

QuickstepProposal

Abstract

Quickstep is a high-performance database engine. It is designed to (1) convert data to insights at bare-metal speed, (2) support multiple query surfaces including SQL (the first, and current, version only supports SQL), and (3) deliver bare-metal performance on any hardware (including running on a laptop, running on a high-end (single node) server, and running on a distributed cluster). Since its inception, the project has been planned to deliver a high-performance single node system first, followed by a distributed system.

Quickstep is composed of several different modules that handle different concerns of a database system. The main modules are:

- Utility - Reusable general-purpose code that is used by many other modules.
- Threading - Provides a cross-platform abstraction for threads and synchronization primitives that abstract the underlying OS threading features.
- Types - The core type system used across all of Quickstep. Handles details of how SQL types are stored, parsed, serialized & deserialized, and converted. Also includes basic containers for typed values (tuples and column-vectors) and low-level operations that apply to typed values (e.g. basic arithmetic and comparisons).
- Catalog - Tracks database schema as well as physical storage information for relations (e.g. which physical blocks store a relation's data, and any physical partitioning and placement information).
- Storage - Physically stores relational data in self-contained, self-describing blocks, both in-memory and on persistent storage (disk or a distributed filesystem). Also includes some heavyweight run-time data structures used in query processing (e.g. hash tables for join and aggregation). Includes a buffer manager component for managing memory use and a file manager component that handles data persistence.
- Compression - Implements ordered dictionary compression. Several storage formats in the Storage module are capable of storing compressed column data and evaluating some expressions directly on compressed data without decompressing. The common code supporting compression is in this module.
- Expressions - Builds on the simple operations provided by the Types module to support arbitrarily complex expressions over data, including scalar expressions, predicates, and aggregate functions with and without grouping.
- Relational Operators - This module provides the building blocks for queries in Quickstep. A query is represented as a directed acyclic graph of relational operators, each of which is responsible for applying some relational-algebraic operation(s) to transform its input. Operators generate individual self-contained "work orders" that can be executed independently. Most operators are parallelism-friendly and generate one work-order per storage block of input.
- Query Execution - Handles the actual scheduling and execution of work from a query at runtime. The central class is the Foreman, an independent thread with a global view of the query plan and progress. The Foreman dispatches work-orders to stateless Worker threads and monitors their progress, and also coordinates streaming of partial results between producers and consumers in a query plan DAG to maximize parallelism. This module also includes the [QueryContext](#) class, which holds global shared state for an individual query and is designed to support easy serialization/deserialization for distributed execution.
- Parser - A simple SQL lexer and parser that parses SQL syntax into an abstract syntax tree for consumption by the Query Optimizer.
- Query Optimizer - Takes the abstract syntax tree generated by the parser and transforms it into a runnable query-plan DAG for the Query Execution module. The Query Optimizer is responsible for resolving references to relations and attributes in the query, checking it for semantic correctness, and applying optimizations (e.g. filter pushdown, column pruning, join ordering) as part of the transformation process.
- Command-Line Interface - An interactive SQL shell interface to Quickstep.

Quickstep is implemented in C++ and does not require many external libraries to run. Quickstep is currently an open source project licensed under the Apache License Version 2.0 and governed by a group of engineers at Pivotal.

Quickstep began in 2011 as a research project in the Computer Sciences Department at the University of Wisconsin <https://quickstep.cs.wisc.edu/> and the copyrights underlying the project was transferred to a company called Quickstep Technologies, which was acquired by Pivotal in 2015.

Proposal

The goal of this proposal is to bring an already existing open source project into the Apache Software Foundation (ASF) family thus leveraging a very successful "Apache Way" governance model in order to increase community participation and diversity. We hope that it will allow us to build a vibrant, diverse and self-governed open source community around the technology. Pivotal has agreed to transfer the brand name "Quickstep" to ASF and will stop using Quickstep to refer to this software if the project gets accepted into the ASF Incubator under the name of "Apache Quickstep (incubating)". Pivotal may market and sell products that include Apache Quickstep (incubating) under a different brand name, but no determination has been made regarding that. While Quickstep is our primary choice for a name of the project, in anticipation of any potential issues with PODLINGNAMESEARCH we have come up with two alternative names: (1) Bolero or (2) Hustle.

Pivotal is submitting this proposal to transfer the Quickstep source code and associated artifacts (documentation, web site content, wiki, etc.) from its current Github location to the ASF Incubator under the Apache License, Version 2.0 and is asking the Incubator PMC to establish an open source community.

Background

Quickstep is a next-generation relational data processing kernel currently being developed as a collaboration between the academic community and Pivotal. Quickstep aims to deliver efficient and sustainable data processing performance on current and future hardware by using a hardware-software co-design philosophy.

For the hardware available today, this means effectively exploiting large main memories, fast on-die CPU caches, highly parallel multi-core CPUs, and NVRAM storage technologies.

For the hardware available in the future, the project aims to co-design hardware and software primitives that will allow data processing kernels to work on increasing amounts of data economically – both from the raw performance perspective, and from the perspective of the energy consumed by data processing kernels.

Rationale

In the past decade, ASF has established itself as one of the quintessential sources of innovation in data management and data processing frameworks. At the same time, there is a clear need for a modern, flexible framework capable of exploiting the hardware characteristics of today and make it available as a set of building blocks to as wide a community of developers as possible. We strongly believe that Quickstep technology can benefit a broader ecosystem of database developers and researchers but this "world domination" needs to be achieved through a vibrant, diverse, self-governed community collectively innovating around a single codebase while at the same time cross-pollinating with various other data management communities. ASF is the ideal place to meet those ambitious goals. We also believe that our experience bringing various Pivotal data products into ASF family - including Apache Geode (incubating), Apache HAWQ (incubating) and Apache MADlib (incubating) can be leveraged to make the Quickstep transition a success, thus improving the chances of it becoming a truly vibrant Apache community.

Initial Goals

Our initial goals are to bring Quickstep into ASF, transition internal engineering processes into the open, and foster a collaborative development model according to the "Apache Way." Pivotal and its academic partners plan 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.

Current Status

Currently, the project code base is licensed under the Apache License v.2 and is available in a [GitHub](https://github.com/pivotalsoftware/quickstep) repository <https://github.com/pivotalsoftware/quickstep>. The documentation and wiki pages are available at same repository. Throughout its history Quickstep was developed in a hybrid closed/opens source mode but it has its roots in open source database management communities. The internal engineering practices adopted by the development team lend themselves well to an open, collaborative and meritocratic environment.

The Quickstep team has always focused on building a robust end user community of researchers. The existing documentation along with various publications are expected to facilitate conversions between our existing users so as to transform them into an active community of Quickstep members, stakeholders and developers.

Meritocracy

Our proposed list of initial committers include the current Quickstep R&D team and several existing academic partners. 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, contributors will earn merit. They also will be encouraged to provide non-code contributions (documentation, events, 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 Quickstep is accepted for incubation, the primary initial goal will be transitioning the core community towards embracing the Apache Way of project governance. We would solicit major existing contributors to become committers on the project from the start.

Core Developers

A small percentage of Quickstep core developers are skilled in working as part of openly governed Apache communities (mainly around the Hadoop ecosystem). That said, most of the core developers are currently NOT affiliated with the ASF and would require new ICLAs before committing to the project.

Alignment

The following existing ASF projects can be considered when reviewing the Quickstep proposal:

- Apache Hive: Potential alignment here is to consider a version of Hive that run on the Quickstep executor.
- Apache HAWQ (incubating): Potential alignment here is to consider exchanging ideas and/or code for execution across both systems.
- Apache YARN: Work has started on a distributed version of Quickstep, and its current path is to run as a YARN application.
- Apache Mesos: Potential alignment here is for Quickstep to run in Apache Mesos.

Known Risks

Development has been done mostly by a tightly knit group of University of Wisconsin researchers and later was sponsored mostly by a single company (Pivotal) thus far and coordinated mainly by the core Quickstep team. The Quickstep team now spans Pivotal and the University of Wisconsin.

For the project to fully transition to the Apache Way governance model, development must shift towards the meritocracy-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 Quickstep product are compatible with the ASF infrastructure and thus we do not anticipate any on-boarding pains.

The project went through a very thorough vetting as part of Pivotal open sourcing it under the Apache License v. 2.0 only a few month ago. This gives us reasonable confidence to conclude that the code base is clean and free from IP complications. Orphaned products Pivotal is fully committed to maintaining its position as one of the leading providers of database management and data processing solutions and the corresponding Pivotal commercial product will continue to be developed around the Quickstep project.

Moreover, Pivotal has a vested interest in making Quickstep successful by driving its close integration with both existing projects contributed to open source by Pivotal including Apache HAWQ (incubating) and Greenplum Database, and sister ASF projects. We expect this to further reduce the risk of orphaning the product.

Inexperience with Open Source

Pivotal has embraced open source software since its formation by employing contributors/committers and by shepherding open source projects like Cloud Foundry, Spring, RabbitMQ and MADlib. Individuals working at Pivotal have experience with the formation of vibrant communities around open technologies with the Cloud Foundry Foundation, and continuing with the creation of a community around Apache Geode (incubating), Apache HAWQ (incubating) and Apache MADlib (incubating). Although some of the initial committers have not had the experience of developing entirely open source, community-driven projects, we expect to bring to bear the open development practices that have proven successful on longstanding Pivotal open source projects to the Quickstep community. Additionally, several ASF veterans have agreed to mentor the project and are listed in this proposal. The project will rely on their collective guidance and wisdom to quickly transition the entire team of initial committers towards practicing the Apache Way.

Homogeneous Developers

While many of the initial committers are employed by Pivotal or at the University of Wisconsin, we have already seen a healthy level of interest from 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

Many of the contributors are paid to work in the Big Data and data processing space and nearly all are committed to a career in that space. While they might wander from their current employers, they are unlikely to venture far from their core expertise 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, Quickstep may consider various degrees of integration and code exchange with Apache Hive, Apache HAWQ (incubating), Apache YARN and Apache Mesos.

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 Quickstep into Apache Incubator.

Documentation

The documentation is currently available at <http://quickstep.cs.wisc.edu/>

Initial Source

Initial source code is currently licensed under Apache License v.2 and is available at <https://github.com/pivotalsoftware/quickstep>.

Source and Intellectual Property Submission Plan

As soon as Quickstep is approved to join the Incubator, the source code will be transitioned via an exhibit to Pivotal's current Software Grant Agreement onto ASF infrastructure. We know of no legal encumbrances inhibiting the transfer of source code to the ASF.

External Dependencies

Runtime dependencies:

- farmhash: <https://github.com/google/farmhash> [License: MIT]
- gflags: <https://github.com/gflags/gflags> [License: BSD]
- glog: <https://github.com/google/glog> [License: BSD]
- gperftools: <https://github.com/gperftools/gperftools> [License: BSD]

- linenoise: <https://github.com/antirez/linenoise> [License: BSD 2-Clause]
- protobuf: <https://github.com/google/protobuf> [License: BSD]

Build only dependencies:

- cmake: <https://cmake.org> [License: BSD]
- bison: <https://www.gnu.org/software/bison/> [License: GPL with exception for generated parsers]
- flex: <http://flex.sourceforge.net> [License: BSD]

Test only dependencies:

- benchmark: <https://github.com/google/benchmark> [License: Apache 2.0]
- cpplint: <https://github.com/google/styleguide> [License: BSD]
- gtest: <https://github.com/google/googletest> [License: BSD]
- iwyu: <http://include-what-you-use.org/> [License: UIUC BSD-Like]

Cryptography: N/A

Required Resources

Mailing lists

- private@quickstep.incubator.apache.org (moderated subscriptions)
- commits@quickstep.incubator.apache.org
- dev@quickstep.incubator.apache.org
- issues@quickstep.incubator.apache.org
- user@quickstep.incubator.apache.org

Git Repository

<https://git-wip-us.apache.org/repos/asf/incubator-quickstep.git>

Issue Tracking

JIRA Project QUICKSTEP (QUICKSTEP)

Other Resources

Means of setting up regular builds for Quickstep on builds.apache.org will require integration with Docker support.

Initial Committers

- Jignesh M. Patel
- Harshad Deshmukh
- Jianqiao Zhu
- Zuyu Zhang
- Marc Spehlmann
- Saket Saurabh
- Hakan Memisoglu
- Rogers Jeffrey Leo John
- Adalbert Gerald Soosai Raj
- Udip Pant
- Siddharth Suresh
- Rathijit Sen
- Craig Chasseur

- Qiang Zeng
- Shoban Chandrabose
- Navneet Potti
- Yinan Li
- Sangmin Shin
- James Paton
- Shixuan Fan
- Roman Shaposhnik
- Konstantin Boudnik
- Julian Hyde
- Dhruba Borthakur

Affiliations

- University of Wisconsin: Jignesh M. Patel, Harshad Deshmukh, Jianqiao Zhu, Zuyu Zhang, Marc Spehlmann, Saket Saurabh, Hakan Memisoglu, Rogers Jeffrey Leo John, Adalbert Gerald Soosai Raj, Udip Pant, Siddharth Suresh, Rathijit Sen, Navneet Potti, Shixuan Fan
- Pivotal: Roman Shaposhnik
- Google: Craig Chasseur
- Facebook: James Paton, Dhruba Borthakur
- Pinterest: Sangmin Shin
- Microsoft: Yinan Li
- Hortonworks: Julian Hyde
- Memcore: Konstantin Boudnik

Sponsors

Champion

Roman Shaposhnik

Nominated Mentors

The initial mentors are listed below:

- Konstantin Boudnik - Apache Member, Memcore
- Roman Shaposhnik - Apache Member, Pivotal
- Julian Hyde, IPMC Member, Hortonworks

Sponsoring Entity

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