Twister2

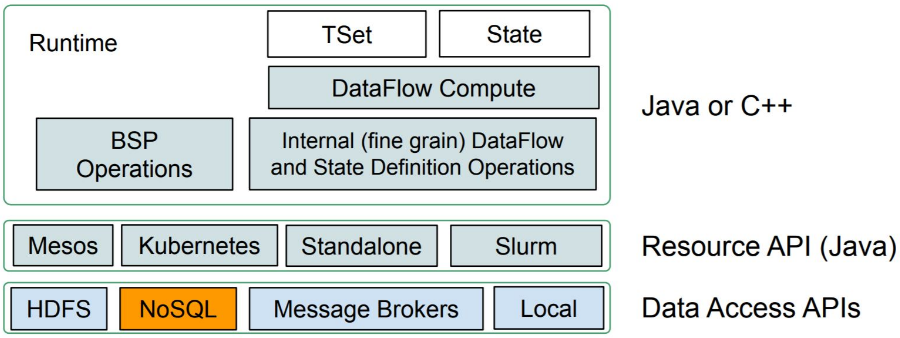
Sep 23, 2019

Digital Science Center
Indiana University Bloomington

Github	https://github.com/DSC-SPIDAL/twister2
Mailing List	twister2@googlegroups.com
Documentation	https://twister2.gitbook.io/twister2/
Slack	https://dsc-twister.slack.com

1 Introduction

A data analytics platform should support preprocessing, analytics and postprocessing steps efficiently in different computing environments. Computing environments can be characterized by hardware, virtualization technologies (i.e. Cloud vs HPC), programming languages and operating systems. A data analytics platform should be able to utilize the features of the environment to execute applications efficiently. Twister2 is a flexible, high performance data processing engine. The project is open source and available under the Apache License Version 2.0. It enables streaming and batch analysis of large data sets with high performance. Twister2 is designed from ground up to be a cloud native and scalable system. It leverages high performance hardware for better scalability and performance when available. Twister2 defines a set of core components in a layered architecture for developing and executing data analytics applications.



The core components include a resource provisioning layer to interface with cluster resource managers, parallel communication operator layers recognizing the need for both data operators and bulk synchronous parallel (BSP) operators, compute engine, and data representation for data manipulation. The dataflow operators implement data operations such as shuffle and join for batch applications and streaming applications. The fine grain operators are BSP operators specified in MPI specification. These components can be implemented in different programming languages, taking advantage of hardware and environments. For example, the distributed operators can use hardware such as Infiniband or GPUs to increase the performance.

Twister2 will have three deployment modes supporting different use cases. 1. Java Framework, 2. C++ Library, 3. Python API. The Java framework provides data management capabilities. C++ library enables advanced use cases where there is a need for deep integration of algorithms and data analytics capabilities. Python API is used primarily for AI and data integration. Python API will be backed by C++ implementation and the Java implementation to support different use cases for Python users.

2 Core Components

Twister2 consists of several core components that provide the essential features of a data analytics platform. These components can be implemented in different programming languages, taking advantage of hardware to accelerate data processing.

2.1 Resource provisioning

Resource provisioning component handles the job submission and management. It provides abstractions to acquire resources and manage the life cycle of a parallel job. The current system supports Standalone, Kubernetes, Mesos, Slurm, Nomad resource managers and would like to add others such as Yarn in the future. Twister2 jobs run in isolation without sharing resources among them. Unlike in other big data systems such as Spark or Flink, Twister2 can be easily deployed in both HPC and cloud environments.

2.2 Parallel and Distributed communication operators

Twister2 recognizes the importance of network communication operators for parallel applications and provides a set of abstractions and implementations that satisfy the needs of different applications. Both streaming operators and batch operators are provided. Three types of operator implementations are provided to the user.

1. Twister:Net - a data level dataflow operator library for streaming and batch operations
2. Harp - a BSP (Bulk Synchronous Processing) innovative collective framework for parallel applications and machine learning at the message level
3. OpenMPI (HPC Environments only) at the message level

In other big data systems, communication operators are hidden behind high-level APIs. Twister2 design allows a user to program with these low level APIs even though they can use more higher level APIs. These APIs allow the development of different methods to optimize the performance. Users can utilize them to program high-performance applications when needed. These APIs are relatively difficult to program compared to the higher level APIs provided by the system. Twister:Net can use both socket-based or MPI based (ISent/IRecv) to communicate allowing it to perform well on advanced hardware.

2.3 Compute Engine

This component provides the abstractions to hide the execution details and an easy to program API for parallel applications. A user can create a streaming task graph or a batch dataflow graph to analyze data. The abstractions have similarities to Storm and Hadoop APIs. Compute engine consists of the following major components.

1. Dataflow graph - Create dataflow graphs for streaming and batch analysis including iterative computations
2. Scheduler - Schedules the graph into cluster resources with different algorithms
3. Executor - Batch and streaming executions with use of processes, threads

In Spark and Flink, this component is hidden from the user. Apache Storm and Hadoop API's are similar this level of abstraction. Twister2 allows pluggable executors and task schedulers to extend the capabilities of the system. Compute Engine supports the efficient execution of iterative applications by optimizing the network operations and graph executions.

2.4 Distributed Data API

A typed API similar to Spark RDD, BEAM PCollections or Heron Streamlet is provided here. It can be used to program a data pipeline, a streaming application or an iterative application. In Twister2, iterations (for loops) are carried on each worker. Spark uses a central driver to control the iterations, which can lead to poor performance for applications with frequent iterations (less computation in an iteration). Flink uses the task graph itself to code the iterations (cyclic graphs) and doesn't support nested iterations. Data pipelines in Twister2 are similar to Flink or Spark. Twister2 is a pure streaming engine making it similar to Flink or Storm (not a minibatch system like Spark).

2.5 Auxiliary Features

Apart from the main futures, it provides the following components.

1. Web UI for monitoring Jobs
2. Connected DataFlow, this is for workflow type jobs
3. Data access API for connecting to different data sources (File systems)

3 Deployment Modes

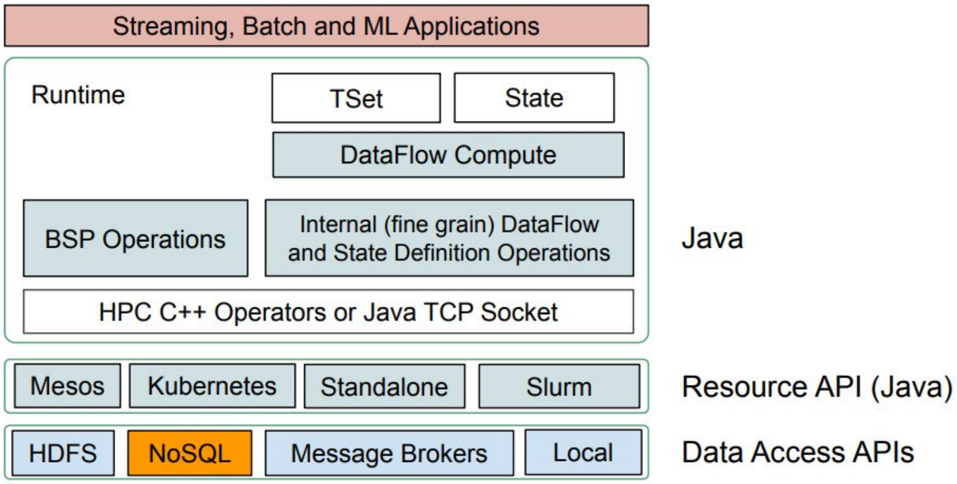
The three deployment modes will facilitate different use cases starting from standard data management, data and AI integration and data and HPC application integration.

3.1 Java Framework

Java framework is a full data processing framework for streaming and batch data pipelines. It provides higher level APIs and capabilities similar to other big data frameworks. Here are some key factors that differentiate the Java framework from other big data frameworks.

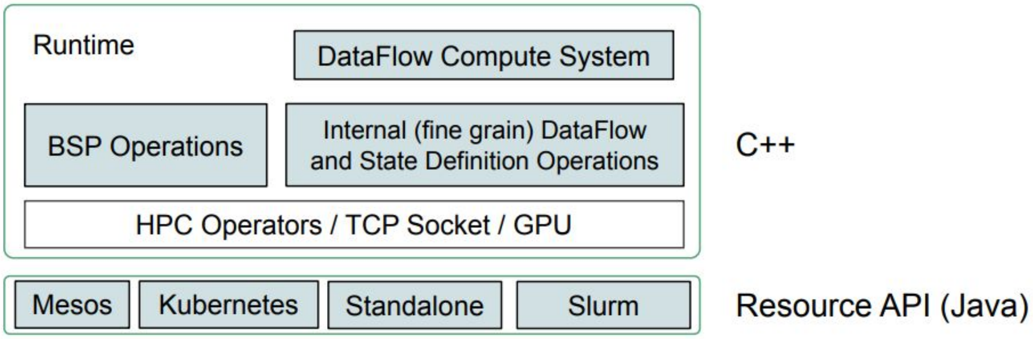
Twister2 is the only big data that supports true streaming and batch applications. Iterations are much more efficient compared to other frameworks. This is because they happen parallelly in all workers. For example Spark does iterations centrally and needs to distribute tasks for each iteration to workers (high overhead). Flink tries to encode iterations into dataflow graph and cannot support nested iterations.

Twister2 provides multiple APIs for programming an application giving users the flexibility to choose between performance and ease of programming. Ability to utilize high-performance networks for better throughput and latency



3.2 C++ Library

C++ version is in development and first version is planned to be completed at the end of year 2019. This implementation will allow data scientists to integrate data analytics, AI and HPC simulations seamlessly. The data copying and data transformation bottlenecks in the current systems will be non-existent in this approach. Also, it can target new data analytics technologies such as FPGAs and GPUs. We believe C++ implementation will be used as a library in other environments such as MPI implementations to gain data analytics capabilities.



3.3. Python API

Python API is targeted towards data and AI integration. A data scientist can experiment with her data quickly on the laptop and deploy the solution to a data center with the Python API. Python API will be backed by both C++ implementation and Java implementation giving users the flexibility to choose the environment they prefer for the backend.

4 Supported Applications and APIs

Twister2 supports three APIs for programming data applications. These are Operator API, Compute API and Dataset API.

Java provides all three API levels. C++ provides the operator level and compute level abstractions and Python API will only be available at the data level. Dataset API is similar to Apache Spark RDD API, Flink DataSet API or Beam PCollection API. It views computation as a set of data transformations. Compute API provides higher level abstractions similar to Hadoop API or APache Storm API for programming a dataflow application. Fine control of the parallel applications both with coarse grain dataflow operations and fine grain HPC operations.

4.1 Apache Beam and Apache Storm API Compatibility

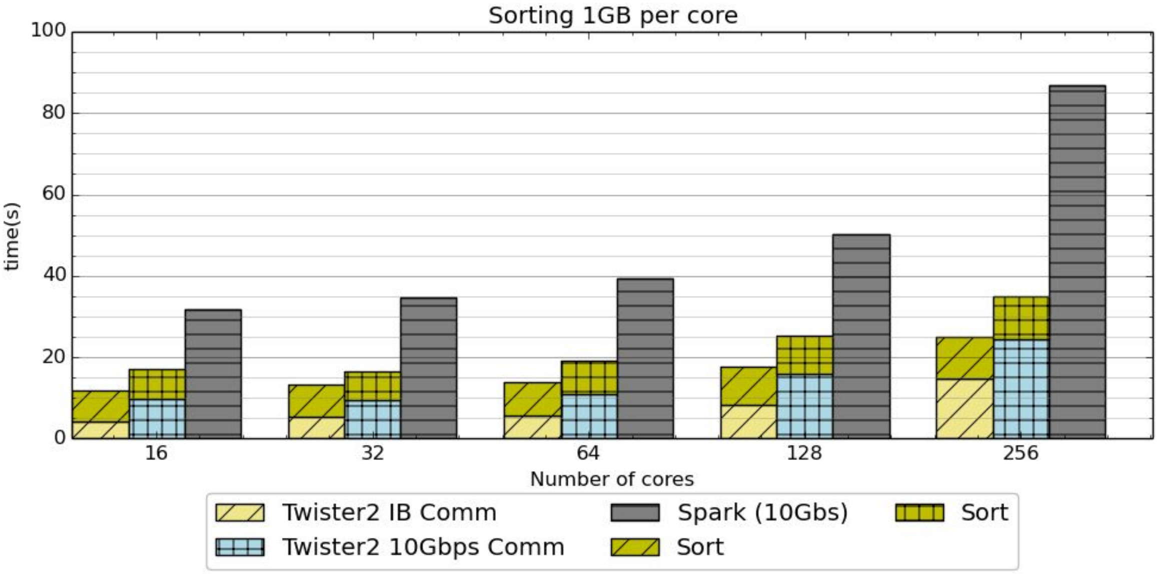
Twister2 supports Apache Beam API for developing applications. Beam provides an API with pluggable engines and both Spark and Flink can run with the Beam API. Beam API can be used to write data pipeline applications including batch and streaming. Also, it provides SQL capabilities to Twister2. Apache Storm API compatibility allows a user to drop a Storm application directly into Twister2 and execute it with the Twister2 backend.

5 Performance

Twister2 has been designed and implemented for high performance in all its components including dataflow, communications (with different modes optimized for different usage models), iterative applications, streaming, scheduling and orchestration. We measured the performance compared to leading frameworks in key areas.

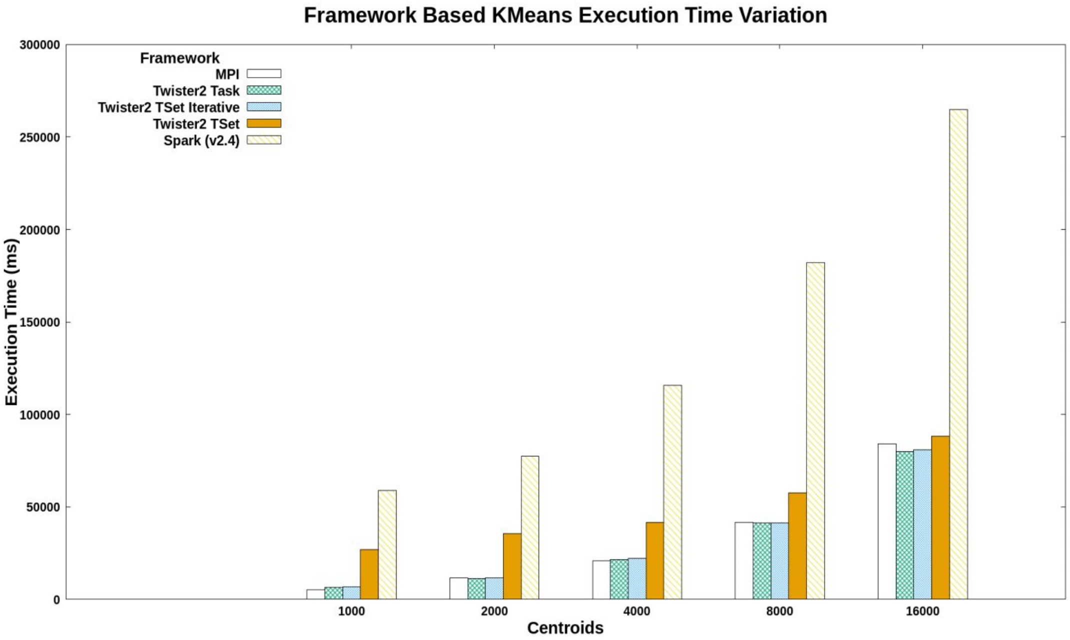
5.1 Big data

TeraSort is an important benchmark in big data systems as it measures the performance of the shuffle operation. Below is the performance of the TeraSort benchmark compared to Spark on 10Gbs Ethernet and 32Gbps Infiniband network. In this experiment 1GB of data is assigned to each core, meaning when there are 256 cores, there is 256GB of data. Each record has a 10byte key and 90byte value.

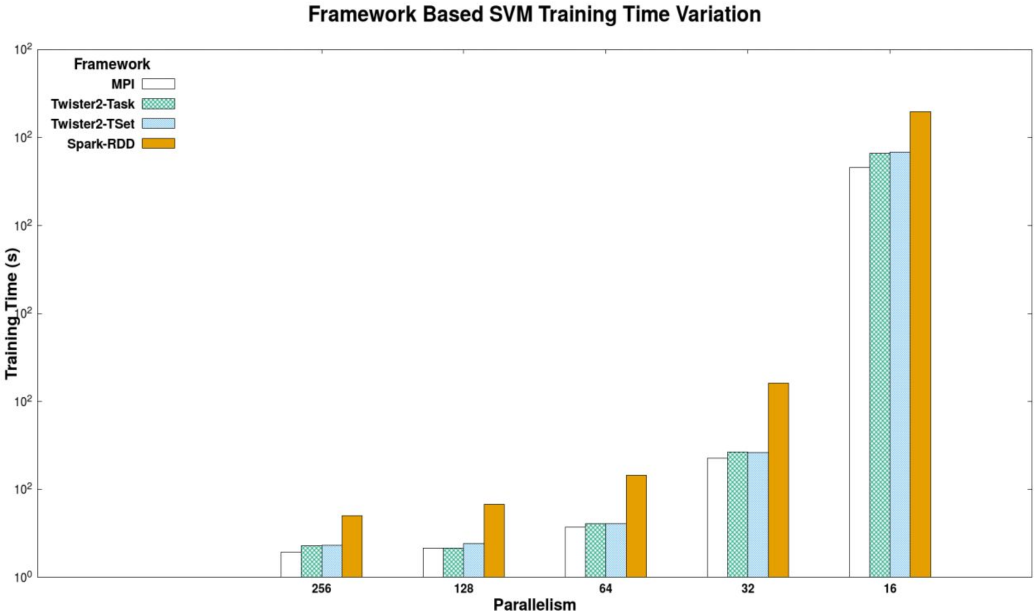


5.2 Machine Learning

The figure below shows the performance of K-Means machine learning algorithm compared to Spark and OpenMPI implementations. The experiment was done on 16 nodes with a varying number of centroids (x-axis), 2 million data points, 128-way parallelism, and 100 iterations.

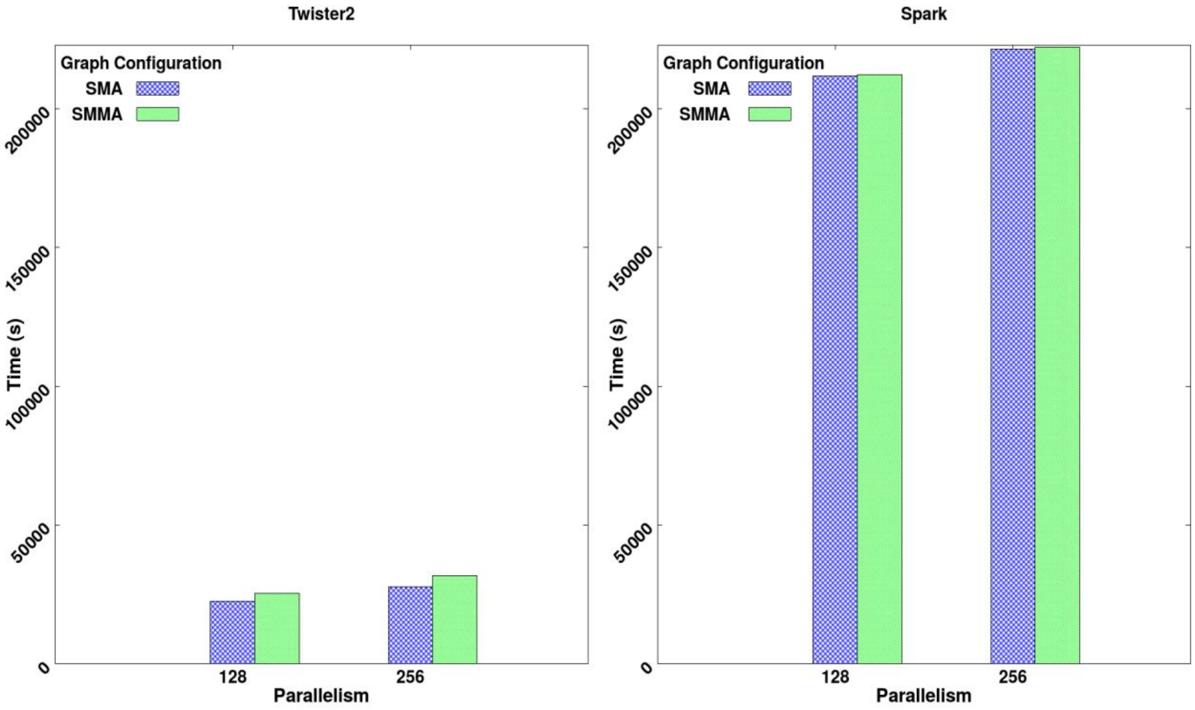


Below is the performance of SVM training with different frameworks, 320K data points, 2000 features and 500 iterations, on 16 nodes with varying parallelism.



5.3 DataFlow

Below is a graph that demonstrates the dataflow performance of Twister2 compared to Spark. Execution time for Source-Map-AllReduce (SMA) and Source-Map-Map-AllReduce(SMMA) graph configurations with 200K data points and 100 iterations.



Twister2 achieves better performance compared to leading big data frameworks. We believe performance can be further improved with the addition of more sophisticated algorithms for execution of the system.

6 Roadmap

In the near term, we will focus on AI integration, C++ implementation, and consolidation of the Java Framework.

6.1 Q4 2019

The main focus will be on consolidating the Java framework further and adding the C++ framework. Integration with Rapids.ai from Nvidia for high performance data acceleration with GPUs is another area we will focus. Python interface integration with TensorFlow will be done for integration of AI.

We will continue to integrate Twister2 with other Apache projects and implement applications on top of the framework. At the end of the year we will be pushing towards getting into Apache Incubator.

6.2 2020 Year

There are many optimizations and enhancements possible in the system. Some of the possible improvements include adding support for UCX for network acceleration, fault tolerance for OpenMPI based applications.

7 Discussion

The latest release of the Twister2 is 0.3.0 version. 0.4.0 release of Twister2 is a major milestone release where it is expected to complete the major features of Twister2 when compared to other Java frameworks. This release is expected to be at the beginning of October 2019. This release will include Python API backed by the Java runtime. Twister2 will have a fully supported Apache Beam API. Fault tolerance of Twister2 will be completed with this release as well. Twister2 team is working on implementing the C++ implementation. This implementation is expected to propel Twister2 far ahead of the existing systems in terms of capabilities and performance by greatly enhancing the integration of AI and data analytics.

Twister2 is an efficient data processing engine with capabilities to support different programming models and applications. The Java version is fully functional and can be used for data analytics tasks.