

EPSRC

Engineering and Physical Sciences
Research Council

POETS 

Event-driven concurrent programming in POETS

David Thomas

dt10@imperial.ac.uk

Imperial College London

UKDF, May 2019

Andrew D. Brown (UoS), Simon W. Moore (Cam), Andrey Mokhov (UoN),
Matt Naylor (Cam), Jonathon Beaumont (Imp)

POETS project team



POETS: The big idea



Hardware: 10^5 concurrent threads
Application: 10^7 isolated fragments
Network: 10^9 messages / second

Asynchronous and event driven

Choose communication over computation

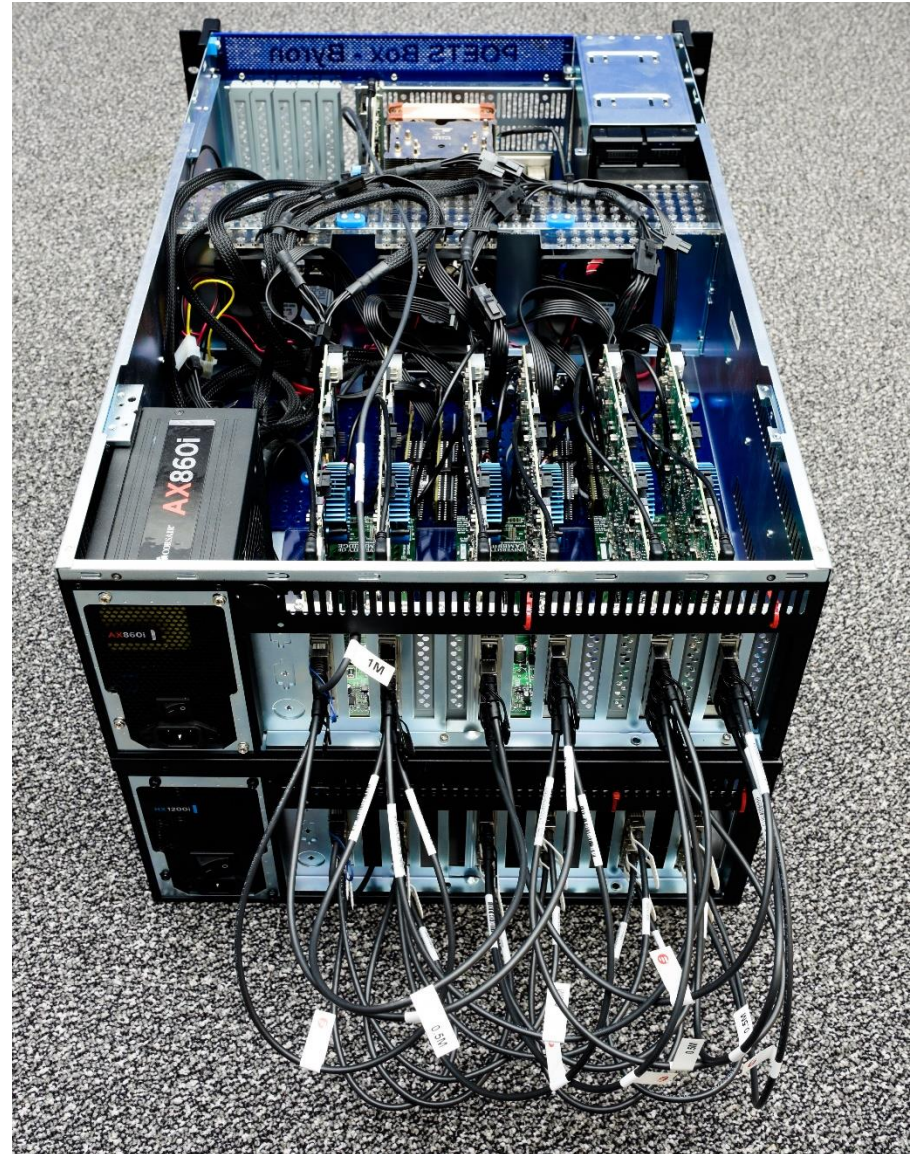
A potential niche

- Communication oriented
- Irregular
- Sparse
- Highly concurrent

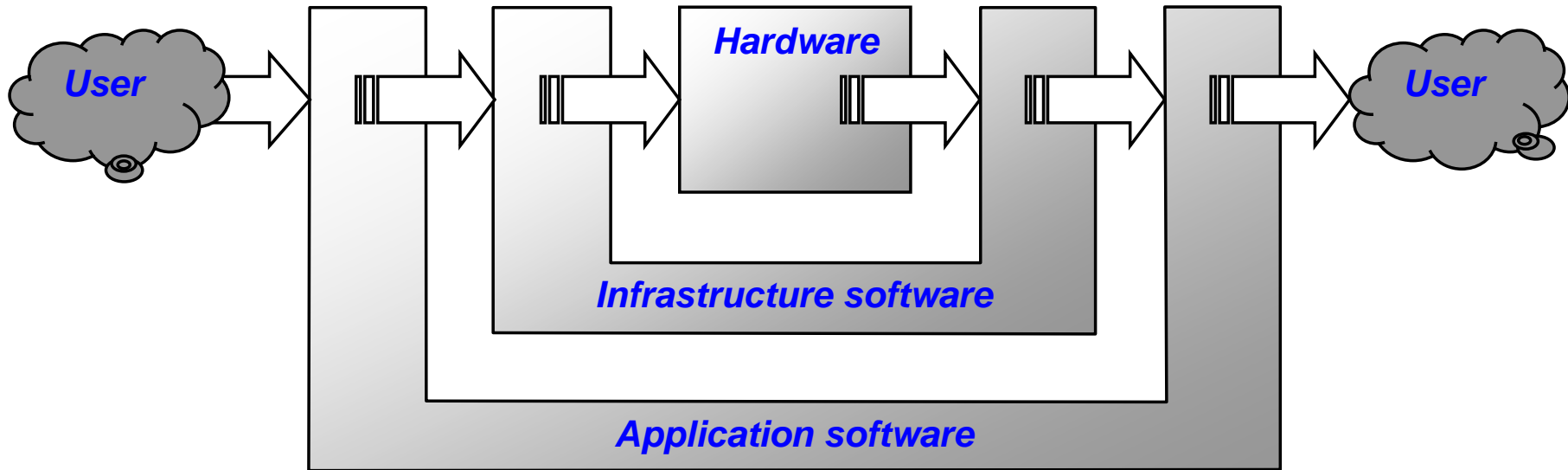
- Tinsel is a new RISC-V multi-threaded core
 - 9-stage in-order pipeline (no forwarding)
 - Switches threads to hide cache-misses
- Heavily optimised for LUT-based FPGAs
 - Can fit 250 cores in half a Stratix-V
 - 1 MIPS/LUT or 100 GOps/Sec/FPGA

Hardware: Multi-FPGA

- Box: 6 FPGAs
 - 6K threads/box
- Cluster: 4 boxes
 - 24K threads/cluster
- Next generation
 - 256K threads/cluster



Information Flow



- Applications are split into ***devices***
 - “device” = finite state machine
- Device state is a tiny part of the global state
 - ***Only*** the device can read and write it’s state
 - No shared memory – only messages

Device event handlers

Receive: a message m is sent to device d

$$d' = \text{receive_handler}(d, m)$$

Send: device d sends a message m

$$(d', m) = \text{send_handler}(d)$$

State changes only occur on send or receive

All state changes are atomic

Hardware schedules both sends and receives

Graph Processing

