- However, things are getting really bad, when network utilization increases

Mahout

- Statistics only gathered for elephant flows
- Significantly lower number of controllers needed compared to Hedera
Analytical Evaluation

Hedera

- Table entries need to be maintained for all flows
- No OpenFlow switch can support this high number of flows
Analytical Evaluation

Hedera

- Table entries need to be maintained for all flows
- No OpenFlow switch can support this high number of flows

Sampling
Analytical Evaluation

Hedera

- Table entries need to be maintained for all flows
- No OpenFlow switch can support this high number of flows

Sampling

- Adds low overhead
Analytical Evaluation

Hedera

- Table entries need to be maintained for all flows
- No OpenFlow switch can support this high number of flows

Sampling

- Adds low overhead
- However, things are getting really bad, when network utilization increases
Analytical Evaluation

Hedera

- Table entries need to be maintained for all flows
- No OpenFlow switch can support this high number of flows

Sampling

- Adds low overhead
- However, things are getting really bad, when network utilization increases

Mahout
Analytical Evaluation

Hedera

- Table entries need to be maintained for all flows
- No OpenFlow switch can support this high number of flows

Sampling

- Adds low overhead
- However, things are getting really bad, when network utilization increases

Mahout

- Statistics only gathered for elephant flows
Analytical Evaluation

Hedera

- Table entries need to be maintained for all flows
- No OpenFlow switch can support this high number of flows

Sampling

- Adds low overhead
- However, things are getting really bad, when network utilization increases

Mahout

- Statistics only gathered for elephant flows
- Significantly lower number of controllers needed compared to Hedera
Mahout’s detection time of elephant flows is significantly lower than the one of Hedera!
Experimental Results (2)

Mahout detects elephant flows \textbf{3 times} sooner than in-network schemes do
Experimental Results (3)
Strong aspects of Mahout

• Controller handles less flows
• Reduces the in-switch resource requirements
• Significant throughput improvement compared to static load balancing techniques
• Sooner detection of elephant flows
Strong aspects of Mahout

- Controller handles less flows
Strong aspects of Mahout

- Controller handles less flows
- Reduces the in-switch resource requirements
Strong aspects of Mahout

- Controller handles less flows
- Reduces the in-switch resource requirements
- Significant throughput improvement compared to static load balancing techniques
Strong aspects of Mahout

- Controller handles less flows
- Reduces the in-switch resource requirements
- Significant throughput improvement compared to static load balancing techniques
- Sooner detection of elephant flows
Weak aspects of Mahout

• DSCP bits may be needed for other uses in some datacenters
• Mahout shim layer needs to be deployed in every virtual machine in virtualized datacenters
• No specific way to propose certain thresholds for the detection of elephant flows
Weak aspects of Mahout

- DSCP bits may be needed for other uses in some datacenters
Weak aspects of Mahout

- DSCP bits may be needed for other uses in some datacenters
- Mahout shim layer needs to be deployed in every virtual machine in virtualized datacenters
Weak aspects of Mahout

- DSCP bits may be needed for other uses in some datacenters
- Mahout shim layer needs to be deployed in every virtual machine in virtualized datacenters
- No specific way to propose certain thresholds for the detection of elephant flows
Conclusion

**Mahout** is a low overhead yet effective traffic management system
Conclusion

**Mahout** is a low overhead yet effective traffic management system

- Manages elephant flows based on an end-host detection scheme
Conclusion

Mahout is a low overhead yet effective traffic management system

- Manages elephant flows based on an end-host detection scheme
- Experimental results showed the feasibility of its deployment
Thank you very much for your attention!
References
