

Image used completely without permission.



A Bug's Life

Operating Systems and Process
Oriented Programming (1DT096)

Oliver Eriksson Edholm
Aleksander Lundqvist
Henrik Sommerland
Ludvig Strömberg
Edvin Wahlberg
Oscar Wallster

What are we going to do?

- We are going to simulate an ant colony, inspired by real ants and ACO
- Optimizing the simulation using concurrency
- Every ant is an actor
- Each ant only sees its immediate surrounding and has very limited cognitive abilities.

Why is this an awesome project

- Ants and swarm intelligence is ultra awesome

Everybody gets super excited about it :D

- Suitable for concurrency
- Scalable

Easy to get the basics working, unlimited possibilities for expansion

- Easily split in to modules ; easy to distribute workload
- Will look incredibly cool!

Cool.....



What are we going to learn?

- The advantages of using concurrency when simulating complex systems with lots of actors
- Working with the actor model
- Basic swarm intelligence
- A new language

Concurrency

- Free from global locks

No stop the world scenarios.

- Free from global bottlenecks

Performance issues should only affect the local neighbourhood.

- Purely actor model based

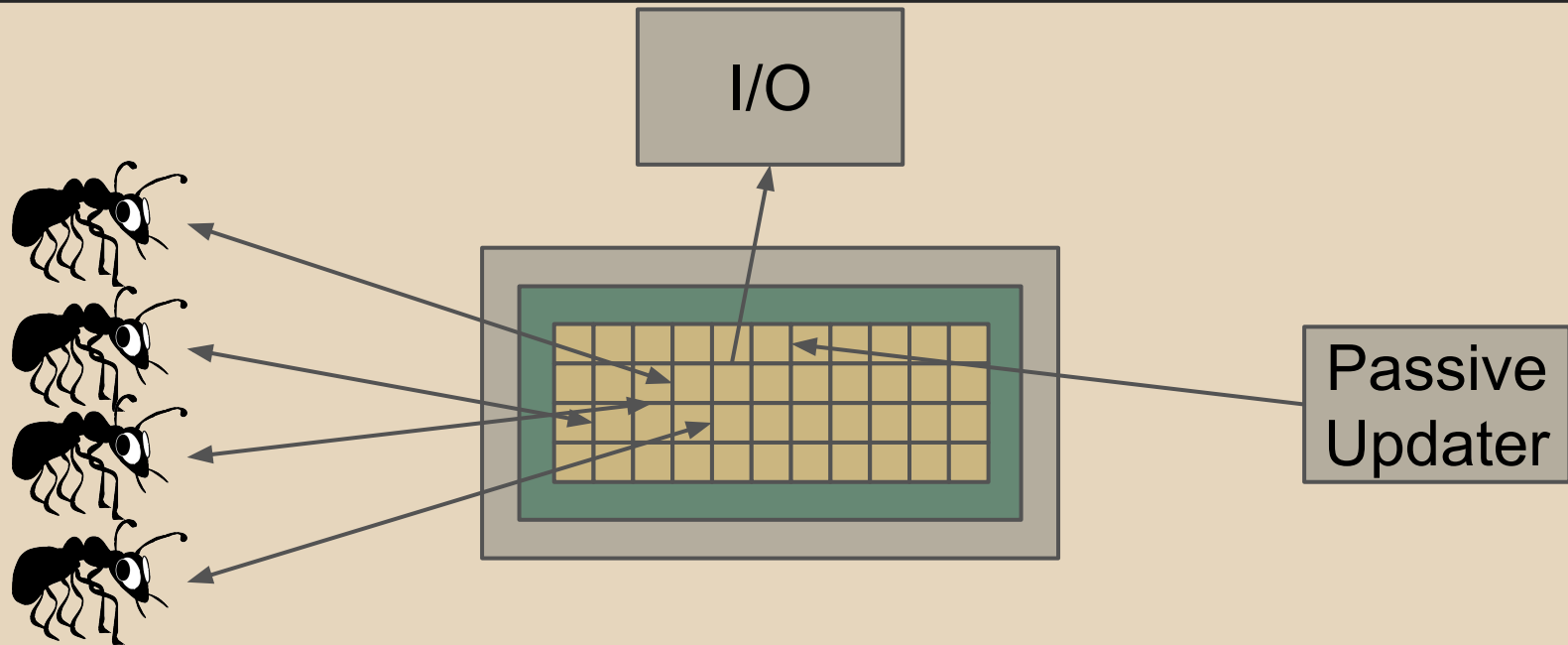
All interprocess communication and state sharing happens through message passing.

- Massive concurrency

Over 9000 actors.

- Nondeterminism is a feature, not a bug

Architecture



EVERYTHING IS AN ACTOR

Technical Challenges

- **Massive parallelism**

May be really tricky to avoid deadlocks and starvation

- **Lots of parameters for the rules**

Life span, availability of food, pheromone saturation rate , etc

- **Learning a new language**

Social Challenges

- Distributing work in an effective manner
- Keeping up the communication within the group
- Scheduling in a way that is convenient for everyone in the group

Language choice

We need a language which can enable us to meet our concurrency goals

Obvious disqualifications:

- C/C++
Too low level. A need to manually implement the actor model.
- Java
Meh... Not the proper concurrency model.
- Python
Slow and bad concurrency

Possible Candidates:

- Rust
- Nim
- Erlang
- Encore

Rust



- Pros

- Fast
- Safe
- Built for concurrency
- Memory safety
- Actor model
- New and cool!

- Cons

- Complicated
- Beta
- Restrictive memory handling

Nim



- Pros

- Fast
- Simple
- Built for concurrency
- Python like syntax
- Easy to learn
- Some library support
- New and cool!

- Cons

- Only version 0.10
- Not memory safe when multithreaded
- Some library support
- Not very well-documented

- Pros

- Based around the actor model
- Simple
- Stable and established
- Well documented

- Cons

- Old and lame
- Slow
- Functional/Hard to modify states

- Pros

- Completely built on the actor model
- Simple and clean
- Fast
- Cool to contribute to science

- Cons

- Pre alpha
- Subject to constant change
- Limited functionality

Verdict

NIM

But with erlang as backup

Tools

No restriction on usage of editors or IDEs

- Trello
- Github
Avancerad git
- Skype
- Slack