Software Engineering
Collaboratories as a Service

Dr. Robert Dyer
Bowling Green State University
Open Access Publications

Open Source Software

Open Data

Open Methodology

Open Science
Replicating MSR results

- Robles, MSR'10
- 2/154 experimental papers "replication friendly"
- 48 due to lack of published data
ICSE 2019 Technical Track

• 38 papers badged Available

• 32 papers badged Reusable

• 1 paper badged Reproduced
Collaboratories can help!
What is a Collaboratory?
Collab(orate) + laboratory
“center without walls, in which the nation’s researchers can perform their research without regard to physical location, interacting with colleagues, accessing instrumentation, sharing data and computational resources”

- (Wulf, 1989)
Software Engineering Collaboratories (SECLabs)

ESEC/FSE 2018 NIER
Software Engineering Collaboratory (SECLab)
Let’s see what a SECLab might look like...
Consider a task to answer

"How many bug fixes add checks for null?"
Find null checks in each source

mine source code

Output count of all null checks

foreach project

Find all Java source files

Fixes bug?

Yes

Fixes

mine revisions

Access repository

No

Is fork?
A solution in Java...

```java
class AddNullCheck {
    static void main(String[] args) {
        /* create and submit a Hadoop job */
    }

    static class AddNullCheckMapper extends Mapper<Text, BytesWritable, Text, LongWritable> {
        static class DefaultVisitor {
            /* define default tree traversal */
            void map(Text key, BytesWritable value, Context context) {
                final Project p = /* read from input */
                new DefaultVisitor() {
                    boolean preVisit(Expression e) {
                        if (e.kind == ExpressionKind.EQ || e.kind == ExpressionKind.NEQ)
                            for (Expression exp : e.expressions)
                                if (exp.kind == ExpressionKind.LITERAL && exp.literal.equals("null")) {
                                    context.write(new Text("count"), new LongWritable(1));
                                    break;
                                }
                }.visit(p);
            }
        }
    }

    static class AddNullCheckReducer extends Reducer<Text, LongWritable, Text, LongWritable> {
        void reduce(Text key, Iterable<LongWritable> vals, Context context) {
            int sum = 0;
            for (LongWritable value : vals)
                sum += value.get();
            context.write(key, new LongWritable(sum));
        }
    }
}
```

Full program
over 140 lines of code

Uses JSON, SVN, and Eclipse JDT libraries

Uses Hadoop framework

Explicit/manual parallelization
A better solution...

\[ p: \text{Project} = \text{input}; \]
\[ \text{count: output sum of int;} \]

\[ \text{visit}(p, \text{visitor} \{ \}
\[ \hspace{1em} \text{before e: Expression ->} \]
\[ \hspace{2em} \text{if (e.kind == ExpressionKind.EQ || e.kind == ExpressionKind.NEQ)} \]
\[ \hspace{3em} \text{exists (i: int; isliteral(e.expressions[i], "null"))} \]
\[ \hspace{4em} \text{count} \ll 1; \]
\[ \}) \];

Full program \textbf{8 lines of code}!

\[ \textbf{Automatically parallelized!} \]

\textbf{No external libraries} needed!

\[ \text{Analyzes \textbf{28.8 million} source files in about \textbf{15 minutes}!} \]
\[ \text{only 32 \textit{micro}seconds each!}\]
The Boa language and data-intensive infrastructure

http://boa.cs.iastate.edu/
The research and educational activities described in this talk was supported in part by the US National Science Foundation (NSF) under grants CNS-15-12947, CNS-15-13263, CCF-13-49153.
Boa’s Architecture

Boa Language
- MapReduce
- Domain-specific Types
- Visitors

Boa’s Compiler
- MapReduce
- Domain-specific Types
- Cached Data input reader
- Runtime
- Quantifiers
- User Functions
- Visitors

Boa’s Data Infrastructure
- SF.net
  - Replicator
  - Caching Translator
- Local Cache

1 Pike et al, Scientific Prog. Journal, Vol 13, No 4, 2005
Visitor Syntax

id := visitor {
    before id:T -> statement
    after  id:T -> statement
    ...
};

visit(startNode, id);

Execute statement either before or after visiting the children of a node of type T
Support for Program Analysis Features

- **Control-Flow Graphs**
  \[ g := \text{getcfg}(m); \]

- **Control-Dependence Graphs**
  \[ g := \text{getcdg}(m); \]

- **Data-Dependence Graphs**
  \[ g := \text{getddg}(m); \]

- **Program-Dependence Graphs**
  \[ g := \text{getpdg}(m); \]
int x = 3

int y = x + 2

int i = 0

i < y

q < y

y = y - 1

i++

"f".substring(q, 1)
Graph Traversals

- Supports complex program analysis using data-flow

```plaintext
t := traversal (CFGNode) : int {
    ... # can call getValue() on nodes
    return ..; # sets the value on this node
}

traverse(cfg, TraversalDirection.FORWARD, TraversalKind.DFS, t, fixp);
```
int x = 3
int y = x + 2
int i = 0

i < y
q < y

"f".substring(q, 1)
y = y - 1
i++
Available in the next release...
Recently extended to enable more re-use

UPCOMING FEATURE: materialized views
Example Analysis

**RQ.** Find the buggy files by examining past bug fixing behavior.
Using shared results

```java
view Filter {
    Retained: output collection[pid: string] of int;

    v := J12345/AverageFRC;
    ...
}

o: output top(5)[pid: string] of fileName: string weight int;

if (len(Filter/Retained[input.id]) > 0) {
    v: output sum[file: string] of int =
    @rdyer/FixingRevision/FixFileCount[p.id];
    r: v._row;
    while (v >> r)
        o[input.id] << r.file weight r._2;
}
```
Over 1,000 users
Collaboratories as a Service (CaaS)

ESEC/FSE 2018 NIER
Collaboratories as a Service (CaaS)

Static Analysis
- SA1
- SA2
- SA3
- SA4

Points-to Analysis

Constant Propagation

Testing
- T1
- T2
- T3
- T4

Regression Testing

New Research Area
- N1
- N2
- N3
- N4

Static Analysis
- Points-to Analysis
- Constant Propagation
- Testing
- New Research Area

Regression Testing
Future Work

- What data/software/artifacts get published?

- Who vets it? How?

- Does every SECLab have to use the same interface?

- Do they all handle every step of a SECLab (e.g., cloning projects)?
Program Analysis Collaboratory (PAClab)

http://paclab.cs.bgsu.edu/
The research and educational activities described in this talk were supported in part by the US National Science Foundation (NSF) under grants CNS-18-23294, CNS-18-23357.
Evidence-based SE Research

• Research papers require substantial empirical evaluation
• Finding suitable programs is difficult
  • Need to compile
  • Need integer operations
  • etc
• Often very small set of benchmark programs
• Manual process to modify them so the tool works on them
• Old benchmarks are re-used due to the effort required
1. Find candidate projects
   Boa, NJR, SIR, GHTorrent, etc.

2. Clone candidate repositories
   GitHub etc.

3. Transform candidates based on specifications
   Docker

4. (optional) Run analysis on program variants using Docker

Initiate request via website
(send selection criteria + transformations, and (optional) Docker image URL)

Download results/program variants