

Multithreading Magic

**Why Everything
You Thought You Knew
about Concurrency is Bogus,
if not actually Totally Insane**

by
Pieter Hintjens, CEO, iMatix

FOSDEM 2011



Pieter Hintjens, CEO, iMatix
FOSDEM 2011 – cc-by-sa



Hintjens' Law of Concurrency

$$e = mc^2$$

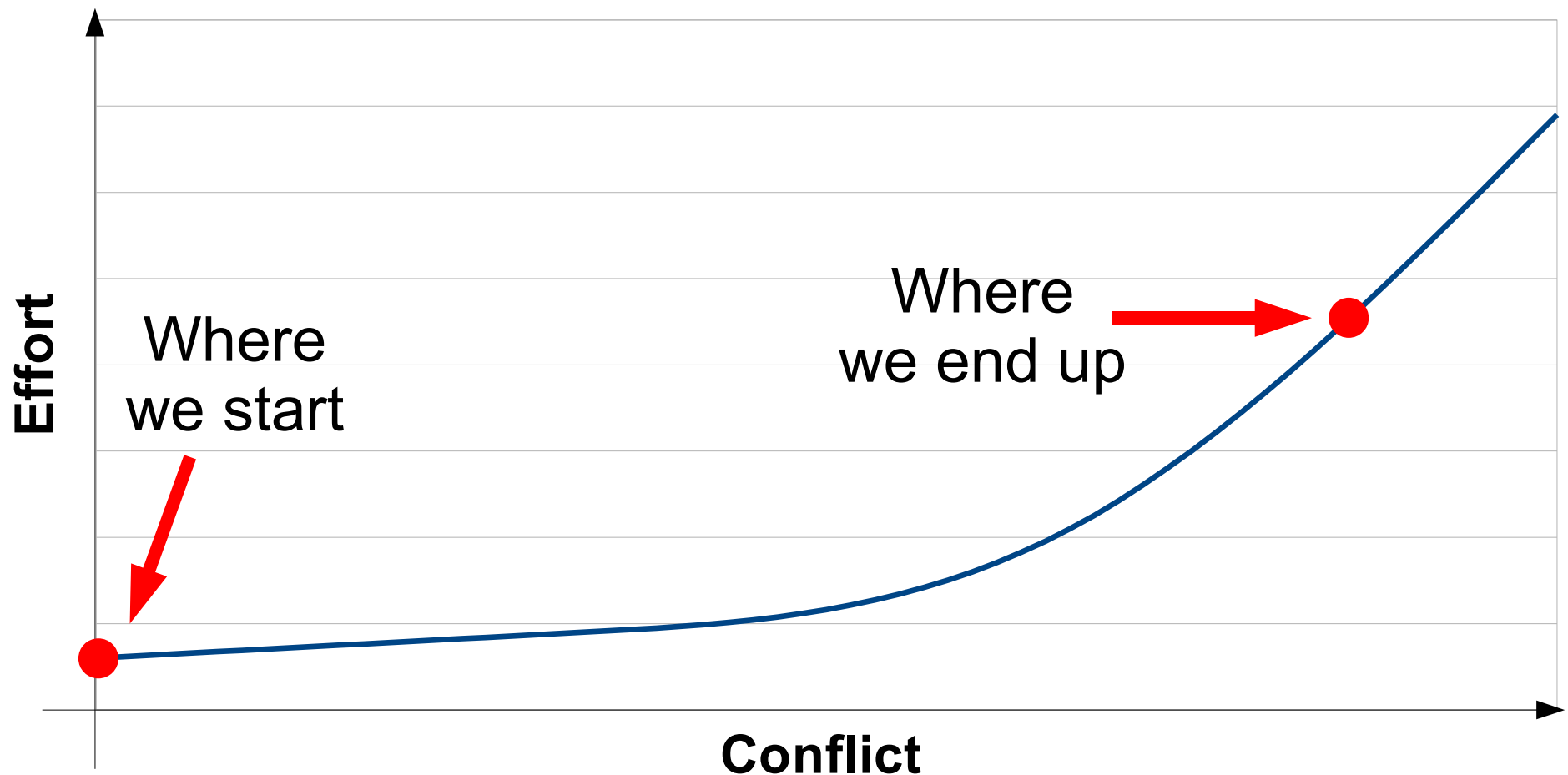
e is for *effort*, the pain that it takes

m is for *mass*, the size of the code

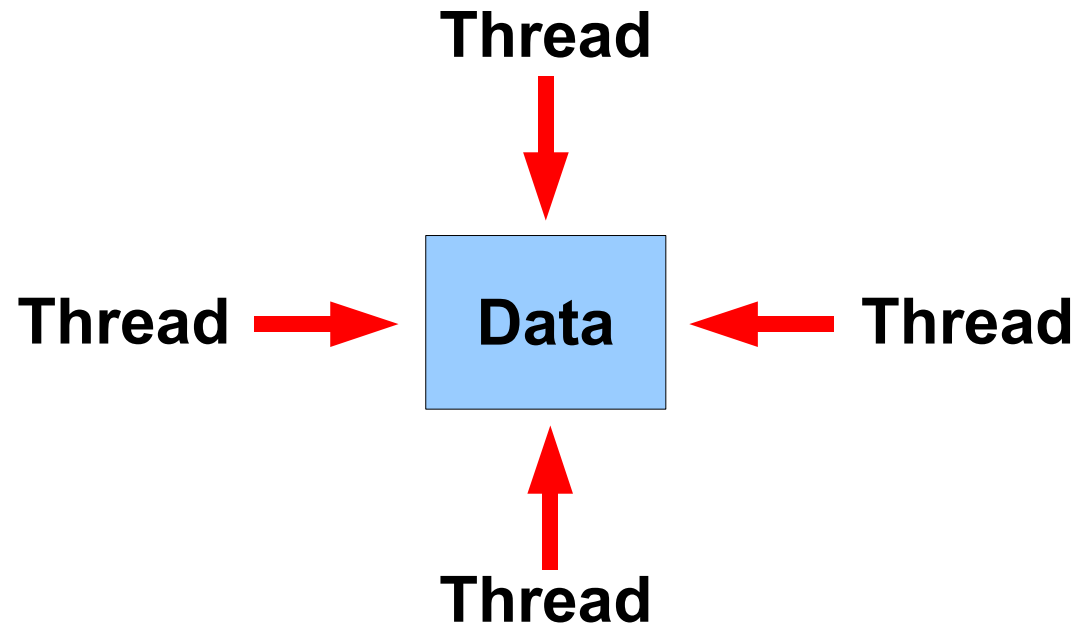
c is for *conflict*, when **c** threads collide



The Cost of Concurrency



The Data-Centric Model



The industry standard model for concurrency,
based on the Big Iron computing model

What You (Probably) Know

Concurrency = **data shared between threads**

So, locks, semaphores, critical sections

So, code that fails by default

So, complexity, weird bugs, race conditions

Diminishing returns as **c** exceeds '*a few*'

A bogus, if not totally insane model



The Ideal World



What You (Probably) Don't Know

The Actor Model

Carl Hewitt, 1973, *"Motivated by the prospect of thousands of independent microprocessors, each with its own local memory, communicating via a high-performance network."* – Wikipedia

Gives us concurrency with **c = 1**

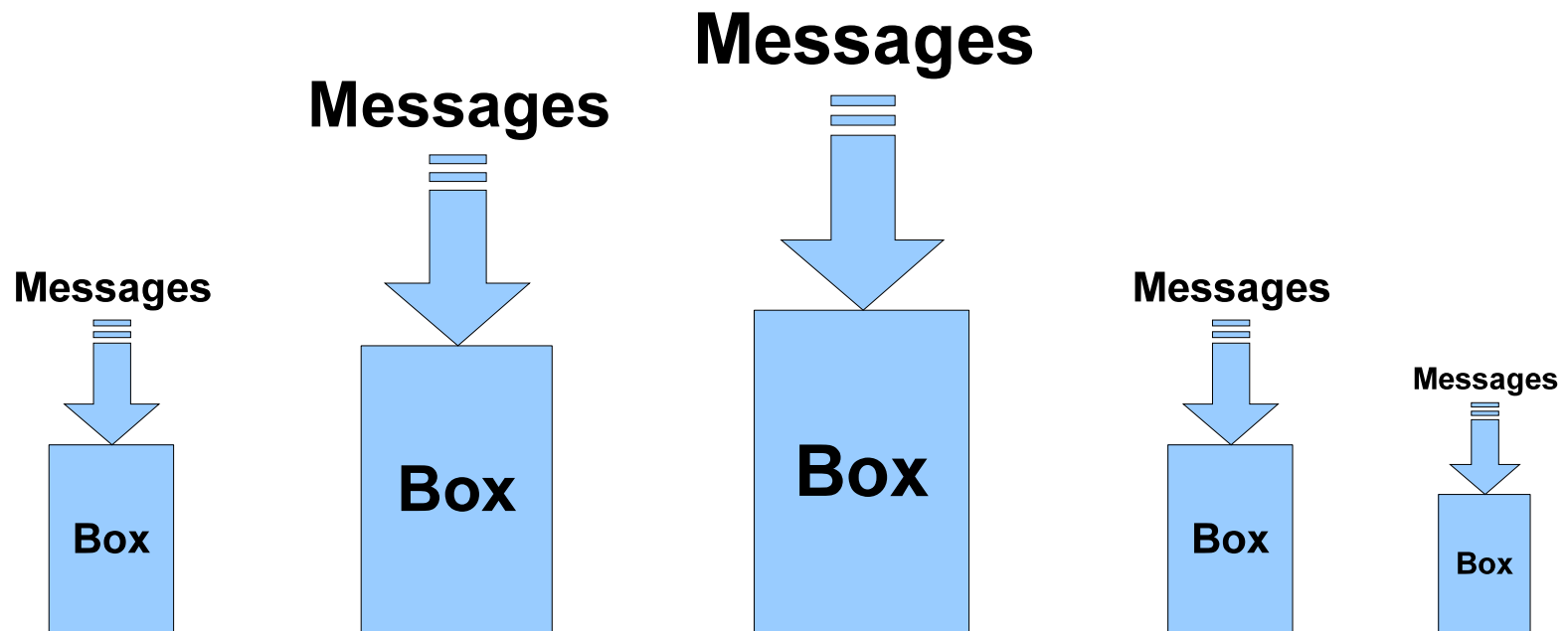
For 30 years, mostly ignored

Distributed systems were too small and rare

Maybe best known via Erlang



Boxes, sending messages



A better match for a world filled with CPUs



Pieter Hintjens, CEO, iMatix
FOSDEM 2011 – cc-by-sa



Ingredients

Lots of cheap boxes – *check!*

Thousands of threads, processes, cores, systems

Messages as contractual APIs – *check!*

XML, YAML, JSON, etc.

Any level of meta we want

A high-performance network?

Brings us to **ZeroMQ**

Cheap and fast networking – *check!*



Pieter Hintjens, CEO, iMatix
FOSDEM 2011 – cc-by-sa



ZeroMQ: $e = mc^2$, for $c = 1$

Zero shared data – a modern Actor model

Connect threads, processes, boxes

The same socket-style API for any transport

Add any message contracts you want

ZeroMQ transports blobs

Connect tasks using *messaging patterns*

Request-reply, pub-sub, pipeline,...



Pieter Hintjens, CEO, iMatix
FOSDEM 2011 – cc-by-sa



Your ZeroMQ Applications

Any mix of languages and platforms

No locks, semaphores, critical sections

- Each task has its state

- Process incoming messages from all sources

- Send onward messages to other tasks

Scale to any number of cores

- Threads or processes on one box

- Multiple boxes on one network



Pieter Hintjens, CEO, iMatix
FOSDEM 2011 – cc-by-sa



Get ZeroMQ

<http://www.zeromq.org>

Free software, LGPL licensed

In continuous development since 2007

Large, active community (~70 people on IRC)

APIs in C, C++, Python, Ruby, Perl, Java, Go,
Erlang, C#, PHP, Lua, Clisp, Haskell,...

Linux, UNIX, OS/X, Windows,...



Pieter Hintjens, CEO, iMatix
FOSDEM 2011 – cc-by-sa



About iMatix

Messaging, free software, communities

In 2010 we turned ZeroMQ fully over to its community (which we are part of :-)

Our business is commercial support for companies using ZeroMQ in infrastructure projects

Contact me at **ph@imatix.com**



Pieter Hintjens, CEO, iMatix
FOSDEM 2011 – cc-by-sa

