A Semantics For Distributed Erlang

Hans Svensson
Koen Claessen
“Common knowledge”

“Distribution is Transparent” [1]

“Message passing between a pair of processes is assumed to be ordered” [2]


procA() ->
    PidC =
        spawn(?N1, ?MODULE, procC, []),
    PidB =
        spawn(?N2, ?MODULE, procB, [PidC]),
    PidC ! hello,
    PidB ! world.

procB(PidC) ->
    receive X ->
        PidC ! X
    end.

procC() ->
    receive X ->
        ok
    end,
    receive Y ->
        ok
    end,
    io:format("~p ~p", [X,Y]).
Hello World

procA  -->  procB  -->  procC

hello

world  world

Hello World
World Hello?!!

Only when processes are on separate nodes
“Distribution is Transparent”

- Local system (one ERTS)
  - Messages are delivered instantly
  - The result is always “Hello World”

- Distributed system (many ERTSs)
  - Messages are really ‘sent’ between processes
  - Only message order between pair of processes
  - The result can be “World Hello”

Even on the same machine
Erlang Semantics

• Fredlund: Single-node semantics
  – Faithfully describes a single-node system
  – Used in model checking of Erlang software

Process communication
  Process evaluation
  Expression evaluation
Single-node process communication

Message is added directly in the receivers queue
Distributed Semantics

• Changes to existing semantics
  – Introduce the concept of nodes
  – Alter *spawn*-function
  – Restrict communication to one node

• Additions
  – Start and failure of nodes
  – Node-to-node communication
  – One intermediate mailbox per node
  – Fairness
Distributed Semantics

\[ \text{input} \quad \frac{s \xrightarrow{\text{pid?sig}} s'} {\text{nmatch}(nq, \text{from}, \text{pid}) = \text{sig}} \]

\[ \langle s, \text{node}, nq \rangle \xrightarrow{\text{pid? from} \text{ sig}} \langle s', \text{node}, nq \setminus (\text{from, pid, sig}) \rangle \]
Distributed process communication

Messages are later delivered to processes, not necessarily in order of delivery, but without breaking the order for each process-pair.
Conclusions

• Distribution is only almost Transparent
• There exist problems where a single-node semantics isn’t descriptive enough
  – Leader election implementation
• Model checking: future work
  – More accurate => Harder problem
  – Larger state space