Message Passing Concurrency Shootout

Dr Kevin Chalmers

School of Computing
Edinburgh Napier University
Edinburgh
k.chalmers@napier.ac.uk

Edinburgh Napier UNIVERSITY
We have talked about trying to get more ideas out from this community for a number of years.

Recently there has been a number of languages providing some form of message passing concurrency model.

- Google’s Go being the most prominent example.

- Many claim CSP or other process calculi as inspiration.

- I was having lunch with Adam Sampson around spring time and he mentioned another language - nim.

- This got me thinking about comparing these different languages with regards to their message passing concurrency support - so I came up with a project idea.
Initially I had the thought of looking at all message passing languages and support libraries.

I now think this is too ambitious - so let us just concentrate on languages.

I’m essentially interested in 3 areas:

1. The languages actual adherence to their process calculi roots
2. The simplicity of the language to write programs in a message passing style
3. Performance metrics
1. How well supported are the primitives and ideas of CSP, CCS, and the $\pi$-calculus in the range of languages supporting message passing concurrency?

2. What are the metrics of the languages supporting message passing concurrency?
Languages - have to have a Linux compiler

- Ada
- Ateji PX
- Clojure
- D
- Elixir
- Ensemble
- Erlang
- Go
- Guppy
- Hume
- Kilim
- Hume
- Limbo
- Nim
- occam-pi
- Oz
- ProcessJ
- Perl
- Rust
- Unicon
- Message passing support (this is the minimum criteria)
- Type of message passing support - synchronous and/or asynchronous
- First Order Channels (not all languages provide a channel construct)
- Higher Order Channels (channels that can send channels)
- First order processes
- Higher order processes (channels can send processes)
- Parallel execution statement
- Process ownership (e.g. a process cannot complete until all its child processes have)
- Selection on incoming messages
- Other selection types? (e.g. skip, timeout)
- Selection on outgoing messages
- Multiway synchronisation
• Channel communication time
• Channel selection time
• Lines of code
• Speedup and efficiency
• Memory usage
• CPU utilisation
Test Applications - just ideas at the moment

- Commstime
- Stessed alt
- Dining philosophers
- N-body
- Monte Carlo simulation
- etc.
The aim is to produce a journal paper around these results - a state of the art article on message passing concurrency.
• The aim is to produce a journal paper around these results - a state of the art article on message passing concurrency
• But this is a big job
• The aim is to produce a journal paper around these results - a state of the art article on message passing concurrency
• But this is a big job
• I need people to write applications
• The aim is to produce a journal paper around these results - a state of the art article on message passing concurrency
• But this is a big job
• I need people to write applications
• And do some writing
The aim is to produce a journal paper around these results - a state of the art article on message passing concurrency.

But this is a big job.

I need people to write applications.

And do some writing.

So that’s what I want people to do - work.
• The aim is to produce a journal paper around these results - a state of the art article on message passing concurrency
• But this is a big job
• I need people to write applications
• And do some writing
• So that’s what I want people to do - work
• I have a GitHub repo set up for people to join
The aim is to produce a journal paper around these results - a state of the art article on message passing concurrency. But this is a big job. I need people to write applications, and do some writing. So that’s what I want people to do - work. I have a GitHub repo set up for people to join. We’ll create an arxiv report detailing all the results.
The aim is to produce a journal paper around these results - a state of the art article on message passing concurrency.

But this is a big job.

I need people to write applications.

And do some writing.

So that’s what I want people to do - work.

I have a GitHub repo set up for people to join.

We’ll create an arxiv report detailing all the results.

And write the journal around the interesting bits.
So...

- Take out your laptop
- Pick a language
- And write commstime
- Rosetta Stone has examples of Dining Philosophers for many languages
- If you want to join the GitHub repo let me know