# PigProposal

## Proposal for Pig Project

### Abstract

Pig is a platform for analyzing large data sets.

### Proposal

The Pig project consists of high-level languages for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

At the present time, Pig's infrastructure layer consists of a compiler that produces sequences of Map-Reduce programs, for which large-scale parallel implementations already exist (e.g., the Hadoop subproject). Pig's language layer currently consists of a textual language called Pig Latin, which has the following key properties:

1. *Ease of programming*. It is trivial to achieve parallel execution of simple, "embarrassingly parallel" data analysis tasks. Complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, making them easy to write, understand, and maintain. 2. *Optimization opportunities*. The way in which tasks are encoded permits the system to optimize their execution automatically, allowing the user to focus on semantics rather than efficiency. 3. *Extensibility*. Users can create their own functions to do special-purpose processing.

### Background

Pig started as a research project at Yahoo! in May of 2006 to combine ideas in parallel databases and distributed computing. The first internal release took place in July 2006. The first release was a simple front-end to the Hadoop Map/Reduce framework. The following releases added new features and evolved the language based on user feedback. In July 2007, pig was taken over by a development team and the first production version is due to be released on 9/28/07.

Since its inception, we had observed a steady growth of the user community within Yahoo!. In April 2007, Pig was released under a BSD-type license. Several external parties are using this version and have expressed interest in collaborating on its development.

### Rationale

In an information-centric world, innovation is driven by ad-hoc analysis of large data sets. For example, search engine companies routinely deploy and refine services based on analyzing the recorded behavior of users, publishers, and advertisers. The rate of innovation depends on the efficiency with which data can be analyzed.

To analyze large data sets efficiently, one needs parallelism. The cheapest and most scalable form of parallelism is cluster computing. Unfortunately, programming for a cluster computing environment is difficult and time-consuming. Pig makes it easy to harness the power of cluster computing for ad-hoc data analysis.

While other language exist that try to achieve the same goals, we believe that Pig provides more flexibility and gives more control to the end user.

SQL typically requires (1) importing data from a user's preferred format into a database system's internal format (2) well-structured, normalized data with a declared schema, and (3) programs expressed in declarative SELECT-FROM-WHERE blocks. In contrast, Pig Latin facilitates (1) interoperability, i.e. data may be read/written in a format accepted by other applications such as text editors or graph generators (2) flexibility, i.e. data may be loosely structured or have structure that is defined operationally, and (3) adoption by programmers who find procedural programming more natural than declarative programming.

Sawzall is a scripting language used at Google on top of Map-Reduce. A sawzall program has a fairly rigid structure consisting of a filtering phase (the map step) followed by an aggregation phase (the reduce step). Furthermore, only the filtering phase can be written by the user, and only a pre-built set of aggregations are available (new ones are non-trivial to add). While Pig Latin has similar higher level primitives like filtering and aggregation, an arbitrary number of them can be flexibly chained together in a Pig Latin program, and all primitives can use user-defined functions with equal ease. Further, Pig Latin has additional primitives such as cogrouping, that allow operations such as joins (which require multiple programs in Sawzall) to be written in a single line in Pig Latin. Further, Pig Latin is designed to be embedded into other languages, and can use functions written in other languages. Thus, in contrast to Sawzall, it directly caters to a large community of developers without having to make them learn an entirely new programming language.

### Current Status

#### Meritocracy

Pig was started as a project that was developed by Yahoo! research team. Recently we have added a development team that works in harmony with the research team with both teams actively and successfully contributing to the project. We are planning to create the environment that encourages meritocracy and is consistent with the meritocracy principles of Apache. Within the team we have people actively participating in the Hadoop subproject.

#### Community

Pig has an active user community within Yahoo! that has been steadily growing. Pig also attracted external users since its release under a BSD-type license. Several external parties are using the product and have expressed interest in collaborating on its development.

Also, since the current version of Pig is built on top of the Hadoop we believe that we will be able to quickly extend our community by attracting both the Hadoop users and developers to the project.

### Core Developers

Our contributors come from both research and development world and most have background in database internals and large scale distributed systems.

### Alignment

Yahoo! seeks to develop Pig collaboratively with others, not to control and maintain it independently. Apache offers the best legal and social framework for such community-based software development.

Also, the current version of Pig runs on top of the Hadoop's Map-Reduce infrastructure which is part of Apache. We believe there would be a lot of synergy between the projects both in terms of users and developers.

## Known Risks

### Orphaned products

All current contributors are part of Yahoo which is a major player in the space and is committed to grid computing. Also we expect high degree of synergy with Hadoop subproject.

### Inexperience with Open Source

Two of the committers have extensive experience with open source and Apache. The rest are new to open source and will be guided through the process by the team members with experience.

### Homogenous Developers

The current list of committers is confined to Yahoo employees. Our plan is to recruit more committers once the project gets on the way.

### Reliance on Salaried Developers

Currently, all contributors are Yahoo employees. By extending the development community we are hoping to mitigate this risk.

### Relationships with Other Apache Products

Pig is built on top of Hadoop and we expect deep collaboration with Hadoop subproject.

### An Excessive Fascination with the Apache Brand

Yahoo already have a strong brand and is not interested in Apache as a way to gain visibility. Yahoo! seeks to develop Pig collaboratively with others, not to control and maintain it independently. Apache offers the best legal and social framework for such community-based software development.

## Documentation

http://research.yahoo.com/project/pig

## Initial Source

The initial source will be donated by Yahoo Inc. The donating company will contribute the initial code base once the proposal is accepted and necessary infrastructure has been set up.

## External Dependencies

1. bzip2: http://www.kohsuke.org/bzip2/:Apache license 2. javacc: https://javacc.dev.java.net/:BSD license 3. hadoop: http://lucene.apache.org/hadoop/:Apache license 4. log4j: http://logging.apache.org/log4j/: Apache license 5. jsch: http://www.jcraft.com/jsch: BSD style license: http://www.jcraft.com/jsch/LICENSE.txt

## Required Resources

## Mailing lists

We would need the following mailing lists

1. pig-private (with moderated subscriptions) 2. pig-dev 3. pig-commits 4. pig-user

### Subversion Directory

https://svn.apache.org/repos/asf/incubator/pig

### Issue Tracking

JIRA PIG (PIG)

## Initial Committers

1. Nigel Daley (ndaley@yahoo-inc.com) 2. Alan Gates (gates@yahoo-inc.com) 3. Olga Natkovich (olgan@yahoo-inc.com) 4. Chris Olston (olston@yahoo-inc.com) 5. Owen O'Malley (oom@yahoo-inc.com) 6. Ben Reed (breed@yahoo-inc.com) 7. Utkarsh Srivastava (utkarsh@yahoo-inc.com)

## Affiliation

All initial committers are affiliated with Yahoo!

## Sponsors

### Champion

Doug Cutting

### Nominated Mentors

1. Doug Cutting 2. Torsten Curdt 3. Bertrand Delacretaz 4. Yoav Shapira 5. Sylvain Wallez

### Sponsoring Entity

Incubator