A Study of Bugs Found in the Moby Configuration Management System

Author
Mattia Bonfanti - m.bonfanti@student.tudelft.nl

Responsible Professor
Prof. Dr. Ir. Diomidis Spinellis - d.spinellis@tudelft.nl

Supervisor
Thodoris Sotiropoulos - theosotr@aueb.gr

1 Research Background: What and Why?

What is Moby?

- Open framework by Docker (but it's NOT Docker).
- Allows the creation of secure container systems [1].

Why studying bugs?

- Understand complex software systems.
- Improve detection, prevention and software quality.
- Many methodologies already researched [2, 3, 4].

Why this research then?

- Expand existing methodologies to configuration management systems.
- Provide container systems perspective to research.

2 Research Questions

1. What are the **symptoms** of these bugs?
2. What are the **root causes** of these bugs?
3. What is their **impact**?
4. How do developers **fix** these bugs?
5. Are these bugs **system dependent**?
6. What **triggers** these bugs?

Get the paper! →

Get the code! →

3 Research Methodology

Step 1: Create Bugs Database

1. Filter issues and pull requests to find bugs
   a. From **21378 issues** to **2539**
   b. From **22079 pull requests** to **2539**
2. Keep only bugs with a fix, from **5078** entries to **2285**

Get issues and pull requests from GitHub → Filter results using keywords and synonyms → Clean data and keep only bugs with a fix → Store data in database

Step 2: Analyze Bugs Data

1. Pick a random sample of 100 bugs
2. Define categories for bugs and fixes
3. Identify symptoms and root causes of bugs
4. Identify impact and fixes of bugs
5. Define system dependency and triggers

Pick random sample from bugs database → Define categories for bugs and fixes → Identify bugs categories to answer research questions → Store data in database

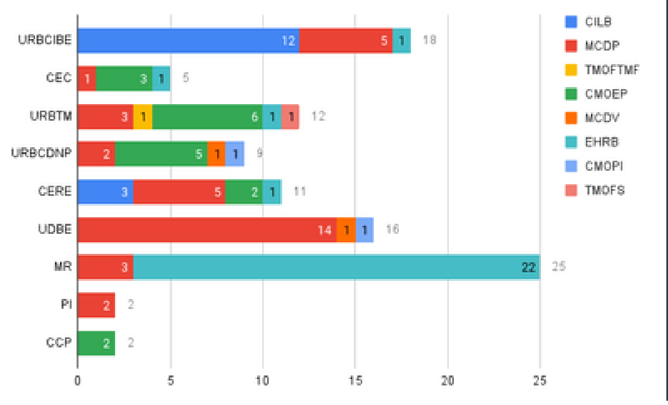
Step 3: Find Patterns

1. Group similar bugs based on category, areas and fixes
2. Identify common patterns
3. Broaden the focus and draw insights and main takeaways from the results

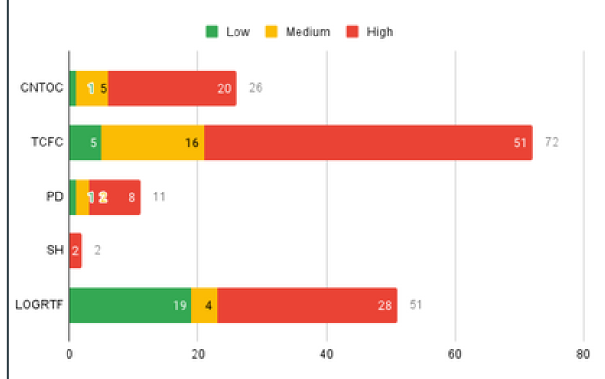
Group similar bugs → Identify common patterns → Find insights and takeaways

4 Research Results

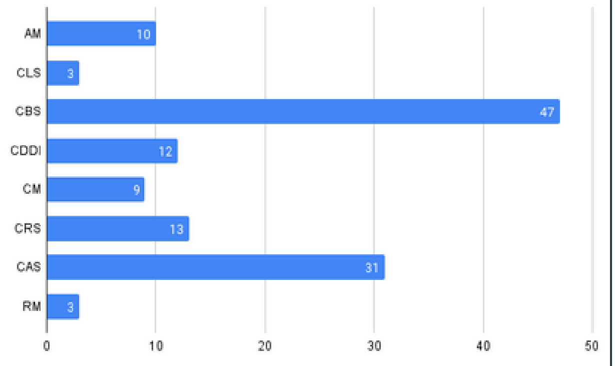
Symptoms & Root Causes
Distribution shows how symptoms (y-axis) relate to specific root causes



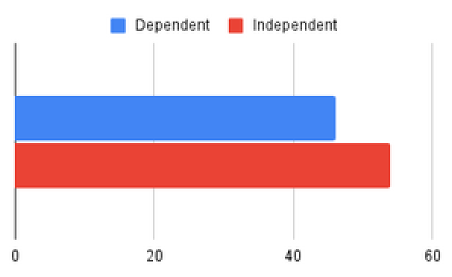
Impact & Consequences
Impact is mostly high and affects configuration and logging

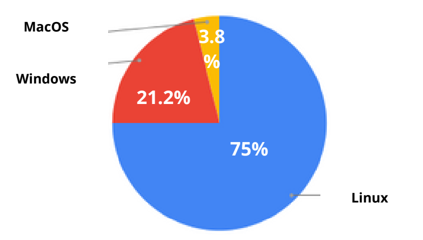


Fixes
Fixes affect small amount of lines of code. Focus on specific components.

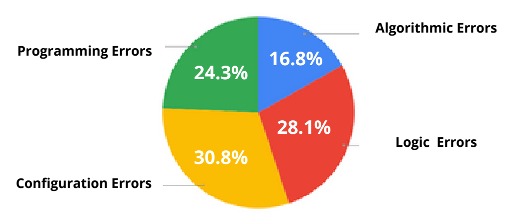


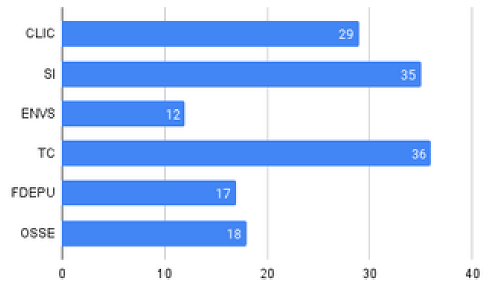
System Dependent





Triggers





5 Conclusion and Further Work

Insights and Takeaways

1. **Symptoms**: misleading reports, containers errors and dependencies errors. Mostly linked to a certain root cause.
2. **Root Causes**: faulty dependency configuration, errors reporting, containers and core functions run-time.
3. **Impact** is generally **high** and mostly results in: wrong targets configuration and logging problems.
4. **Fixes** involve small changes in branch and assignment statements. They focus on execution components the most.
5. **Balanced split of system dependent and independent** bugs.
6. **Triggers**: errors in configuration and logic. Lack of test cases requires specific instructions and setup to reproduce bugs.

What is Next?

1. **Improve keywords filtering** to find bugs in GitHub by looking at the bugs reports as a whole rather than finding only specific terminology.
2. **Develop a dedicated test suite** leveraging the patterns between symptoms and root causes to prevent and reduce the most common bugs.
3. **More structured and stricter contribution workflow** to Moby. Test cases should be mandatory.
4. **Develop a tool to overview the history of bugs** in Moby. Developers can use this to avoid bug patterns to appear again after major refactoring.

References

[1] Moby. Moby project, 2017
[2] Wang Y., et al. An empirical study of multi-entity changes in real bug fixes. pages 287–298. IEEE, 9 2018.
[3] Y. Zhao, et al. Towards an understanding of change types in bug fixing code. Information and Software Technology, 86:37–53, 2017.4
[4] Stefanos Chaliasos. Well-typed programs can go wrong: A study of typing-related bugs in jvm compilers. 2021.

Images: Moby, TU Delft, GitHub, Material Icons, FA Icons, Canva, Own work.