Understanding and Addressing Scalability Bugs in Large-Scale Distributed Systems

Shuang Liang
The Ohio State University
June 10, 2024

1. Introduction

Large-scale distributed systems are crucial to the infrastructure of a myriad of applications and services, facing inherent challenges related to availability, consistency, concurrency, and more. The continuous evolution in this field, especially with the advent of cloud computing, has introduced unprecedented scalability challenges [2, 3]. This proposal seeks to investigate scalability bugs in open-source large-scale systems to understand their impact and develop methodologies for addressing them.

2. Problem Statement

Scalability bugs in large-scale distributed systems often lead to significant operational challenges, including system downtime and data loss [4, 1]. These bugs, while latent in small-scale settings, emerge prominently as the system scales up, requiring a focused study to develop effective detection and resolution strategies [3, 5].

3. Project Goals and Objectives

The main goal of this project is to systematically analyze, characterize, and document scalability bugs in large-scale distributed systems. Objectives include:

- Provide detailed accounts of bug reproduction for recently reported scalability bugs.
- Identify and address challenges in reproducing scalability bugs.
- Develop protocols for effectively triggering and quantifying the impact of scalability bugs.

4. Expected Deliverables

The project’s expected deliverables include:

- A suite of Trovi replayable artifacts for reproducing scalability bugs.
- Jupyter notebook scripts for step-by-step investigation of these bugs.
- A comprehensive analysis of the challenges in reproducing scalability bugs compared to traditional bugs.

5. Implementation Plan

We will adopt a comprehensive approach to address scalability bugs in large-scale distributed systems such as Cassandra, HDFS, and Hadoop.

This includes meticulously investigating bug reports, comprehending bug fixes, rigorously reproducing bugs, and thoroughly comparing system behavior before and after bug resolution.
Our systematic approach aims to improve the current understanding and management of scalability bugs, providing valuable knowledge and resources to both the open-source community and the broader field of distributed systems.

**Bug Report Analysis.** We will adopt a meticulous approach to reading and understanding bug reports, focusing on capturing essential information about triggers, symptoms, manifestations, and root causes. These insights will inform the development of reproducibility protocols and facilitate the identification of recurring bug patterns.

**Engagement with Development Platforms.** Leveraging platforms such as Jira and GitHub, we will delve into the narrative of bug resolution by examining code differentials and participating in developer discussions. This engagement will guide the construction of our development environment and the configuration of tools essential for building these systems.

**Reproducing Bugs.** We will methodically reproduce scalability bugs by configuring the development environment to closely match reported conditions, including software versions, configuration settings, environmental variables, and system loads. We will engage with regression tests to validate bug presence and resolution success, exploring variations in reproduction steps when necessary.

**Quantitative Comparison.** We will run both buggy and patched versions of the system under identical conditions to conduct a comprehensive side-by-side comparison of behavior. Quantitative data on metrics such as memory and CPU utilization will be collected to measure the bug’s impact and patch effectiveness. Visualization tools will be employed to provide clear depictions of improvements made.

**Documenting the Process.** We will prioritize documentation throughout all stages of the project. We will maintain detailed records of bug analysis, reproduction attempts, and comparative studies, which will serve as valuable references for this project and will guide future efforts in the field. This systematic approach aims to enhance our current understanding and management of scalability bugs, contributing valuable knowledge and resources to the open-source community and the wider distributed systems field.

**Challenges and Proposed Solutions.** Addressing scalability bugs presents distinct challenges, notably in effectively managing memory pressure, handling elastic workloads, and ensuring machines are accurately provisioned for the jobs running on top. To expose such issues, we will work on two solutions, namely incorporating delay injection techniques and deploying specialized detection tools. These aim to pinpoint inefficiencies in resource utilization with unprecedented precision. Moreover, we anticipate that our strategies could inform future system design, guiding the development of inherently scalable architectures that can gracefully handle evolving workload demands.

### 6. Project Timeline

**Week 1: Project Initiation**
- Setup the development environment and start analysis of bug reports. Establish criteria for selecting bugs.

**Week 2: Detailed Bug Report Analysis**
- Analyze bug reports in-depth. Finalize the list of bugs for reproduction.

**Weeks 3-4: Environment Configuration and Initial Bug Reproduction**
- Configure tools and environments. Begin reproducing the first set of bugs (2-3 bugs).

**Weeks 5-7: Intensive Bug Reproduction Phase 1**
- Reproduce new scalability bugs (targeting 2-3 per week). Document the process for each bug.

**Weeks 8-10: Intensive Bug Reproduction Phase 2 and Comparative Analysis**
- Continue with bug reproduction (2-3 per week). Start comparative analysis for bugs that have been reproduced.
7. Biographical Information

Biography

Shuang Liang, currently a third-year undergraduate at the Ohio State University, is an extremely self-driven student with a passion for Cloud/Distribution Systems, High-performance Computing, etc.

Email: liangshuang2333@gmail.com
Github: ThreadEater

7.1 Education

• B.S. in Computer and Information Science, The Ohio State University, GPA: 3.63/4, August 2023 - Present
• B.S. in Computer Science (Transferred out), China University of Geoscience, Wuhan, GPA: 3.61/5, September 2021 - July 2022

7.2 Technical Skills

• Programming Languages: C, C++, Java, Python, Matlab
• Tools: Git, Github, Visual Studio, Linux, \LaTeX

7.3 Experience

• The International Collegiate Programming Contest Club, Skills: Algorithm Design and Analysis, Advanced Data Structures, Silver Medal at ICPC Central China Regional Competition 2021, 2022 ICPC East Asia Jinan Regional Gold Award. October 2021 - June 2022, Wuhan, China.
• Throttling in Large-Scale Distributed Systems, Advisor: Yang Wang. Investigated and reproduced throttling bugs in Apache’s popular large-scale distributed systems (e.g., Spark, Yarn, HDFS, Hadoop). Attempted to build a tool that can detect throttling bugs in large distributed systems. October 2023 - Present, Columbus, OH, USA.
• Undergraduate Teaching Assistant, CSE 1222: Introduction to C++ Programming, Instructor: John Wilcox. Collaborated with professors and other TAs to grade student work. Tutored students in lab classes. Held office hours to answer students’ questions. September 2023 - January 2024, Columbus, OH, USA.
• A 2D Planar Drawing System, Object-oriented Programming Course Project. Used C++ language and Qt framework to write a raster image vectorization program and utilized the raster image as the base map to enable the drawing function of points, lines, and polygons, and stored the vectors as files. December 2021 - January 2022, Wuhan, China.
References


