

Álvaro Ibáñez Coedo UO282206

Sergio Quintana Fernández UO288290

Jorge Joaquín Gancedo UO282161
Fernández

Miguel Estapé Fernández UO277566

Clean Code, Horrible Performance

Table of contents

Introduction

Application architecture and latency

Clean code and information hiding

Clean code vs performance

Monoliths and microservices

Decisions against trade-offs

Premature optimization

Other questions

Introduction

Casey Muratori



Clean Code ==
Bad Performance?



Application architecture and latency

A discussion about modern day applications
performance limits



Giovanni's View



- Applications nowadays are IO bound (their speed depends highly on IO subsystems)
- In big databases that are slow, increasing code performance won't make such a difference in terms of the application performance.

Casey's View



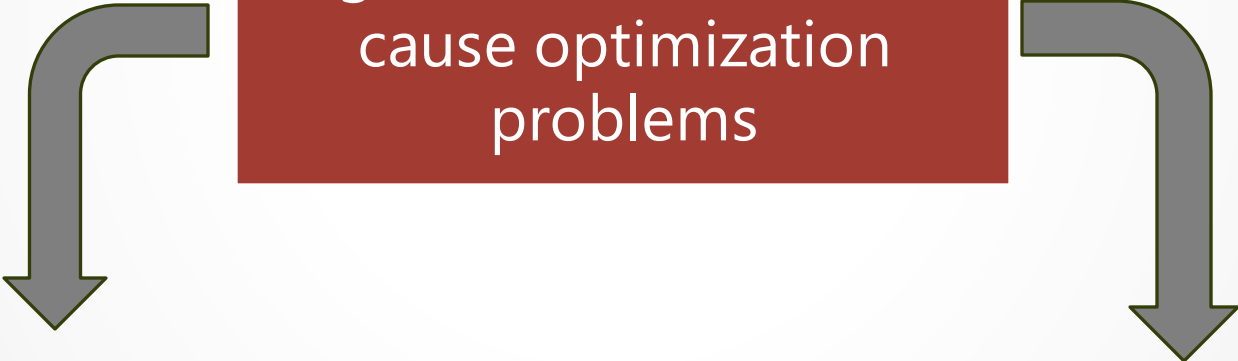
- Applications aren't really IO bound but built to be IO bound.
- The fact that a part of the application depends on a part that is slow isn't an excuse.
- Facebook performance improvements announcements.

Clean
code and
information
hiding

Information hiding damages
performance



Big class hierarchies can
cause optimization
problems



Compiler
optimization

Ifs and
switches

Compiler optimization

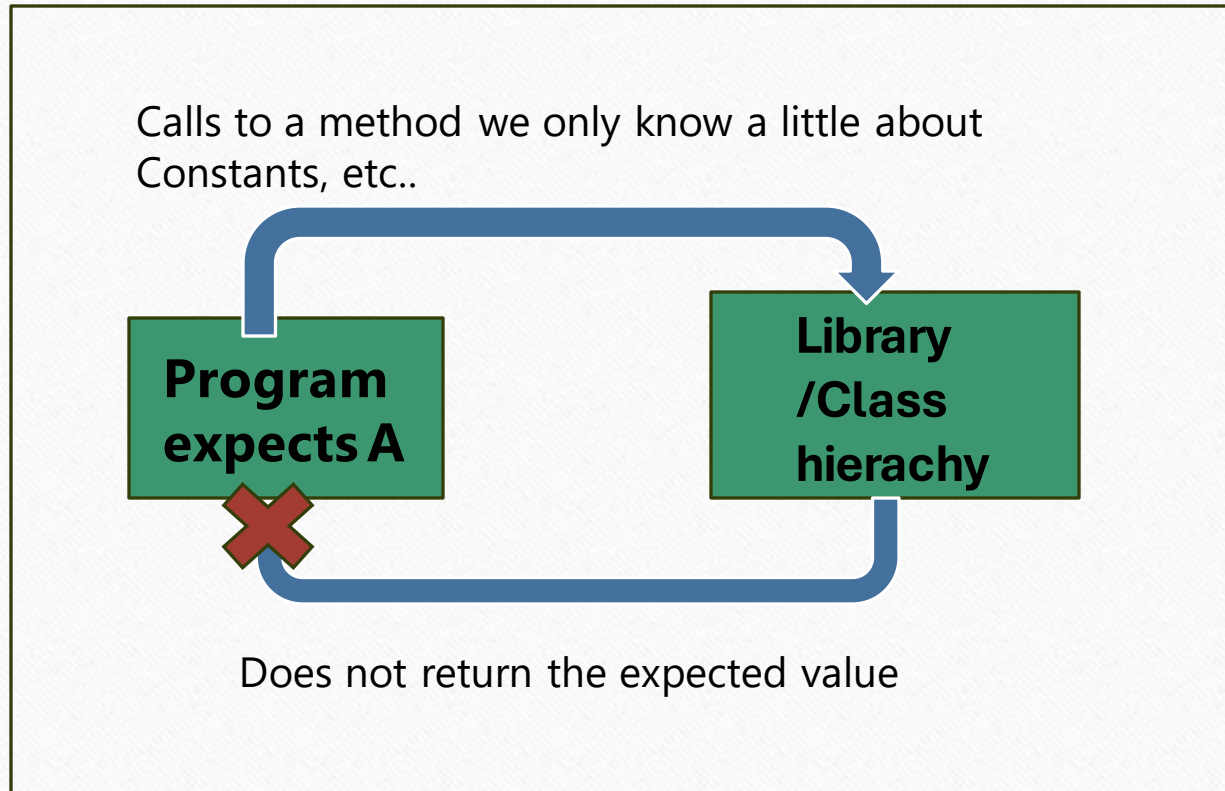
- Big class hierarchies call a lot of accessing functions
- Exposed data types don't have this issue since the compiler can see their definition

If's and switches

- Ifs and switches can be optimized easily (vectorization)
- Class hierarchies are much harder to optimize

Information hiding

Understandability and external libraries



If the method does not do what we expect it may be a problem with us, not understanding what the code does or a bug in the code

Proposed solution

Exposing the contents of the library so now we know exactly what the code does

- Easier to understand the functioning of the code
- Optimization with details of the implementation

More about this in the next point

Object Oriented Programing

MAIN ISSUE



Class hierarchies

Damages
performance

Not really more
readable

Bloated amount
of code lines

Does not read
as it performs

Readability advantage

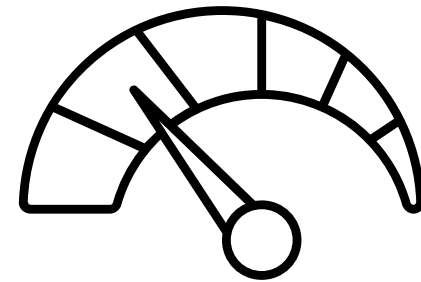
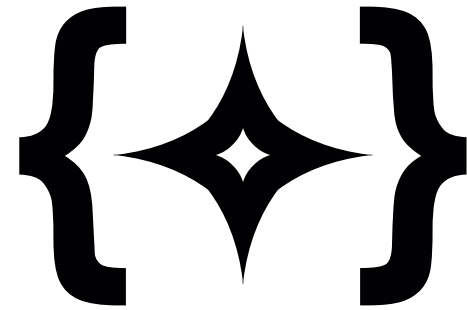
ORP is not more readable than Procedural Programing because of class hierachies

It can be made equally readable

- Making functions with repeated code
- Good names

Produral is faster, has smaler size code and can be as readable

Clean code vs performance



Monoliths and microservices

Casey Muratori advocates for monoliths against microservices, stating that the latter are “*software engineering’s solution for Conway’s Law*”.

He also argues that microservices is not a bad architecture in the case that the work division is unenforceable in any other way and if the code quality and speed trade-offs are considered.

Trade-offs are informed decisions in which their own impact has been already considered and accepted beforehand.

Decisions are all other decisions which have been accepted and their impact has not been considered beforehand.

Example: Python against JIT-compiled languages.

Decisions against trade-offs

Premature optimization



“Don’t worry about writing non-efficient code.”



“If you have some source code of, say, 1000 lines, that is not as fast as it should be and 30 of those 1000 lines are executed more often than the rest, then you should focus your attention on those 30 lines.”



Performance should be considered since day zero (even during architecture) instead of forcing the code to adapt to the architecture.

Other questions



Objective vs subjective quality metrics



Clean code and cognition



Study of code performance