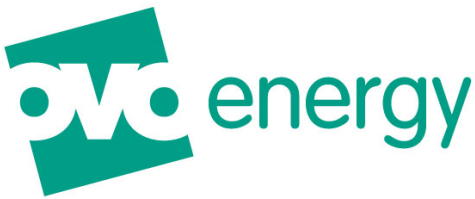
An Introduction to Akka and the Actor-Based Model

Daniela Sfregola

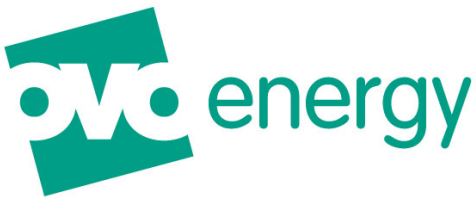
@DanielaSfregola
http://danielasfregola.com/

Scala Italy 2015



Outline

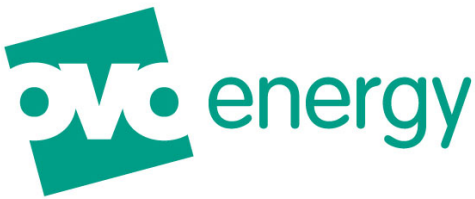
- Concurrency
- Thread-Lock-Monitor Approach
- Message-Passing Approach
- Akka as Message-Passing Framework
- Actors in Akka
- Actor Life Cycle
- Actors Hierarchy
- Supervision Rules
- Demo



Motivation: Concurrency

- Multiple Core CPUs
- Distributed Systems
- Low Cost Hardware
- Parallelism

- We need to deal with shared resource between processes!



Thread-Lock-Monitor Approach

Thread-Lock-Monitor approach can be challenging:

- Logic complicated for humans
- Difficult to design and maintain
- High unpredictability

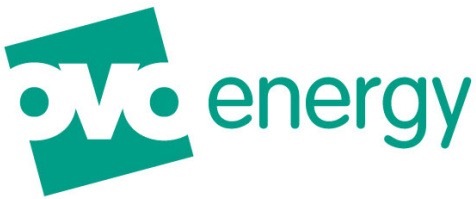
Do we have an easier approach to tackle concurrency problems?

ovo energy

Message-Passing Approach

The Message-Passing Approach:

- Encapsulates all the shared information in messages
- Messages are used to communicate between processes



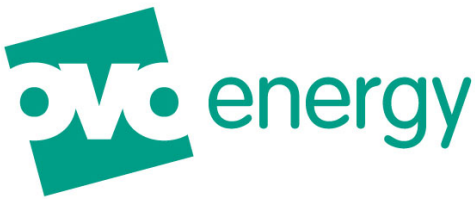
Akka as Message-Passing Framework

Akka - from `akka.io`

Build powerful concurrent & distributed applications more easily.

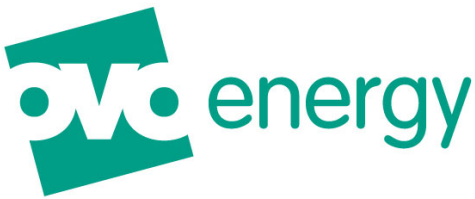
- Simple Concurrency & Distribution
- Resilient by Design
- High Performance
- Elastic & Decentralized
- Extensible

It supports Scala, Java... .NET is coming soon!

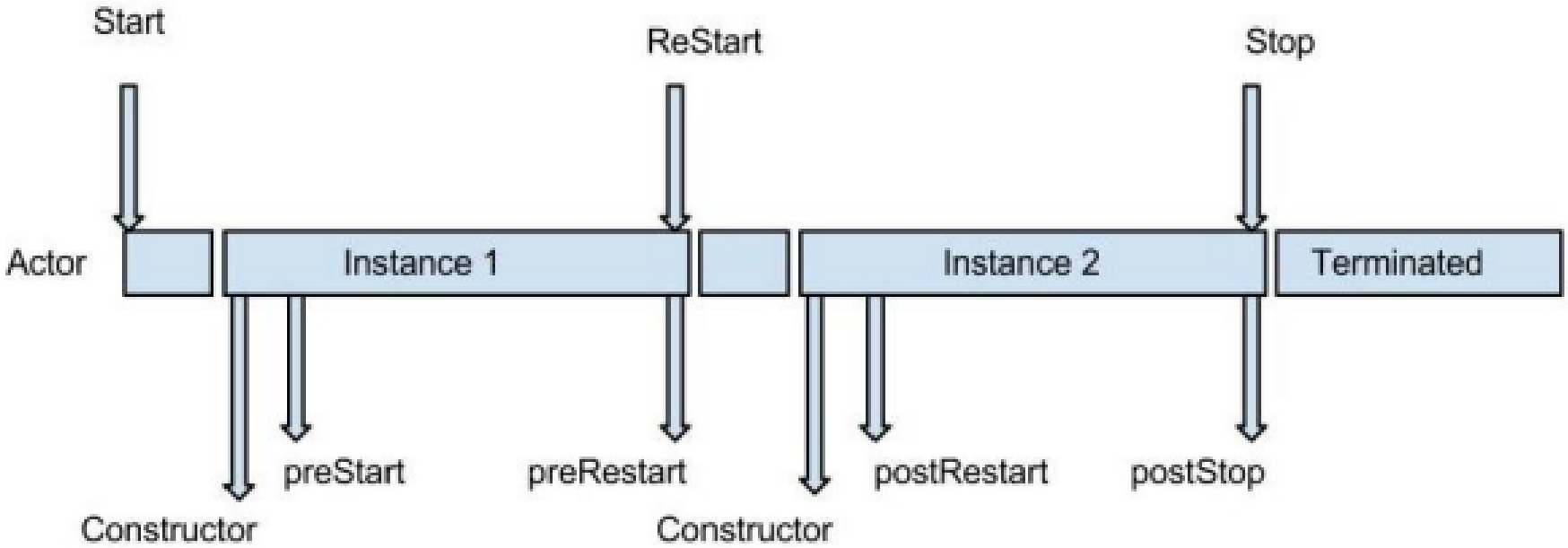


Actors in Akka

- Lightweight concurrent entity
- Event-driven
- Mailbox of messages processed asynchronously
- Can hold status/mutability

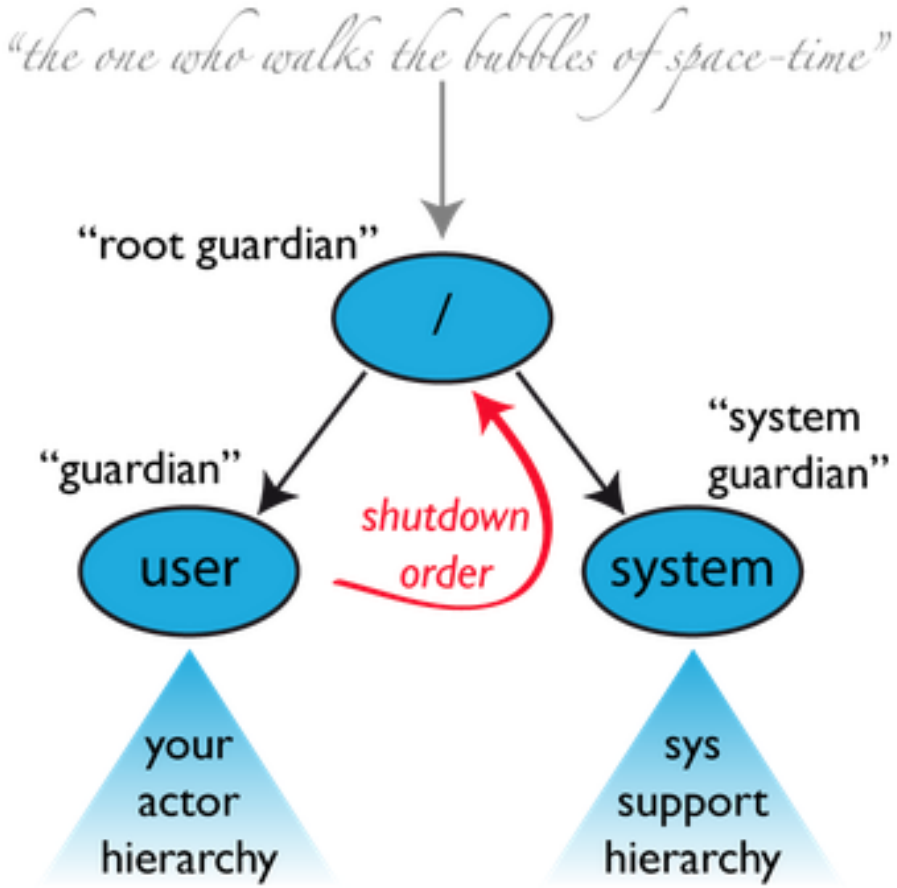


Actor Life Cycle



Akka in action (2013) by R. Roestenburg, R. Bakker and R. Williams.
Shelter Island: Manning Publication

Actors Hierarchy



from `http://doc.akka.io/docs/akka/snapshot/general/supervision.html#supervision`, accessed on May 2015

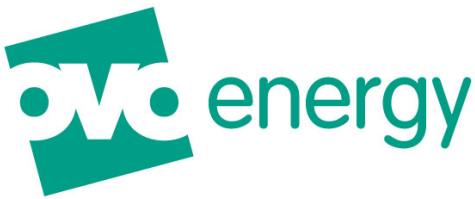
Supervision Rules

- Your Father is your Supervisor
- Every Actor has a Supervisor, but the Guardian Actor (/user)
- Your Children follow your destiny
- If unable to handle an exception, escalate it to your Supervisor
- If the Guardian Actor is unable to handle an exception, the system will shutdown

Actor Core Operations

There are four core operations on Actors:

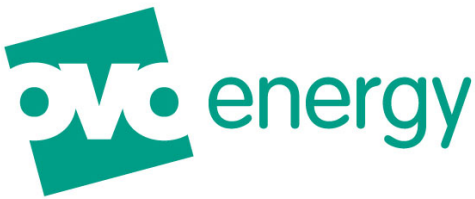
- create
- send
- become
- supervise



Demo

Demo Time!

Gist available at
`https://gist.github.com/DanielaSfregola/6dc52bffa2ed566de9b2`



Conclusions

- Akka as Message-Passing approach to concurrency
- Main components: Actors
- Best Practices:
  - Never Block
  - Communicate only via messages
  - Messages should be immutable
  - Messages should be complete and self-contained

Conclusions

- Akka as Message-Passing approach to concurrency
- Main components: Actors
- Best Practices:
  - Never Block
  - Communicate only via messages
  - Messages should be immutable
  - Messages should be complete and self-contained

Thank you!