ApexProposal

Abstract

Apex is an enterprise grade native YARN big data-in-motion platform that unifies stream processing as well as batch processing. Apex processes big data in-motion in a highly scalable, highly performant, fault tolerant, stateful, secure, distributed, and an easily operable way. It provides a simple API that enables users to write or re-use generic Java code, thereby lowering the expertise needed to write big data applications.

Functional and operational specifications are separated. Apex is designed in a way to enable users to write their own code (aka user defined functions) as is and leave all operability to the platform. The API is very simple and is designed to allow users to drop in their code as is. The platform mainly deals with operability and treats functional code as a black box. Operability includes fault tolerance, scalability, security, ease of use, metrics api, webservices, etc. In other words there is no separation of UDF (user defined functions), as all functional code is UDF. This frees users to focus on functional development, and lets platform provide operability support. The same code runs as is with different operability attributes. The data-in-motion architecture of Apex unifies stream as well as batch processing in a single platform. Since Apex is a native YARN application, it leverages all the components of YARN without duplication. Apex was developed with YARN in mind and has no overlapping components/functionality with YARN.

The Apex platform is supplemented by project Malhar, which is a library of operators that implement common business logic functions needed by customers who want to quickly develop applications. These operators provide access to HDFS, S3, NFS, FTP, and other file systems; Kafka, ActiveMQ, RabbitMQ, JMS, and other message systems; MySql, Cassandra, MongoDB, Redis, HBase, CouchDB and other databases along with JDBC connectors. The Malhar library also includes a host of other common business logic patterns that help users to significantly reduce the time it takes to go into production. Ease of integration with all other big data technologies is one of the primary missions of Malhar.

Proposal

The goal of this proposal is to establish the core engine of DataTorrent RTS product as an Apache Software Foundation (ASF) project in order to build a vibrant, diverse, and self-governed open source community around the technology. DataTorrent will continue to sell management tools, application building tools, easy to use big data applications, and custom high end business logic operators. This proposal covers the Apex source code (written in Java), Apex documentation and other materials currently available on https://github.com/DataTorrent/Apex. This proposal also covers the Malhar source code (written in Java), Malhar documentation, and other materials currently available on https://github.com/DataTorrent/Malhar. We have done a trademark check on the name Apex, and have concluded that the Apex name is likely to be a suitable project name.

Background

DataTorrent RTS is a mature and robust product developed as a native YARN application. RTS 1.0 was launched in summer of 2014; RTS 2.0 was launched in Jan 2015. Both were well received by customers. RTS 3.0 was launched at end of July 2015. RTS is among the first enterprise grade platform that was developed from the ground up as native YARN application. DataTorrent RTS is currently maintained by engineers as a closed source project. Even though the engineers behind RTS are experienced software engineers and are knowledge leaders in data-in-motion platforms, they have had little exposure to the open source governance process. Customers are currently running applications based on DataTorrent RTS in production.

Rationale

Big data applications written for non-Hadoop platforms typically require major rewrites to get them to work with Hadoop. This rewriting creates a significant bottleneck in terms of resources (expertise) which in turn jeopardizes the viability of such an endeavour. It is hard enough to acquire big data expertise, demanding additional expertise to do a major code conversion makes it a very hard problem for projects to successfully migrate to Hadoop. Also, due to the batch processing nature of Hadoop's MapReduce paradigm, users often have to wait tens of minutes to see results and act on them due to various delays in data flow. DataTorrent's RTS data-in-motion architecture is designed to address this problem. It enables even the non big data developer to write code and operate it in a scalable, fault tolerant manner. The big data-in-motion architecture of DataTorrent's RTS enables ease of integration into current enterprise infrastructure. This goal was achieved by keeping the API simple and empowering users to put in the connector code as is (or with minimal changes).

Malhar is a manifestation of this reality, and we or the customer engineers were able to create these connectors within a day or so if not within a week. Connectors include those to integrate with message bus(es), file systems, databases, other protocols, and more continue to be added. Over a period of time we expect users to simply pick a connector that already exists in Malhar and quickly begin integrating with their current enterprise infrastructure. Within the data-in-motion architecture a stream application is one with connector(s) to say Kafka, JMS, or Flume; while a batch application is one with connector(s) to HDFS, HBase, FTP, NFS, S3n etc. This allows usage of the platform for both stream as well as batch processing with same business logic. Complete separation of user written application code from all operational aspects of the system, as well as support code for YARN, significantly expands the potential use cases that can migrate to use Hadoop.

Apex will enable Hadoop eco-system to migrate a lot more use cases. It will enable the Hadoop eco-system to deliver on a promise to rapidly transform current IT infrastructure. Apex will help in significantly increasing productization of big data projects. One of the main barometers of success in the Hadoop eco-system is significant reduction of time to market for big data applications migrating to Hadoop. We believe that Apex will be one of the platforms that will enable users to extract value from big data, by reducing time to market. This rapid innovation can be optimally achieved through a vibrant, diverse, self-governed community collectively innovating around Apex and the Malhar library, while at the same time cross-pollinating with various other big data platforms. ASF is an ideal place to meet this goal.

Initial Goals

Our initial goals are to bring Apex and Malhar repositories into the ASF, adapt internal engineering processes to open development, and foster a collaborative development model in accordance with the "Apache Way." DataTorrent plans to develop new functionality in an open, community-driven way. To get there, the existing internal build, test and release processes will be refactored to support open development. We already have an active user community on google groups that we intend to migrate to Apache.

Current Status

Currently, the project Apex code base is available under Apache 2.0 license (https://github.com/DataTorrent/Apex). Project Malhar code base is available under Apache 2.0 license (https://github.com/DataTorrent/Malhar). Project Malhar was open sourced 2 years ago which should make it easy for the project Malhar team to adapt to an open, collaborative, and meritocratic environment. Contributors of Malhar are employees of DataTorrent or have agreed to the shift to Apache. Project Apex, in contrast, was developed as a proprietary, closed-source product, but the internal engineering practices adopted by the development team were common to Malhar, and should lend themselves well to an open environment. DataTorrent plans to execute a software grant agreement as part of the launch of the incubation of Apex as an Apache project.

The DataTorrent team has always focused on building a robust end user community of paying and non-paying customers. We think that the existing community centered around the existing google groups mailing list should be relatively easy to transform into an Apache-style community including both users and developers.

Meritocracy

Our proposed list of initial committers include the current RTS R&D team, and our existing customers. This group will form a base for the broader community we will invite to collaborate on the codebase. We intend to radically expand the initial developer and user community by running the project in accordance with the "Apache Way". Users and new contributors will be treated with respect and welcomed. By participating in the community and providing quality patches/support that move the project forward, they will earn merit. They also will be encouraged to provide non-code contributions (documentation, events, presentations, community management, etc.) and will gain merit for doing so. Those with a proven support and quality track record will be encouraged to become committers.

Community

If Apex is accepted for incubation, the primary initial goal will be transitioning the core community towards embracing the Apache Way of project governance. We will solicit major existing contributors to become committers on the project from the start. It should be noted that the existing community is already more diverse in many ways than some top-level Apache projects. We expect that we can encourage even more diversity.

Core Developers

While a few core developers are skilled in working in openly governed Apache communities, most of the core developers are currently NOT affiliated with the ASF and would require new ICLAs before committing to the project. There would also be a learning curve associated with this on-boarding. Changing current development practices to be more open will be an important step.

Alignment

The following existing ASF projects provide related functionality as that provided by Apex and should be considered when reviewing Apex proposal:

Apache Hadoop is a distributed storage and processing framework for very large datasets focusing primarily on batch processing for analytic purposes. Apex is a native YARN application. The Apex and Malhar roadmap includes plans to continue to leverage YARN, and help the YARN community develop the ability to support long running applications. Apex uses DFS interface of its core checkpoint/commit. Malhar has a large number of operators that leverage HDFS and other Apache projects. Our roadmap includes plans to continue to deepen the currently close integration with HDFS.

Apache HBase offers tabular data stored in Hadoop based on the Google Bigtable model. Malhar has HBase connectors to ease integration with HBase. Malhar roadmap includes plans to continue to enhance integration with Apache HBase.

Apache Kafka offers distributed and durable publish-subscribe messaging. Malhar integrates Kafka with Hadoop through feature rich connectors and supports ingest as well as analytical functions to incoming data. Raw data can be ingested from Kafka and results can be written to Kafka. Malhar roadmap includes plans to continue to enhance integration with Apache Kafka.

Apache Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. Malhar has Flume connectors to ease integration with Flume. These connectors ensures that ingestion with Flume is fault tolerant and thus can be done in real-time with the same SLA as Flume's HDFS connectors. Malhar roadmap includes plans to continue to enhance integration with Apache Flume.

Apache Cassandra is a highly scalable, distributed key-value store that focuses on eventual consistency. Malhar has connectors to ease integration with Cassandra. Malhar roadmap includes plans to continue to enhance integration with Apache Cassandra.

Apache Accumulo is a distributed key-value store based on Google's BigTable design. Malhar has connectors to ease integration with Accumulo. The Malhar roadmap includes plans to continue to enhance integration with Apache Accumulo.

Apache Tez is aimed at building an application framework which allows for a complex DAG of tasks for process data. The Apex and Malhar roadmaps include plans to integrate with Apache Tez but this is not currently supported.

Apache ActiveMQ and its sub project Apache Apollo offers a powerful message queue framework. Malhar has ActiveMQ connectors that ease integration with ActiveMQ.

Apache Spark is an engine for processing large datasets, typically in a Hadoop cluster. Malhar project makes it easy for users to integrate with Spark. The Malhar roadmap includes plans to continue to enhance integration with Apache Spark.

Apache Flink is an engine for scalable batch and stream data processing. Malhar project makes it easy for users to integrate with Flink. There is overlap in how Flink leverages data-in-motion architecture for both stream and batch processing, and it does subscribe to our thought process that data-in-motion can handle both stream and batch, meanwhile a batch only engine will find it harder to manage streams. We differ in terms of how we handle operability, user defined code, metrics, webservices etc. Apex is very operational oriented, while Flink has much more focus on functional elements. Malhar and rapid availability of common business logic is another differentiator. We believe both these approaches are valid and the community and innovation will gain by through cross pollination. We plan to integrate with Apache Flink via HDFS for now.

Apache Hive software facilitates querying and managing large datasets residing in distributed storage. Malhar project makes it easy for users to integrate with Apache Hive. The Malhar roadmap includes plans to continue to enhance integration with Apache Hive.

Apache Pig is a platform for analyzing large data sets. Pig consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The Apex and Malhar roadmaps include plans to integrate with Apache Pig.

Apache Storm is a distributed realtime computation system. Malhar makes it easy for users to integrate with Apache Storm. We plan to integrate with Apache Storm via HDFS for now. Malhar roadmaps include plans to continue to support mechanism for integration with Apache Storm.

Apache Samza is a distributed stream processing framework. Malhar makes it easy for users to integrate with Apache Samza. We plan to integrate with Apache Samza via HDFS or Apache Kafka for now. Malhar roadmaps include plans to continue to support mechanism for integration with Apache Samza.

Apache Slider is a YARN application to deploy existing distributed applications on YARN, monitor them, and make them larger or smaller as desired even when the application is running. Once Slider matures, we will take a look at close integration of Apex with Slider.

Project Malhar and Apex are aligned to many more Apache projects and other open source projects as ease of integration with other technologies is one of the primary goals of this project. These include Apache Solr, ElasticSearch, MongoDB, Aerospike, ZeroMQ, CouchDB, CouchBase, MemCache, Redis, RabbitMQ, Apache Derby.

Known Risks

Development has been sponsored mostly by a single company (DataTorrent, Inc.) thus far and coordinated mainly by the core DataTorrent RTS and Malhar team, with active participation from our current customers.

For the project to fully transition to the Apache Way governance model, development must shift towards the merit-centric model of growing a community of contributors balanced with the needs for extreme stability and core implementation coherency.

The tools and development practices in place for the DataTorrent RTS and Malhar products are compatible with the ASF infrastructure and thus we do not anticipate any on-boarding pains. Migration from the current GitHub repository is also expected to be straightforward.

Orphaned products

DataTorrent is fully committed to DataTorrent Apex and Malhar and the product will continue to be based on the Apex project. Moreover, DataTorrent has a vested interest in making Apex succeed by driving its close integration with sister ASF projects. We expect this to further reduce the risk of orphaning the product.

Inexperience with Open Source

DataTorrent has embraced open source software by open sourcing Malhar project under Apache 2.0 license. The DataTorrent team includes veterans from the Yahoo! Hadoop team. Although some of the initial committers have not been developers on an entirely open source, community-driven project, we expect to bring to bear the open development practices of Malhar to the Apex project. Additionally, several ASF veterans agreed to mentor the project and are listed in this proposal. The project will rely on their guidance and collective wisdom to quickly transition the entire team of initial committers towards practicing the Apache Way. DataTorrent is also driving the Kafka on YARN (KOYA) initiative.

Homogeneous Developers

While most of the initial committers are employed by DataTorrent, we have already seen a healthy level of interest from our existing customers and partners. We intend to convert that interest directly into participation and will be investing in activities to recruit additional committers from other companies.

Reliance on Salaried Developers

Most of the contributors are paid to work in the Big Data space. While they might wander from their current employers, they are unlikely to venture far from their core expertises and thus will continue to be engaged with the project regardless of their current employers.

Relationships with Other Apache Products

As mentioned in the Alignment section, Apex may consider various degrees of integration and code exchange with Apache Hadoop (YARN and HDFS), Apache Kafka, Apache HBase, Apache Flume, Apache Cassandra, Apache Accumulo, Apache Tez, Apache Hive, Apache Pig, Apache Storm, Apache Samza, Apache Spark, Apache Slider. Given the success that the DataTorrent RTS product enjoyed, we expect integration points to be inside and outside the project. We look forward to collaborating with these communities as well as other communities under the Apache umbrella.

An Excessive Fascination with the Apache Brand

While we intend to leverage the Apache 'branding' when talking to other projects as testament of our project's 'neutrality', we have no plans for making use of Apache brand in press releases nor posting billboards advertising acceptance of Apex into Apache Incubator.

Documentation

See documentation for the current state of the project documentation available as part of the GitHub repositories - https://github.com/DataTorrent/Apex; https://github.com/DataTorrent/Malhar. In addition a list of demos that serve as a how to guide are available at https://github.com/DataTorrent/Malhar/tree/master/demos

Initial Source

DataTorrent has released the source code for Apex under Apache 2.0 License at https://github.com/DataTorrent/Apex, and that of Malhar under Apache 2.0 licence at https://github.com/DataTorrent/Malhar. We encourage ASF community members interested in this proposal to download the source code, review it and try out the software.

Source and Intellectual Property Submission Plan

As soon as Apex is approved to join Apache Incubator, DataTorrent will execute a Software Grant Agreement and the source code will be transitioned onto ASF infrastructure. The code is already licensed under the Apache Software License, version 2.0. We know of no legal encumberments that would inhibit the transfer of source code to the ASF.

External Dependencies

All dependencies fall under the permissive licenses categories, or weak copy left (http://www.apache.org/legal/resolved.html#category-b). We intend to remove the dependencies on GPL licensed technologies on which APex or Malhar depend. These technologies are optional and have been marked as such.

Embedded dependencies (relocated):

- None

Runtime dependencies:

- activemq-client
- ant
- async-http-client
- bval-jsr303
- commons-beanutils
- commons-codec
- commons-lang3
- commons-compiler
- embassador
- fastutil
- guava
- hadoop-common
- hadoop-common-tests
- hadoop-yarn-client
- httpclient
- jackson-core-asl
- jackson-mapper-asl
- javax.mail
- jersey-apache-client4
- jersey-client
- jetty-servlet
- jetty-websocket
- jline
- kryo
- named-regexp
- netlet
- rhino (GPL 2.0, optional)
- slf4j-api
- slf4j-log4j12
- validation-api
- xbean-asm5-shaded
- zip4j

Module or optional dependencies

- accumulo-core
- aerospike-client
- amqp-client
- aws-java-sdk-kinesis
- cassandra-driver-core
- couchbase-client
- CouchbaseMock
- elasticsearch
- geoip-api (LGPL, optional)
- hbase
- hbase-client
- hbase-server
- hive-exec
- hive-service
- hiveunit
- javax.mail-api
- jedis

- jms-api
- jri (GPL, optional)
- jriengine (LGPL, optional)
- jruby (LGPL, optional)
- jython (PSF License, optional)
- jzmq (LGPL, optional)
- kafka_2.10
- lettuce (GPL, optional)
- libthrift
- Memcached-Java-Client
- mongo-java-driver
- mqtt-client
- mysql-connector-java (GPL2, optional)
- org.ektorp
- rengine (LGPL, optional)
- rome
- solr-core
- solr-solrj
- spymemcached
- sqlite4java
- super-csv
- twitter4j-core
- twitter4j-stream
- uadetector-resources
- org.apache.servicemix.bundles.splunk

Build only dependencies:

- None

Test only dependencies:

- activemq-broker
- activemq-kahadb-store
- greenmail
- hadoop-yarn-server-tests
- hsqldb
- janino
- junit
- MockFtpServer
- mockito-all
- testng

Cryptography N/A

Required Resources

Mailing lists

- private@apex.incubator.apache.org (moderated subscriptions)
- commits@apex.incubator.apache.org
- dev@apex.incubator.apache.org

Git Repository

- https://git-wip-us.apache.org/repos/asf/incubator-apex-core.git
- https://git-wip-us.apache.org/repos/asf/incubator-apex-malhar.git

Issue Tracking

- JIRA Project Apex (APEX_CORE) // If '_' is not allowed, use APEXCORE
- JIRA Project Malhar (APEX_MALHAR) // If '_' is not allowed use APEXMALHAR

Other Resources

- Means of setting up regular builds for apex-core on builds.apache.org
- Means of setting up regular builds for apex-malhar on builds.apache.org

Rationale for Malhar and Apex having separate git and jira

We managed Malhar and Apex as two repos and two jiras on purpose. Both code bases are released under Apache 2.0 and are proposed for incubation. In terms of our vision to enable innovation around a native YARN data-in-motion that unifies stream processing as well as batch processing Malhar and Apex go hand in hand. Apex has base API that consists of java api (functional), and attributes (operability). Malhar is a manifestation of this api, but from user perspective, Malhar is itself an API to leverage business logic. Over past three years we have found that the cadence of release and api changes in Malhar is much rapid than Apex and it was operationally much easier to separate them into their own repos. Two repos will reflect clear separation of engine (Apex) and operators/business logic (Malhar). It will allow or independent release cycles (operator change independent of engine due to stable API). We however do not believe in two levels of committers. We believe there should be one community that works across both and innovates with ideas that Malhar and Apex combined provide the value proposition. We are proposing that Apache incubation process help us to foster development of one community (mailing list, committers), and a yet be ok with two repos. We are proposing that this be taken up during incubation. Community will learn if this works. The decision on whether to split them into two projects be taken after the learning curve during incubation.

Initial Committers

- Roma Ahuja (rahuja at directv dot com)
- Isha Arkatkar (isha at datatorrent dot com)
- Raja Ali (raji at silverspringnet dot com)
- Sunaina Chaudhary (SChaudhary at directv dot com)
- Bhupesh Chawda (bhupesh at datatorrent dot com)
- Chaitanya Chelobu (chaitanya at datatorrent dot com)
- Bright Chen (bright at datatorrent dot com)
- Pradeep Dalvi (pradeep dot dalvi at datatorrent dot com)
- Sandeep Deshmukh (sandeep at datatorrent dot com)
- Yogi Devendra (yogi at datatorrent dot com)
- Cem Ezberci (hasan dot ezberci at ge dot com)
- Timothy Farkas (tim at datatorrent dot com)
- Ilya Ganelin (ilya dot ganelin at capitalone dot com)
- Vitthal Gogate (vitthal_gogate at yahoo dot com)
- Parag Goradia (parag dot goradia at ge dot com)
- Tushar Gosavi (tushar at datatorrent dot com)
- Priyanka Gugale (priyanka at datatorrent dot com)
- Gaurav Gupta (gaurav at datatorrent dot com)
- Sandesh Hegde (sandesh at datatorrent dot com)
- Siyuan Hua (siyuan at datatorrent dot com)
- Ajith Joseph (ajoseph at silverspring dot com)
- Amol Kekre (amol at datatorrent dot com)
- Chinmay Kolhatkar (chinmay at datatorrent dot com)
- Pramod Immaneni (pramod at datatorrent dot com)
- Anuj Lal (anuj dot lal at ge dot com)
- Dongsu Lee (dlee3 at directv dot com)
- Vitaly Li (blossom dot valley at gmail dot com)
- Dean Lockgaard (dean at datatorrent dot com)
- Rohan Mehta (rohan_mehta at apple dot com)
- Adi Mishra (apmishra at directv dot com, adi dot mishra at gmail dot com)
- Chetan Narsude (chetan at datatorrent dot com)
- Darin Nee (dnee at silverspring dot com)
- Alexander Parfenov (sasha at datatorrent dot com)
- Andrew Perlitch (andy at datatorrent dot com)
- Shubham Phatak (shubham at datatorrent dot com)
- Ashwin Putta (ashwin at datatorrent dot com)
- Rikin Shah (shah_rikin at yahoo dot com)
- Luis Ramos (l dot ramos at ge dot com)
- Munagala Ramanath (ram at datatorrent dot com)
- Vlad Rozov (vlad dot rozov at datatorrent dot com)
- Atri Sharma (atri dot jiit at gmail dot com)
- Chandni Singh (chandni at datatorrent dot com)
- Venkatesh Sivasubramanian (venkateshs at ge dot com)
- Aniruddha Thombare (aniruddha at datatorrent dot com)
- Jessica Wang (jessica at datatorrent dot com)
- Thomas Weise (thomas at datatorrent dot com)
- David Yan (david at datatorrent dot com)
- Kevin Yang (yang dot k at ge dot com)
- Brennon York (brennon dot york at capitalone dot com)

Affiliations

- Apple: Vitaly Li, Rohan Mehta
- Barclays: Atri Sharma
- Class Software: Justin Mclean
- CapitalOne: Ilya Ganelin, Brennon York
- DataTorrent: everyone else on this proposal
- Datachief: Rikin Shah
- DirecTV: Roma Ahuja, Sunaina Chaudhary, Dongsu Lee, Adi Mishra
- E8security: Vitthal Gogate
- General Electric: Cem Ezberci, Parag Goradia, Anuj Lal, Luis Ramos, Venkatesh Sivasubramanian, Kevin Yang
- Hortonworks: Alan Gates, Taylor Goetz, Chris Nauroth, Hitesh Shah
- MapR: Ted Dunning

- SilverSpring Networks: Raja Ali, Ajith Joseph, Darin Nee

Sponsors

Champion

Ted Dunning

Nominated Mentors

The initial mentors are listed below:

- Ted Dunning - Apache Member, MapR
- Alan Gates - Apache Member, Hortonworks
- Taylor Goetz - Apache Member, Hortonworks
- Justin Mclean - Apache Member, Class Software
- Chris Nauroth - Apache Member, Hortonworks
- Hitesh Shah: Apache Member, Hortonworks

Sponsoring Entity

We would like to propose Apache incubator to sponsor this project.