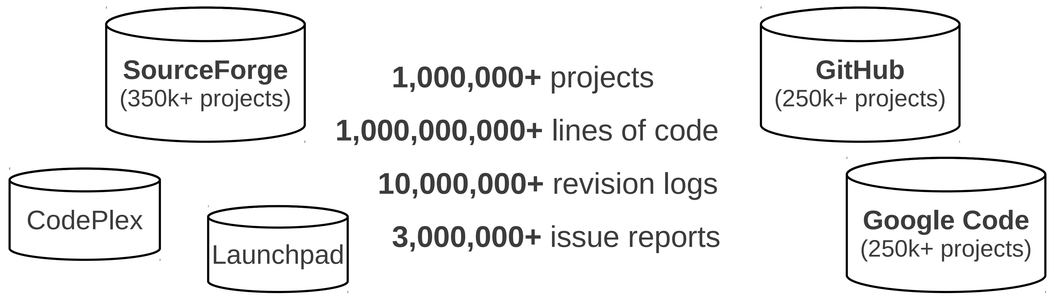
Boa: Analyzing Ultra-Large-Scale Software Repositories

For more information about the Boa project contact Hridesh Rajan at hridesh@iastate.edu

Why mine software repositories?

- Understand software development and evolution
- Improve software design and reuse
- Use knowledge to plan future development
- Make predictions about software development
- Empirically validate research ideas

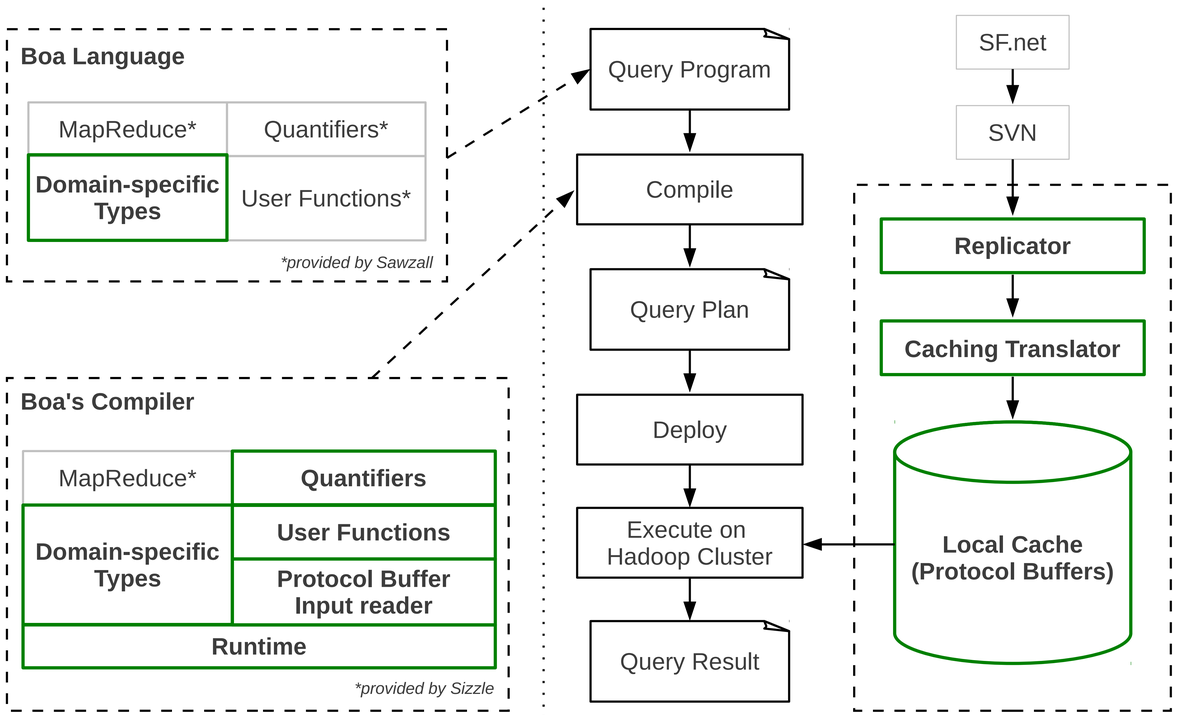
Why mine open source repositories?



1,000,000+ projects
1,000,000,000+ lines of code
10,000,000+ revision logs
3,000,000+ issue reports

Boa Goals

- Easy to use
- Abstract details of **how** to mine software repositories
- Efficient and scalable
- Reproducible research results



Domain-specific Types

Project	id, name, homepage_url, maintainers, code_repositories, etc
CodeRepository	url, repository_type, revisions
Revision	id, author, committer, commit_date, log, files
File	name
Person	email, real_name, username

Output Aggregators

mean	An arithmetic mean of all data.
sum	An arithmetic sum of all data.
top(N)	A sample that records the top N elements.
maximum(N)	A sample of the N highest weighted elements.
minimum(N)	A sample of the N lowest weighted elements.
set(N)	A set of at most N unique elements.
collection	Collection of all data.

Data Infrastructure

- Replicates the data offline, for faster access
- Translates the data into a common format
- Provides mapping from raw data to domain-specific types in the language

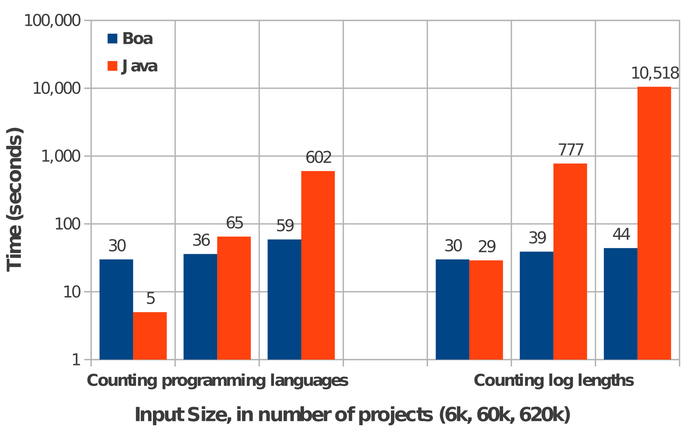
Runtime

- Runs on Hadoop MapReduce framework

Future work

- Other repositories (GitHub, etc)
- Other VCS (cvs, git, bzr, etc)

[Sizzle] A. Urso. https://github.com/anthonyu/Sizzle
[Sawzall] R. Pike et al. Interpreting the data: Parallel analysis with Sawzall. Sci. Program., 13(4), 2005

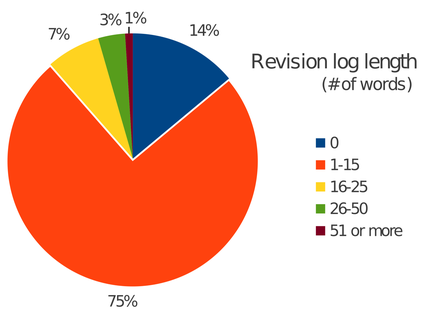


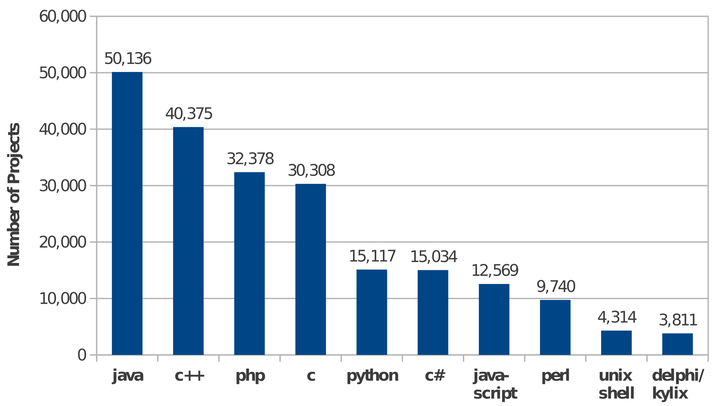
The 10 most used programming languages

```
counts: output top(10) of string weight int;
p: Project = input;

when (i: each int; def(p.programming_languages[i]))
  counts << p.programming_languages[i] weight 1;
```

Number of words in commit log messages

```
counts: output sum[int] of int;
p: Project = input;

when (i: each int; def(p.code_repositories[i]))
  when (j: each int; def(p.code_repositories[i].revisions[j]))
    counts[len(splitall(p.code_repositories[i].revisions[j].log, `\s+`))] << 1;
```





Robert Dyer
Hridesh Rajan
Hoan Nguyen
Tien N. Nguyen
IOWA STATE UNIVERSITY