Universität Paderborn — Fakultät EIM
Fachgebiet Rechnernetze
http://www.upb.de/cs/cn/

BACHELOR / MASTER THESIS

Application-Aware Networking using SDN

Background

Spark [5] has become defacto computing framework for its fast in-memory big data processing. However, networks remain bottleneck as we try to minimize computation time in distributed systems because of clear separation between computation and networking resources. In application-aware networking (AAN) approach, distributed systems communicate their intent to data center networks. Based on the applications' knowledge, networks adapt to traffic demands by optimizing schedulers, reconfiguring topology, and routing mechanisms.

We propose AAN approach for distributed Spark clusters to reduce the computation time of Spark applications using software-defined networking (SDN). Like Coflow [1], we are specially interested to manage networks using application information (e.g., resource requirement) from computing frameworks. In our AAN approach, PANE OpenFlow controller [3] is leveraged as the basic implementation. The existing PANE controller delegates network control to user applications, however, we manage network traffic using Spark computing framework like Hadoop [6].

Thesis Goals

The goal of the thesis is to design and implement AAN architecture for Spark computing system. In the proposed system, the candidate is required to implement Coflow in PANE controller and PANE client in Spark such that the computing framework can communicate intent to PANE controller. For instance, it can reserve guaranteed network resources for flows in shuffle phase. The candidate is required to benchmark [2, 4] the performance of AAN system using CPU, memory, and network intensive workloads for Spark applications executing with and without AAN.

Milestones

- Familiarizing with Spark computing framework and PANE OpenFlow controller.
- Implementing PANE client in Spark.
- Extending PANE OpenFlow controller to implement Coflow.
- Designing and implementing AAN architecture.
- Evaluating and comparing the performance of benchmarks, for instance, job completion time of Spark applications executing with and without AAN.

This thesis topic is scalable and the workload can be adapted to fit bachelor or master thesis requirements.

Required knowledge

- Basic networking knowledge
- Python, Java, or Scala programming
- Software development and testing

References

[1] M. Chowdhury and I. Stoica. Coflow: A networking abstraction for cluster applications. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, pages 31–36, New York, NY, USA, 2012. ACM.

[2] Databricks. Spark Performance. `https://github.com/databricks/spark-perf`.

[3] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi. Participatory networking: An api for application control of sdns. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 327–338, New York, NY, USA, 2013. ACM.

[4] Spark Technology Center. Spark Benchmark. `https://github.com/SparkTC/spark-bench`.

[5] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster computing with working sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.

[6] S. Zhao, A. Sydney, and D. Medhi. Building application-aware network environments using sdn for optimizing hadoop applications. In *Proceedings of the 2016 Conference on ACM SIGCOMM 2016 Conference*, SIGCOMM '16, pages 583–584, New York, NY, USA, 2016. ACM.

Contact: Asif Hasnain, asif.hasnain@upb.de, Phone: 05251 / 60 1757, Room: O3.164