Big Data team work summary

The Linaro Big data team have been involved primarily with two efforts (Contributions to ODPi and Bigtop).

Big Data is vast and have tons of components and tools. We could categorize all the big data efforts into the following:

- Enablement and Testing

The Big data team have taken a ‘top-down’ and ‘bottom-up’ approach in tackling porting of huge number of components. As a ‘top-down’ approach we have been focusing on operations tool - Apache Ambari. And as a ‘bottom-up’ approach, we triage and prioritize the most commonly used/industry standard components and focus on them first. The prioritization is based on voting and vetting process from Linaro steering committee members. As core components are included in ODPi and all top apache big data components (Hadoop - YARN, Mapreduce, HDFS, Spark, Hive, HBase, Zookeeper, Kafka, etc) are included in Bigtop, has helped us to focus on them first. All these components have been tested in docker containerized clustered environment. We have contributed a lot of patches to Bigtop and AArch64 is now a first class citizen with Apache Bigtop. The upstreaming of patches with Bigtop community is pretty quick and responsive. maintainers are also ASF maintainers which help us to speed up the upstreaming process to ASF. But, there are quite a lot of components that are still not part of Bigtop that we still need to focus on through ASF. E.g., Apache Flink, Beam, etc. From bottom-up approach, the components can be grouped into following categories

- Data Collection components
- Batch and Real time processing components
- Storage components
- Distributed database components
- Analytical components
- Data Science and Machine Learning

- Automated build, CI with smoke testing

We have already setup a CI with upstream and inside linaro that utilizes docker containers. We have been working on smoke tests and unit tests. We need to automate the CI with tests. https://ci.bigtop.apache.org/job/Bigtop-trunk-packages/

- Profile and optimize

Big Data applications stress all system components, such as CPU cores, memory, storage and network I/O. Hence, we need to first evaluate the individual peak performance of these components, before running complex data-intensive workloads. For this evaluation, we need to employ benchmarks that are widely used in industry and systems research. For example, we measure how many Million Instructions per Second (MIPS) a core can deliver using traditional Dhrystone and emerging CoreMark benchmark. For storage and network throughput and latency, we use Linux tools such as dd, loping, iperf and ping. For benchmarking big data components itself, there are a lot of tools out there and there is no proper standards. Hence we are also involved and lead the Benchmarking SIG inside ODPi. As part of that we will be developing the standards and guidelines. To start with, we will be looking at industry’s most commonly used benchmarking tools like Big Bench, HiBench and TPC (TPC-HS, TPC-DS, TPC-H, etc) benchmarking tools. We have already ported most of these benchmarking tools to Aarch64. But there is still quite a bit of work in porting. As part of the benchmarking efforts we will also need to look at:

- Micro-benchmarks (Teragen, Teravaldiate and Terasort, wordcount, floating-point intensive benchmarks like PageRank, DFSIO)
- Disk I/O - Energy usage
  - Using TestDFSIO, benchmark HDFS and measure energy consumption of write and read on a single node as well as clustered environment.
- Data compression (Avro and Parquet)
  - Using TPC-C and TPC-H benchmarks evaluate disk-based query processing. Using AMPLab Big Data Benchmark, evaluate distributed, in-memory query processing (Hive, Tez, Spark)
- Benchmark Java execution
  - Run synthetic benchmarks performing integer and floating point operation that stresses core’s pipeline. Measure memory bandwidth using pmbw 0.6.2 (Parallel Memory Bandwidth Benchmark). Measure storage I/O read and write throughput and latency using dd, and loping. And measure networking subsystem bandwidth and latency using iperf and ping. Look at GC performance and optimization.
- Analytical performance Benchmarking
  - Measure performance of OLTP and OLAP workloads. TPC-H queries are read-only, I/O bounded, and represent the most common analytics scenarios in databases. Run all 22 queries of TPC-H benchmark. Using Bigbench and Hibench evaluate processing of high volume and high velocity datasets
- Machine Learning Benchmarking
  - Using Mahout - Naive Bayes Classifier training, K-Means clustering, etc
  - Scale out performance profiling and optimization in clustered environment
  - TCO profiling - Using HiBench, etc

Along with these, we also work on ERP releases. https://collaborate.linaro.org/display/BDTS/ERP+17.08+-+Build%2C+Configure%2C+Install%2C+Test+Hadoop%2C+Spark%2C+Hive%2C+Hbase+in+containerized+environment