The Parallelism and Concurrency Landscape

Where Does Erlang Fit?
Parallelism > Multiple Cores
Parallel ≠ Concurrent
Concurrency > Actors
Transistor count

Multi-core Crisis

Single-core Performance

?!
• Increasing clock speed
• Bit-level parallelism
• Instruction-level parallelism
Operand 1 ➔ ➔ ➔ ➔ ➔ ➔ Result
Operand 2 ➔ ➔ ➔ ➔ ➔ ➔
The diagram shows a sequence of nodes labeled with 'a' and 'b' values. The sequence is as follows:

- $a_6$, $a_7$, $a_8$, $a_9$
- $b_6$, $b_7$, $b_8$, $b_9$
- $a_7b_7$, $a_6b_6$, $a_5b_5$, $a_4b_4$, $a_3b_3$
- $a_1b_1$, $a_2b_2$
inputA | inputB | output

- work-item 0
- work-item 1
- work-item 2
- work-item 1023
“Concurrency is about dealing with lots of things at once.
Parallelism is about doing lots of things at once.”

–Rob Pike
http://concur.rspace.googlecode.com/hg/talk/concur.html
(remove< (factor? 2 ...) ...)

(remove< (factor? 3 ...) ...)

(remove< (factor? 5 ...) ...)
Parallelism > Multiple Cores
Parallel ≠ Concurrent
Concurrency > Actors