ENVIRONMENTAL IMPACT OF STATIC ANALYSIS TOOL CONFIGURATIONS

Sophie van der Linden | savanderlinden@student.tudelft.nl
Supervisors: Xutong Liu, Carolin Brandt
EEMCS, Delft University of Technology, The Netherlands

1. Background

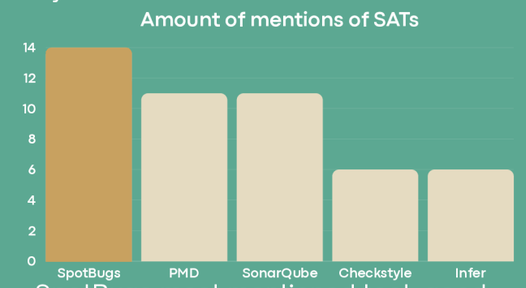
- Continuous Integration relies on frequent commits for project consistency
- Static analysis tools, SATs, are often used in CI workflows
- SATs are used to detect potential bugs in code, without actually executing the code, can have different configurations
- The energy consumption of CI processes is significant

2. Research Questions

- What are the most commonly mentioned static analysis tools for Java in primary studies?
- How does energy consumption vary across static analysis tool configurations?
- Is there a trade-off between performance and energy consumption of different configurations?

3. Commonly Used SATs

- Systematic literature review



- SpotBugs most mentioned tool, good configuration options, used in research

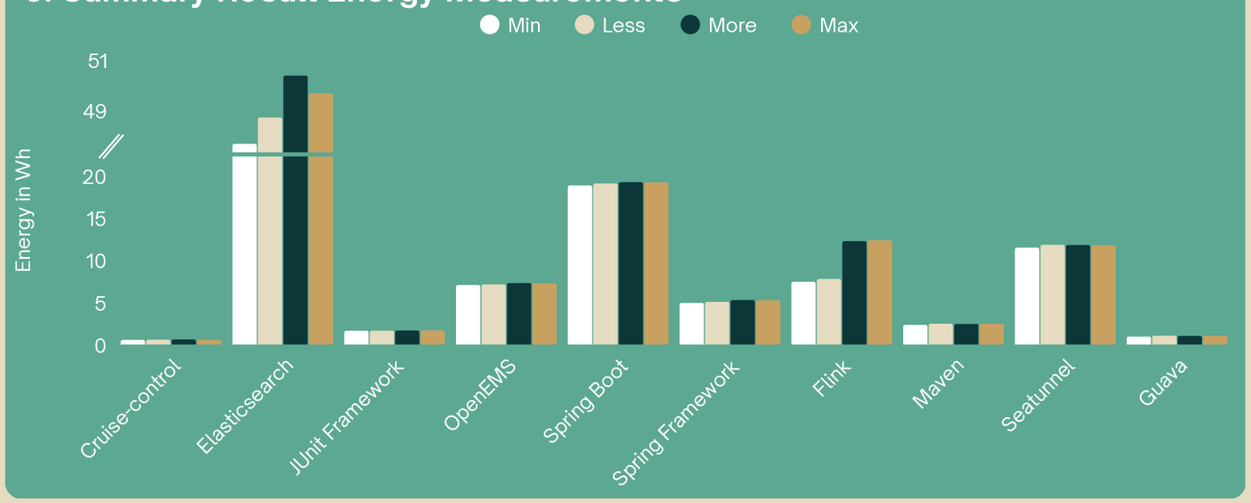
4. SpotBugs

- Configuration: effort Min - Less - More - Max

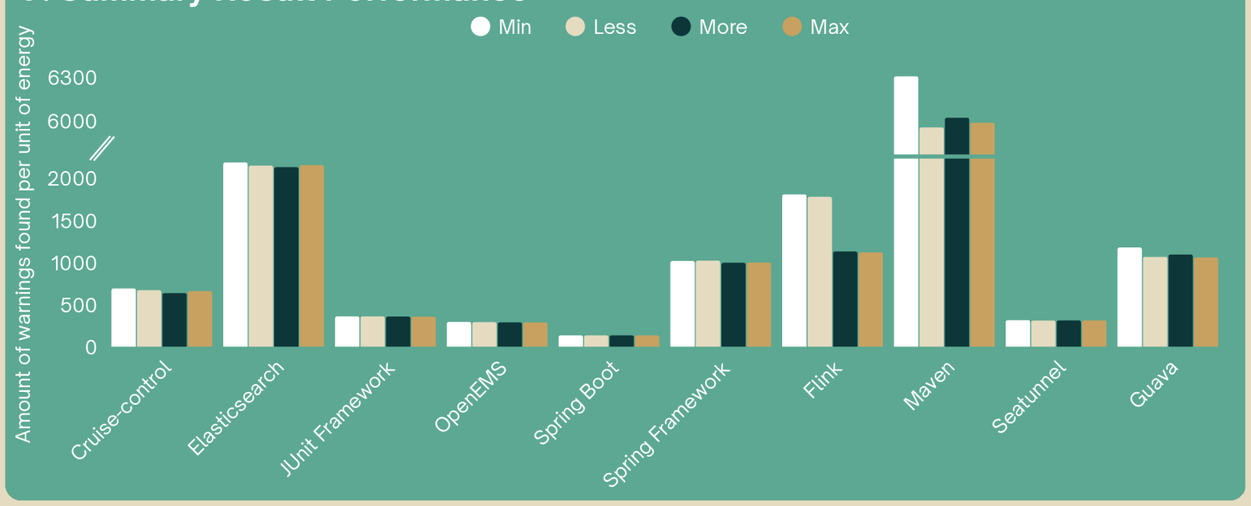
5. Methodology

- Run SpotBugs with different efforts on 10 projects
- All the projects and efforts are run 30 times
- One minute break between each measurement
- Controlled environment
- Measurements done with EnergiBridge
- Caching/incremental builds disabled
- Measure performance by extracting warnings

6. Summary Result Energy Measurements



7. Summary Result Performance



8. Discussion

- Results are useful for comparison, but exact energy values may be affected by background activity
- SpotBugs was the most mentioned tool, but is only for Java
  - Tools for other programming languages may show different results and can be explored
- The setup did not use incremental builds, which can be used in real pipelines
  - Using caching and incremental builds could significantly reduce energy usage
- The highest estimated energy consumption, Elasticsearch with effort More, together with Elasticsearch's yearly CI pipeline execution, equals about one-third of an average EU household's yearly electricity

9. Conclusions

- SpotBugs most frequently mentioned Java static analysis tool
- Almost all higher efforts resulted in higher energy consumptions
- The largest energy differences at Min/Less and Less/More, More/Max was rarely significant
- Increased effort improved bug detection in projects, but came with a higher energy consumption → Trade-off
- Most efficient effort is Min
- Future work: other programming languages, additional SATs, real-world CI setups
- Potential for energy-efficient configurations in CI pipelines, towards more sustainable software engineering