BugZoo:
A Platform for Studying Software Bugs

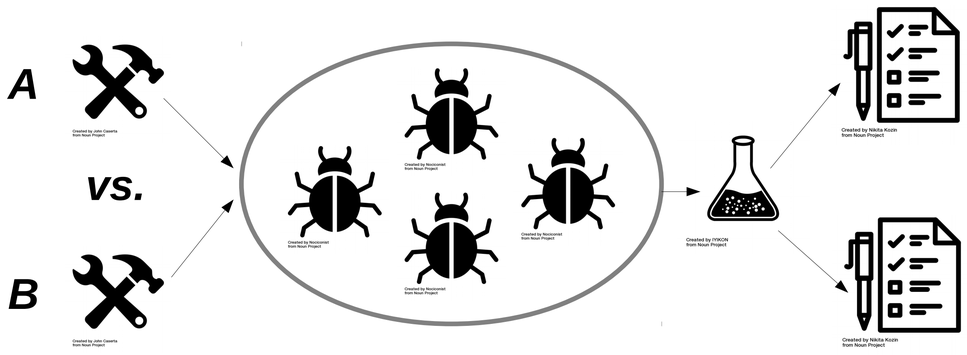
Christopher Steven Timperley,[1] Susan Stepney,[2] Claire Le Goues[1]

[1] Institute for Software Research, Carnegie Mellon University; [2] Department of Computer Science, University of York

What is BugZoo?

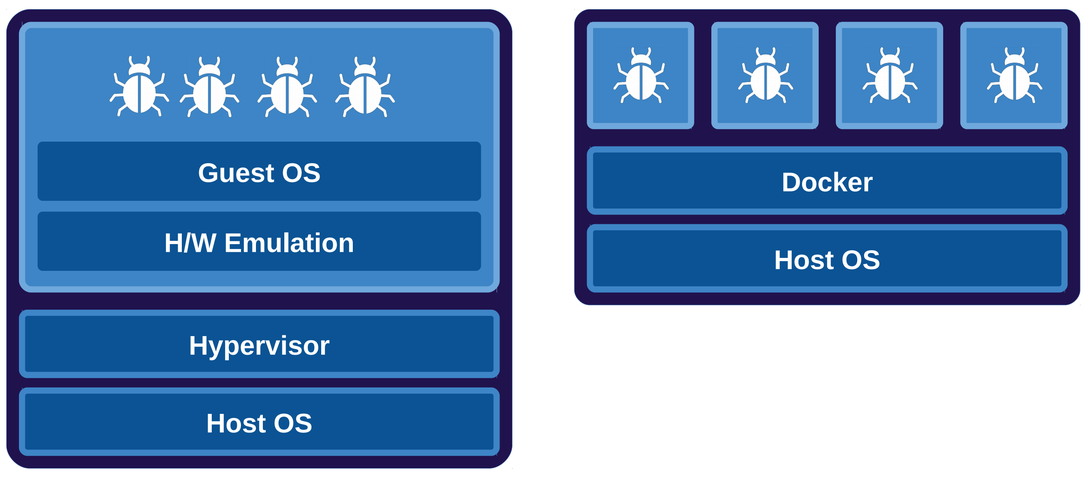
BugZoo is a decentralised platform for distributing, reproducing, and interacting with historical software bugs. BugZoo connects existing datasets and tools to researchers and developers, and provides a controlled environment for safely conducting reproducible experiments.

Performing a good experiment requires both an indicative dataset, as well as a controlled environment for reproducibily interacting with the bugs in that dataset. Datasets of historical bugs (e.g., ManyBugs [1], Defects4J [2], SIR [3]) are commonly used to conduct a fair evaluation of the effectiveness of new techniques for solving particular problems. BugZoo provides an open platform for conducting high-quality experiments.



How does BugZoo work?

The current best practice for conducting empirical evaluations [1] is to use a monolithic virtual machine to provide a single environment for evaluating a fixed set of tools against a dataset of bugs.



Instead, BugZoo separates bugs from the tools that are used to analyse them. Each bug and tool is provided by its own minimal Docker container image responsible for reproducing the bug or providing a barebones environment for the tool. The BugZoo platform automatically takes care of safely composing bugs and tools at run-time, avoiding “DLL hell” and allowing researchers to modify and extend existing experiments.

	Monolithic VMs	BugZoo
Speed	Slow	Close to native performance
Size	Tens of GBs	Hundreds of MBs
Extensibility	Rigid: difficult to add new tools and bugs.	Flexible: new tools and bugs can be added with minimal effort.

References

[1] C Le Goues, N Holtschulte, E K Smith, Y Brun, P Devanbu, S Forrest,and W Weimer. 2015. The ManyBugs and IntroClass Benchmarks for Automated Repair of C Programs. *Transactions on Software Engineering* 41, 12 (Dec. 2015), 1236–1256.

[2] R Just, D Jalali, and M D Ernst. 2014. Defects4J: A Database of ExistingFaults to Enable Controlled Testing Studies for Java Programs. In Proceedings of the 2014 *International Symposium on Software Testing and Analysis* (ISSTA ’14). ACM, New York, NY, USA, 437–440.

[3] H Do, S Elbaum, and G Rothermel. 2005. Supporting Controlled Experimentation with Testing Techniques: An Infrastructure and its Potential Impact. *Empirical Software Engineering* 10, 4 (1 Oct. 2005),405–435.

How can I use BugZoo?

BugZoo provides a range of interfaces, suited to different use cases.

Command-Line Interface

Allows users to easily download and build datasets and tools provided by remote Git repositories.

- add new datasets and tools
- download or build bugs
- interact with bugs
- run simple experiments
- inspect code coverage

```
$ bugzoo -h


 ____                  _____
|  _ \                |___  /
| |_) |_   _  __ _      / / ___   ___
|  _ <| | | |/ _` |    / / / _ \ / _ \
| |_) | |_| | (_| |   / /_| (_) | (_) |
|____/ \__,_|\__, |  /_____\___/ \___/
              __/ |
             |___/


A platform for studying software bugs

https://github.com/squaresLab/BugZoo
```

```
GET     /bugs
GET     /bugs/:id
POST    /bugs/:id/build
POST    /bugs/:id/download
GET     /containers
GET     /containers/:id
DELETE  /containers/:id
PATCH   /containers/:id
```

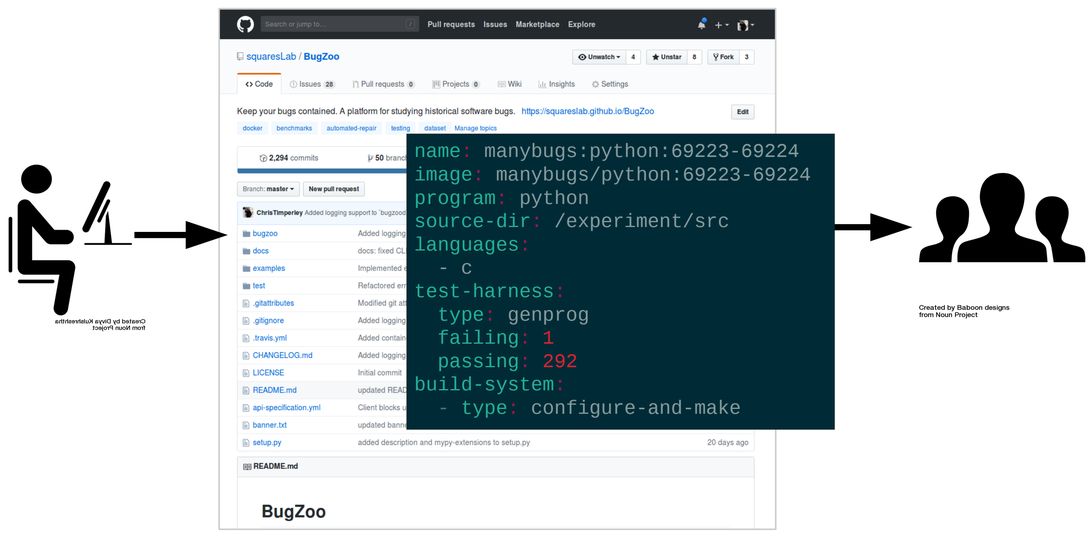
REST-like API

Allows developers to safely perform structured interaction with historical bugs. Comes with Python bindings.

- apply patches
- execute specific tests
- collect coverage
- compile with custom flags
- and more

How can I extend BugZoo?

Developers and curators can connect their existing tools and datasets to the BugZoo platform by simply adding a small number of manifest files to the Git repository for their tool/dataset.



BugZoo’s decentralised architecture lets users maintain ownership and control over their artifacts, and avoids the bottlenecks and scalability issues associated with a centralised approach.

BugZoo is open-source. Download and learn more at:

https://github.com/squaresLab/BugZoo

Images taken from the Noun Project:
“bug” by Nociconist, “tools” by John Caserta,
“users” by Baboon designs, “experiment” by IYIKON,
“checklist” by Nikita Kozin, “tool” by John Caserta

Corresponding Author:
Christopher Timperley
ctimperley@cmu.edu
https://christimperley.co.uk