Accelerating Apache Spark with Fixed Function Hardware Accelerators Near DRAM and NVRAM

Ahsan Javed Awan*,
*KTH Royal Institute of Technology

I. Motivation

Our recent work on performance characterization of Apache Spark on a Scale-up Server shows that performance of Spark workloads is limited by the latency of frequent accesses to DRAM [1] and as the input data size is enlarged, the DRAM capacity becomes the bottleneck due to frequent spilling to hard disk [2]. In-Memory processing and In-storage processing can be combined with a hybrid architecture where the host is connected to DRAM with custom accelerators and flash based NVM with integrated hardware units to reduce the data movement.The main goal of the project is to quantify the performance gain obtained by offloading the memory bound Spark operations to near DRAM accelerators and I/O bound spark operations to near NVRAM accelerators, which we have envisioned to remove the performance bottlenecks in Apache Spark. Figure 1 shows the node architecture [3].

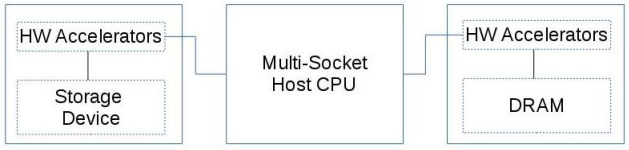

Fig. 1: NDC Supported Single Node in Scale-in Clusters for in-Memory Data Analytics with Spark

Let's consider an example. Many transformations in Spark such as groupByKey, reduceByKey, sortByKey, join etc involve shuffling of data between the tasks. To organise the data for shuffle, spark generates set of tasks; map tasks to organise the data and a set of reduce tasks to aggregate it. Map output records from each task are kept in memory until they can't fit. At that point records are sorted by reduce tasks for which they are destined and then spilt to a single file. Since the records are dispersed throughout the memory, they results in poor cache locality and sorting them on CPU will experience a significant amount of cache misses and using near DRAM hardware accelerators for sort function, this phase can be accelerated. If this process occurs multiple times, the spilt segments are merged later. On the reduce side, tasks read the relevant sorted blocks. A single reduce task can receive blocks from thousands of map tasks. To make this many-way merge efficient, especially in the case where the data does not fit in memory, It is better to use hardware accelerators for merge function near the faster persistent storage device like NVRAM.

II. Proposed Work

In order to address the problem, we aim to perform following tasks

- Characterize Spark-core, Spark MLlib, Graph-X, Spark Streaming and Spark SQL applications into compute bound, memory bound and I/O bound. We will use hardware performance counters to identify compute bound and memory bound applications and OS level metrics like CPU utilisation, idle time, wait time on I/O and major and minor page faults to filter out the I/O bound applications in Apache Spark.
- Identify memory bound and I/O bound Spark operations through intra-phase analysis of Spark applications. We aim of to use Intel Vtune for that purpose.
- Quantify the performance gain by offloading the memory bound Spark operations to near DRAM accelerators and I/O bound Spark operations to near NVRAM accelerators. We will rely on analytical models for this task.
- Write a paper based on the results obtained and submit to International Symposium on Computer Architecture (ISCA).

References

[1] A. Javed Awan, M. Brorsson, V. Vlassov, and E. Ayguade, "Performance characterization of in-memory data analytics on a modern cloud server," in *Big Data and Cloud Computing (BDCloud), 2015 IEEE Fifth International Conference on*. IEEE, 2015, pp. 1–8.

[2] A. J. Awan, M. Brorsson, V. Vlassov, and E. Ayguade, *Big Data Benchmarks, Performance Optimization, and Emerging Hardware: 6th Workshop, BPOE 2015, Kohala, HI, USA, August 31 - September 4, 2015. Revised Selected Papers*. Springer International Publishing, 2016, ch. How Data Volume Affects Spark Based Data Analytics on a Scale-up Server, pp. 81–92.

[3] A. J. Awan, "Performance characterization of in-memory data analytics on a scale-up server," Licentiate Thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, May. 2016.