

# PhoenixProposal

## Abstract

Phoenix is an open source SQL query engine for Apache HBase, a NoSQL data store. It is accessed as a JDBC driver and enables querying and managing HBase tables using SQL.

## Proposal

Phoenix is an open source SQL skin over HBase delivered as a client-embedded JDBC driver targeting low latency queries over HBase data. Phoenix takes your SQL query, compiles it into a series of HBase scans, and orchestrates the running of those scans to produce regular JDBC result sets. The table metadata is stored in an HBase table and versioned, such that snapshot queries over prior versions will automatically use the correct schema. Direct use of the HBase API, along with coprocessors and custom filters, results in performance on the order of milliseconds for small queries, or seconds for tens of millions of rows. Phoenix interfaces with both Pig and Map-reduce for the input and output of data.

## Background

Phoenix initially started as an internal project at Salesforce.com to efficiently analyze big data stored in HBase. It was open sourced on Github about a year ago in Jan 2013. Over time Phoenix, together with HBase as the storage tier, has begun to evolve into a general SQL database with support for metadata management, secondary indexes, joins, query optimization, and multi-tenancy. This is expected to continue as Phoenix implements a cost-based query optimizer and potentially transaction support, and surfaces new HBase security features such as encryption and cell-level security. Phoenix's developer community has also grown to include additional companies such as Intel, who have contributed join support to Phoenix, as well as Hortonworks, who are in the process of porting Phoenix to the 0.96 release of HBase.

## Rationale

As usage and the number of contributors to Phoenix has grown, we have sought for a long-term home for the project, and we believe the Apache foundation would be a great fit. Joining Apache would ensure that tried and true processes and procedures are in place for the growing number of organizations interested in contributing to Phoenix. Phoenix is also a good fit for the Apache foundation: Phoenix already interoperates with several existing Apache projects (HBase, Hadoop, Pig, [BigTop](#)). The Phoenix team is familiar with the Apache process and and believes in the Apache mission - the team already includes multiple Apache committers.

## Initial Goals

The initial goals will be to move the existing codebase to Apache and integrate with the Apache development process. Once this is accomplished, we plan for incremental development and releases that follow the Apache guidelines.

## Current Status

Phoenix has undergone two major and three minor releases (1.0, 1.1, 1.2, 2.0, and 2.1) as well as many patch releases. Phoenix is being used in production by Salesforce.com as well as at other organizations. The Phoenix codebase is currently hosted at github.com, which will form the basis of the Apache git repository.

## Meritocracy

The Phoenix project already operates on meritocratic principles. Phoenix has several developers from various organizations outside of Salesforce.com who have contributed major new features. While this process has remained mostly informal, as we do not have an official committer list, an implicit organization exists in which individuals who contribute major components act as maintainers for those modules. If accepted, the Phoenix project would include several of these participants as initial committers. We will work to identify all committers and PPMC members for the project and to operate under the ASF meritocratic principles.

## Community

Acceptance into the Apache foundation would bolster the already strong user and developer community around Phoenix. That community includes many contributors from various other companies, and an active mailing list composed of hundreds of users.

## Core Developers

The core developers of our project are listed in our contributors and initial PPMC below. Though many are employed at Salesforce.com, there is a representative cross sampling of other organizations including Intel, Hortonworks, and Cloudera.

## Alignment

Our proposed Phoenix effort aligns closely with Apache HBase. The HBase project perimeter is denoted by a simple byte-array based Create, Read, Update, Delete and Scan APIs with no current plans to extend beyond this bounds. Phoenix complements this with a higher level API in SQL with which many are already familiar. At first glance, it may seem that Phoenix should just be folded into HBase as a new module. However, the focus of the two projects will be quite different, especially as Phoenix matures. With secondary indexing and joins just having been introduced into Phoenix, the next big frontier will be to implement a cost-based query optimizer. This is the heart-and-soul of most relational databases and can take a lifetime to get right.

HBase is focused on being a scalable data store agnostic to types and schema. Phoenix would layer typing, and relational facilities on top of this scalable store. By keeping Apache HBase and Phoenix separate, both may evolve independently and at different rates. Though the focus of the two projects is different, the relationship between them is very positive and mutually beneficial. New features in HBase will be leveraged in Phoenix as it makes sense to surface these in a SQL paradigm. In addition, Phoenix may drive new features in HBase, as evidenced by the new type system recently introduced into HBase. This will enable better interoperability between Apache Hive, standalone HBase uses case, and Phoenix by defining a standard serialization format.

Phoenix can be divided into a front end and a back end. The front end is delivered as a JDBC driver and contains, among other things, the SQL parser and query planner. The front end is currently written for the HBase client API but could be extended to support other data stores in the Apache family.

The back end is, currently, HBase specific components for pushing as much work to the server as possible. However, if there were sufficient interest to build them, contributions to Phoenix of new back ends for other data stores in the Apache family would be feasible.

Other projects exist that perform SQL over HBase data (such as Apache Hive), however these products do not provide the same low latency query capabilities as Phoenix. Instead, they are more oriented around maximizing throughput for batched operations. Phoenix opens the door to a completely new set of use cases for Apache HBase that demand a more interactive user experience.

There are also a number of related Apache projects and dependencies that are mentioned in the Relationships with Other Apache products section.

## Known Risks

### Orphaned Products

Given the current level of investment in Phoenix - the risk of the project being abandoned is minimal. All current and planned HBase use cases at Salesforce.com go through Phoenix. In addition, both Intel and Hortonworks plan to include Phoenix in their distributions. Other companies have devoted significant internal infrastructure investment in Phoenix.

### Inexperience with Open Source

Phoenix has existed as a healthy open source project for almost a year. During that time, James, Mujtaba, and others have successfully fostered an open-source community, attracting users and developers from a diverse group of companies including Intel, Intuit, Bloomberg, Tagged, and Hortonworks. Although neither are committers on other Apache projects, both James and Mujtaba have experience working with and contributing to other Apache projects.

### Homogenous Developers

The initial list of committers includes developers from several institutions, including Salesforce, Intel, and Hortonworks.

### Reliance on Salaried Developers

Like most open source projects, Phoenix receives substantial support from salaried developers. A large fraction of Phoenix development is supported by Salesforce.com. In addition, those working from within corporations and universities often devote "after hours" or spare time to the project. We will continue our efforts to ensure stewardship of the project to be independent of salaried developers.

### Relationship with Other Apache Products

Although Phoenix provides a higher level abstraction than Apache HBase by hiding its client APIs, Phoenix relies on Apache HBase for both storing and retrieving data. It also inter-operates with Apache HBase by allowing existing data, not created by Phoenix, to be queried. In addition, both Apache Pig and Hadoop are supported for data input and output. Finally, the Phoenix is included and installable through Apache Bigtop and the build and test suite are run through Apache Maven.

Phoenix offers an alternative query engine to Apache Hadoop (MapReduce). Unlike [MapReduce](#), Phoenix is designed for lower-latency, OLTP, and interactive workloads. This makes the projects complimentary as users may run [MapReduce](#) and Phoenix side-by-side.

We plan to increase the interoperability between Phoenix, Apache Hive, and standalone Apache HBase usage by standardizing on a new type system that has been introduced in the current major release of HBase. By all these products adopting this new serialization format, interoperability between them will take a big step forward.

In addition, we plan to explore providing lower level APIs for other products such as Apache Drill to plug into when querying HBase data so that they get the performance benefits of using Phoenix.

### A Excessive Fascination with the Apache Brand

Phoenix is already a healthy and relatively well known open source project. This proposal is not for the purpose of generating publicity. Rather, the primary benefits to joining Apache are those outlined in the Rationale section.

### Documentation

Additional documentation on Phoenix may be found on its github website:

- Phoenix overview: <https://github.com/forcedotcom/phoenix/blob/master/README.md>
- Phoenix wiki: <https://github.com/forcedotcom/phoenix/wiki>
- Phoenix road map: <https://github.com/forcedotcom/phoenix/wiki#roadmap>
- Phoenix issue tracking: <https://github.com/forcedotcom/phoenix/issues?direction=desc&sort=updated&state=open>
- Phoenix codebase: <https://github.com/forcedotcom/phoenix>
- Phoenix SQL language reference: <http://forcedotcom.github.io/phoenix/>
- Phoenix performance: <https://github.com/forcedotcom/phoenix/wiki/Performance#phoenix-vs-related-products>
- User group: <https://groups.google.com/group/phoenix-hbase-user>

## Initial Source

The Phoenix codebase is currently hosted on Github: <https://github.com/forcedotcom/phoenix>.

## Source and Intellectual Property Submission Plan

Currently, the Phoenix codebase is distributed under a BSD license. Upon entering Apache, the Phoenix license will be migrated to the Apache 2.0 License.

## External Dependencies

Beyond relying on Apache HBase, Phoenix has the following external dependencies:

- ANTLR 3.5 (BSD license: <http://www.antlr3.org/license.html>)
- Sqlline 1.1.2 (BSD license: <https://github.com/julianhyde/sqlline/blob/master/LICENSE>)
- Open CSV 2.3 (Apache 2.0 license)

Upon acceptance to the incubator, we would begin a thorough analysis of all transitive dependencies to verify this information and introduce license checking into the build and release process by integrating with Apache Rat.

## Required Resources

### Mailing list

We will migrate the existing Phoenix mailing lists as follows:

- phoenix-hbase-user@googlegroups.com --> users@phoenix.incubator.apache.org
- phoenix-hbase-dev@googlegroups.com --> dev@phoenix.incubator.apache.org
- private@phoenix.incubator.apache.org for IPMC members
- commits@phoenix.incubator.apache.org

The latter is to be consistent with the new PIAO naming scheme for podlings.

### Source control

The Phoenix team would like to use Git for source control, due to our current use of Git. We request a writeable Git repo for Phoenix, and mirroring to be set up to Github through INFRA.

### Issue Tracking

Phoenix currently uses the github issue tracking system associated with its github repo: <https://github.com/forcedotcom/phoenix/issues?direction=desc&sort=updated&state=open>. We will migrate to the Apache JIRA: <http://issues.apache.org/jira/browse/PHOENIX>

### Other Resources

- Jenkins/Hudson for builds and test running.
- Wiki for documentation purposes
- Blog to improve project dissemination

## Initial Committers

- James Taylor <jtaylor at salesforce dot com>
- Mujtaba Chohan <mchohan at salesforce dot com>
- Jesse Yates <jyates at apache dot org>
- Eli Levine <elevine at salesforce dot com>
- Simon Toens <stoens at salesforce dot com>
- Maryann Xue <wei.xue at intel dot com>
- Anoop Sam John <anoopsamjohn at apache dot org>
- Ramkrishna S Vasudevan <ramkrishna at apache dot org>

- Jeffrey Zhong <jeffreyz at apache dot org>
- Nick Dimiduk <ndimiduk at apache dot org>

## Affiliations

The initial committers are from three organizations: Salesforce.com, Intel, and Hortonworks.

- James Taylor (Salesforce.com)
- Mujtaba Chohan (Salesforce.com)
- Jesse Yates (Salesforce.com)
- Eli Levine (Salesforce.com)
- Simon Toens (Salesforce.com)
- Maryann Xue (Intel)
- Anoop Sam John (Intel)
- Ramkrishna S Vasudevan (Intel)
- Jeffrey Zhong (Hortonworks)
- Nick Dimiduk (Hortonworks)

## Sponsors

### Champion

- Michael Stack

### Nominated Mentors

- Michael Stack
- Lars Hofhansl
- Andrew Purtell
- Devaraj Das
- Enis Soztutar
- Steven Noels

### Sponsoring Entity

The Apache Incubator