Distributed Testing via Serverless Functions

Sanat Deshpande, Lionel Eisenberg, Jon Karyo, James Lubowsky
Johns Hopkins University Department of Computer Science

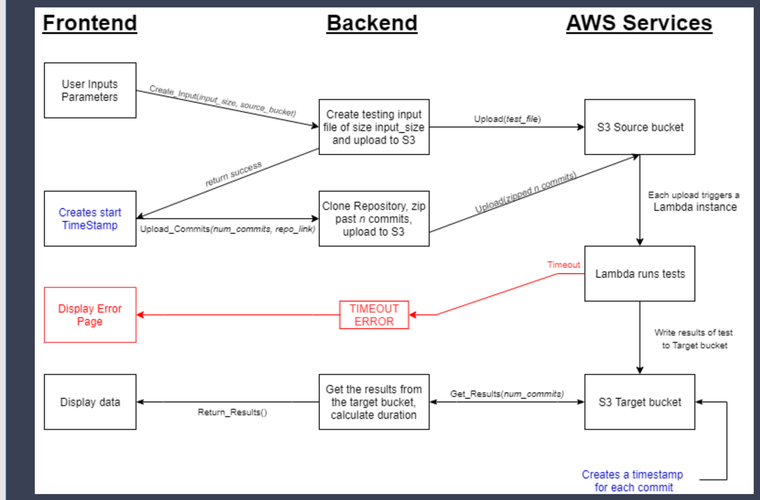
1. Background

- Standard practice on large projects is to maintain a suite of tests that ensure code functionality[1]
- Difficult to detect provenance of failures when new tests are added to large code repositories
- Some tools exist to determine the last failing commit, such as git bisect[2]
  - git bisect can be automated to remove human intervention
- Limitation of *O(k log(n))* performance
  - *k* is the runtime of a test
  - *n* is the number of commits
- Provides poor feedback as to which tests failed

2. Questions

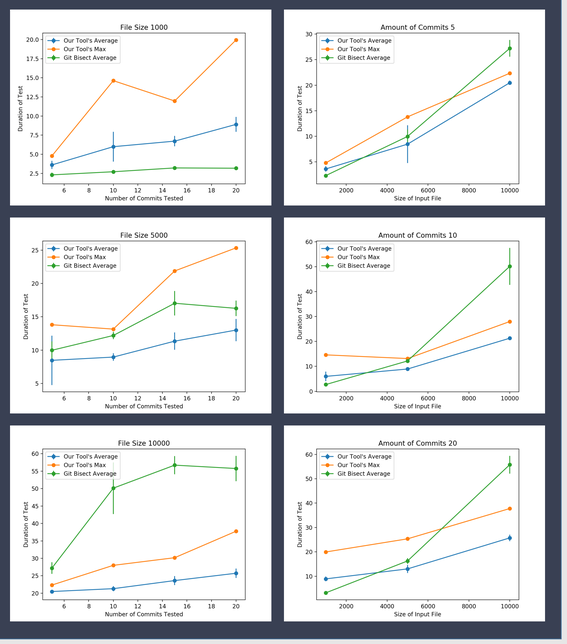
- Can testing over multiple commits be conducted effectively in a parallel, serverless paradigm utilizing Amazon Web Services' Lambda functions? [3, 4]
- Will parallel, serverless testing improve upon the time performance of automated git bisect, or will the overhead outweigh the parallel benefits?
- Can we provide users with more meaningful results, such as which tests pass and fail, than git bisect provides?

3. Methods



4. Results

- Larger file size equates to longer tests per commit
- For fixed test length, our tool slows down at a lower rate than automated git bisect as number of commits increase
  - More pronounced difference in rate for larger test lengths
- For fixed number of commits, our tool slows down at a significantly lower rate than automated git bisect as test length increases
- Confirmed that as it scales, computation of our method indeed operates in *O(k)* time, where *k* is the time is takes for the tests to run on a single commit
- Overhead still exists in uploading commits to S3 bucket
  - Reason for slight increase in completion time as number of commits increases on constant test length



5. Conclusion

- Our tool has better performance than automated git bisect with more commits and higher testing time per commit
- A next step for benchmarking would be a cost-benefit analysis, especially investigating the financial aspects
- Next steps in terms of implementation include:
  - Modularization to allow for dynamic configuration of S3 buckets to increase usability of our tool
  - Credential entry to allow our tool to be used on private repositories

6. References

[1] "White Paper Why Bother to Unit Test?" *QA Systems The Quality Assurance Company*, www.qasystems.com/fileadmin/user_upload/resources/W hite_Papers/Why_Bother_to_Unit_Test.pdf.
[2] Git-Bisect Documentation." *Git*, Git, gitscm.com/docs/git-bisect.
[3] Hendrickson, Scott, et al. "Serverless Computation with OpenLambda." *Usenix.org*, University of Wisconsin, www.usenix.org/system/files/conference/hotclou d16/hotcloud16_hendrickson.pdf.
[4] *Lambda Architecture for Cost-Effective Batch and Speed Big Data Processing - IEEE Conference Publication*, ieeexplore.ieee.org/abstract/document/7364082

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING