NICTA

Improving the Dependability of Device Drivers

Gernot Heiser

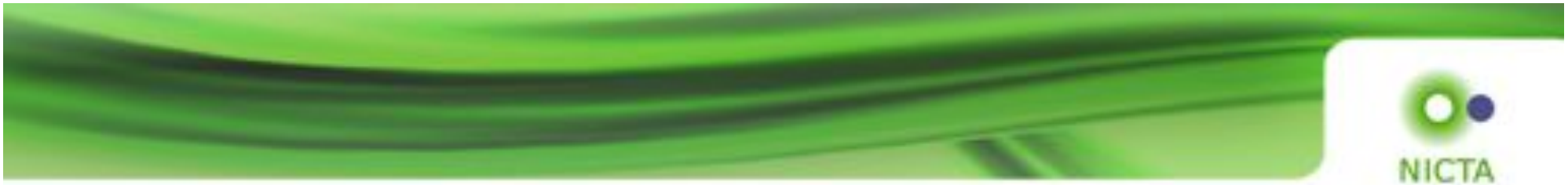
John Lions Professor of Operating Systems, University of New South Wales

Leader, Trustworthy Embedded Systems, NICTA

CTO and Founder, Open Kernel Labs

The Problem with Drivers

- **70%** of OS crashes are caused by device drivers
  [Ganapathi et al. Windows XP kernel crash analysis, 2006]
- Drivers contain **1.5x-7x** bugs per LoC compared to the rest of the kernel
  [Chou et al. An Empirical study of operating system errors, 2001]



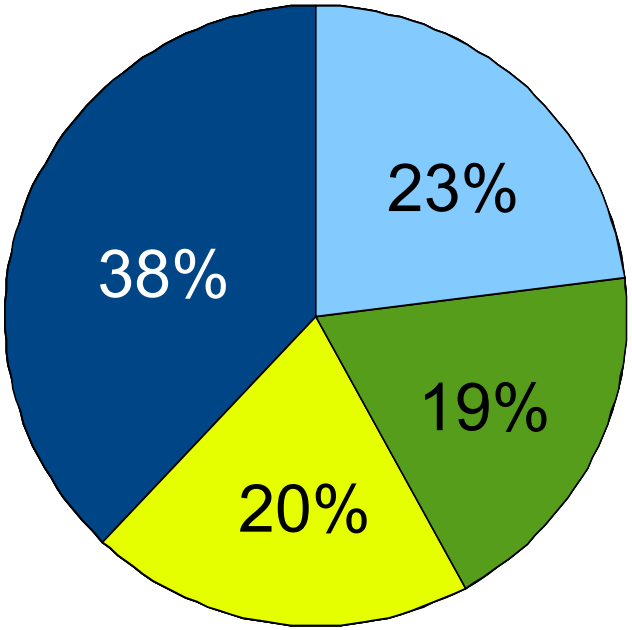
A Study of Linux Driver Bugs

Linux Driver Bugs

NICTA

Driver	#loc	#bugs
USB		
RTL8150 USB-to-Ethernet adapter	827	16
EL1210a USB-to-Ethernet adapter	710	2
KL5kusb101 USB-to-Ethernet apapter	925	15
Generic USB network driver	1028	45
USB hub	2234	67
USB-to-serial converter	989	50
USB mass storage	803	23
Firewire		
IEEE 1394 Ethernet controller	1413	22
SBP-2 transport protocol	1713	46
PCI		
Mellanox Infi niHost Infi niBand adapter	11718	123
BNX2 Ethernet adapter	5412	51
i810 frame buff er	2920	16
CMI8338 audio	2660	22
		498

Bugs by Category



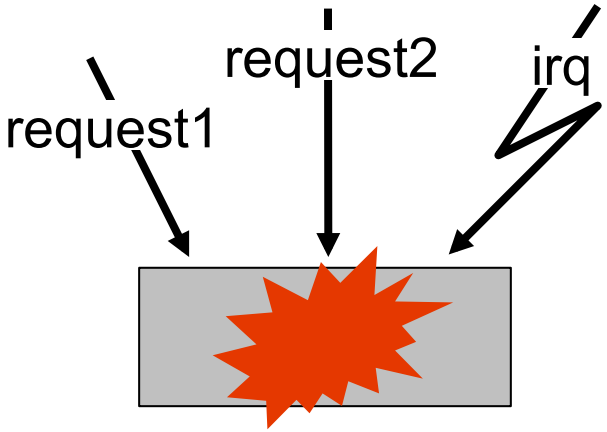
- Device protocol violations
- OS protocol violations
- Concurrency errors
- Generic errors

Eliminating Concurrency Bugs

Event-Driven Device Drivers

NICTA

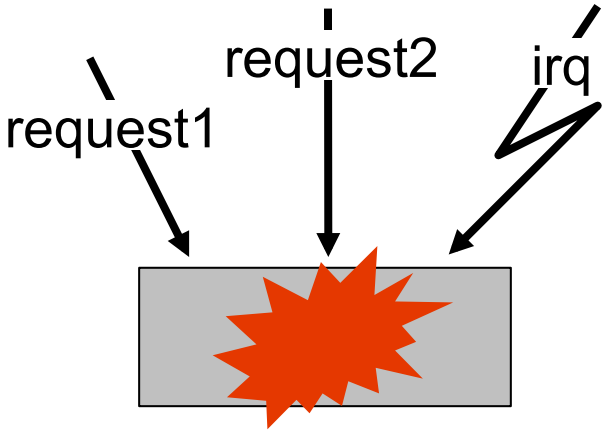
Threads



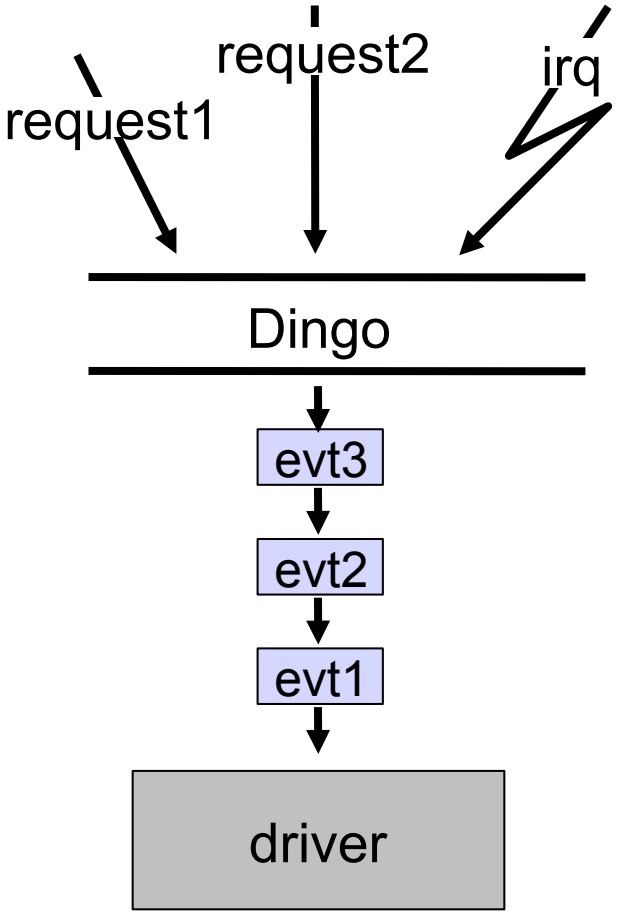
Event-Driven Device Drivers

NICTA

Threads



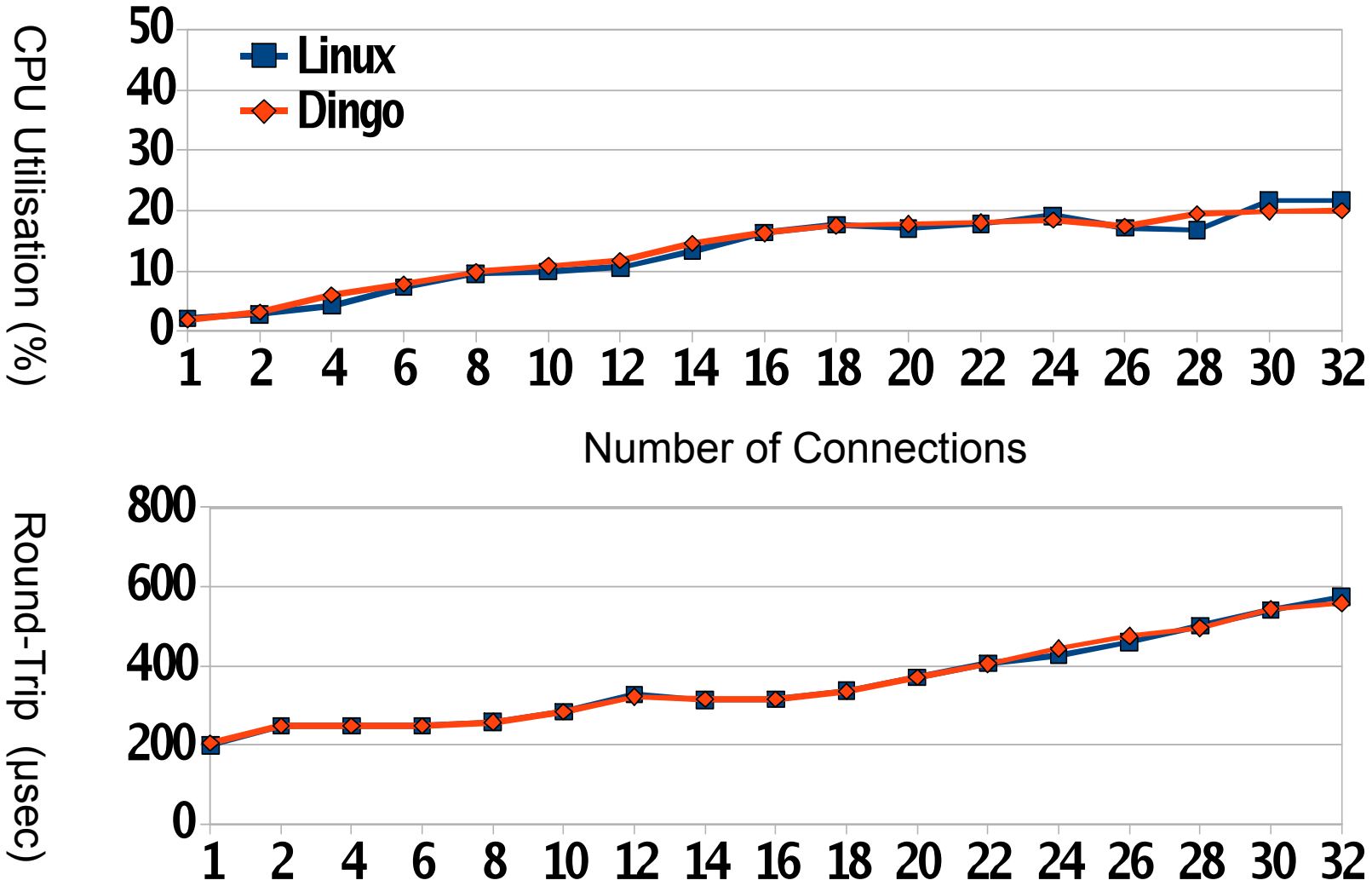
Events



Performance of the AX88772 USB-to-Ethernet Adapter Driver

NICTA

Evaluation platform: 4 x 2GHz Itanium II (SMT, 2 threads per core)



Impact of Serialisation on Performance

Special case: drivers for very-high-performance devices

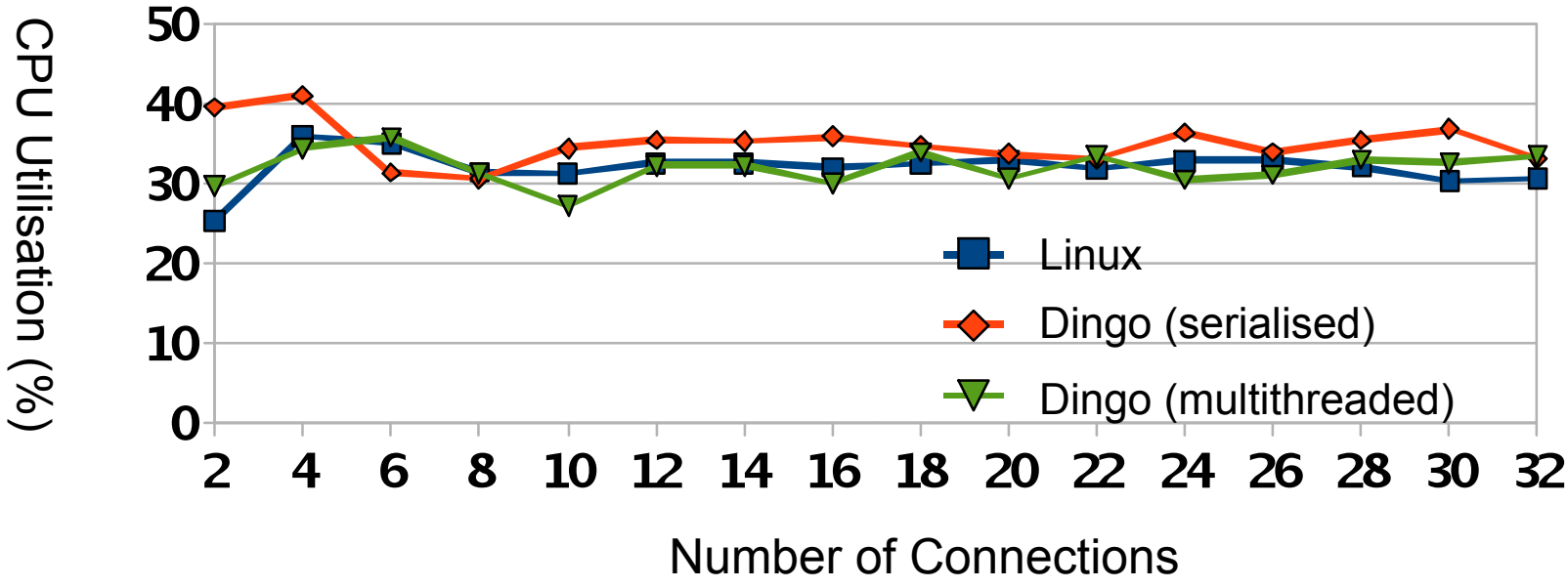
- Examples: 10Gb Ethernet, Infiniband
- For such drivers, serialisation affects performance on multiprocessors

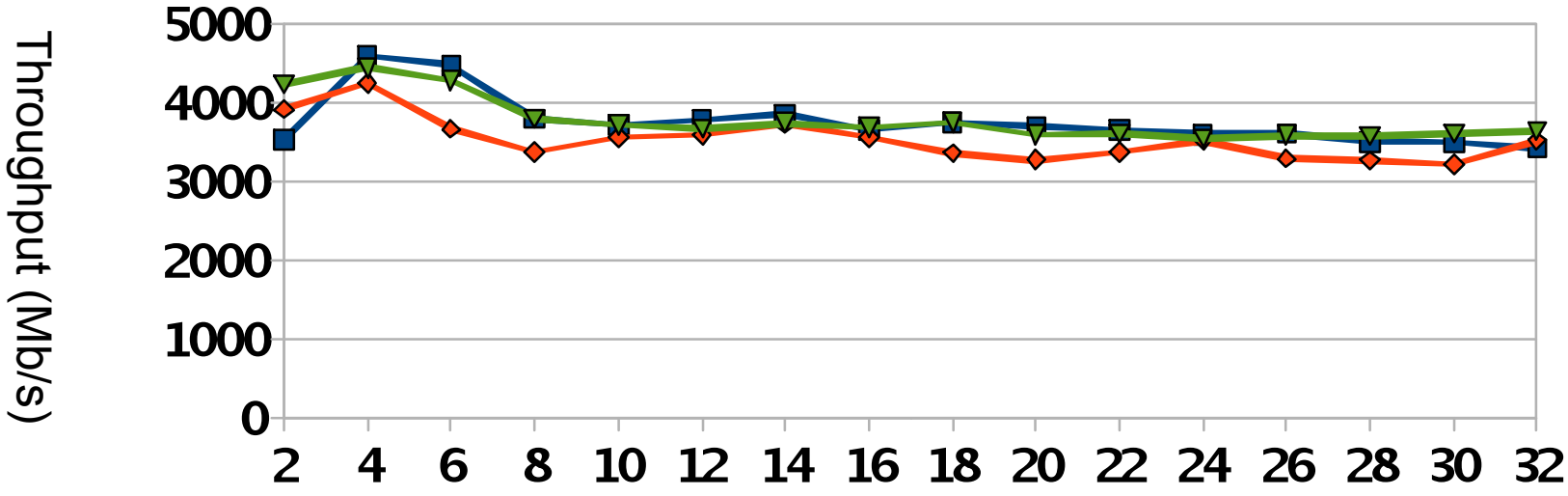
Solution: Re-introduce multithreading at the data path

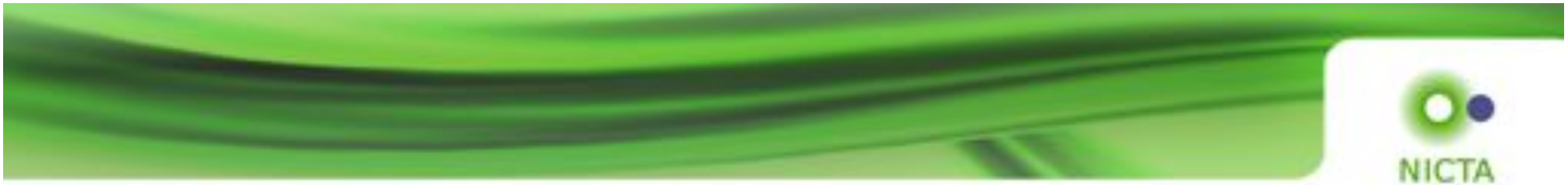
- Avoid concurrency bugs at the control path, while maintaining high performance at the data path

Performance of the Mellanox InfiniBand Adapter Driver

NICTA

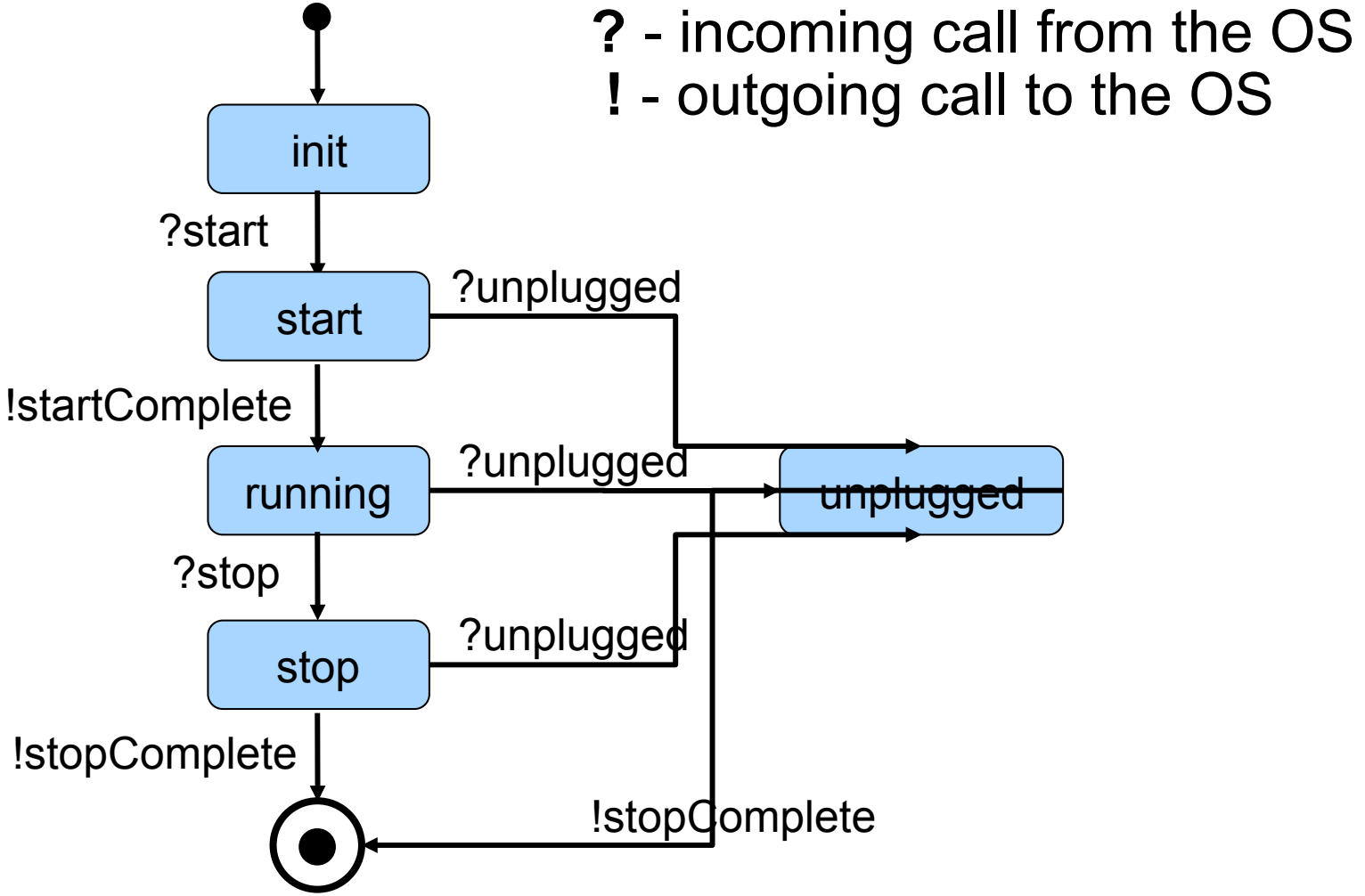




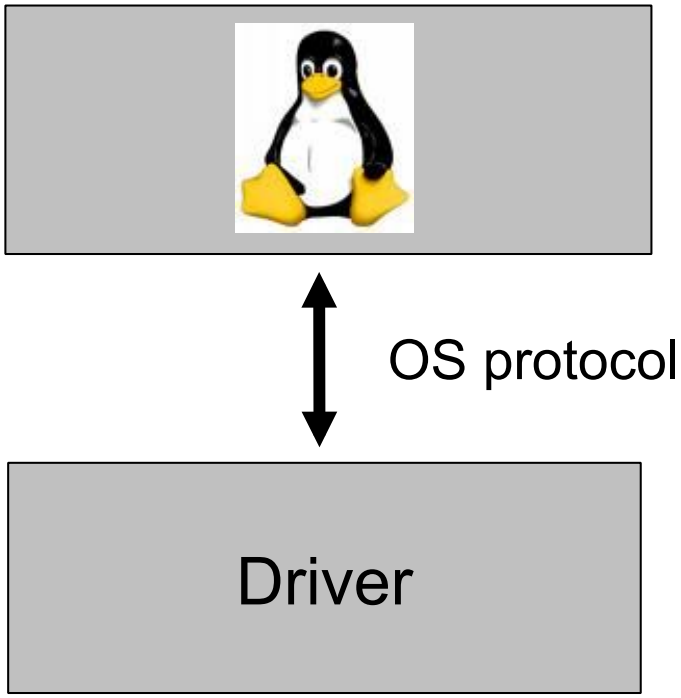


Dealing with OS Protocol Violations

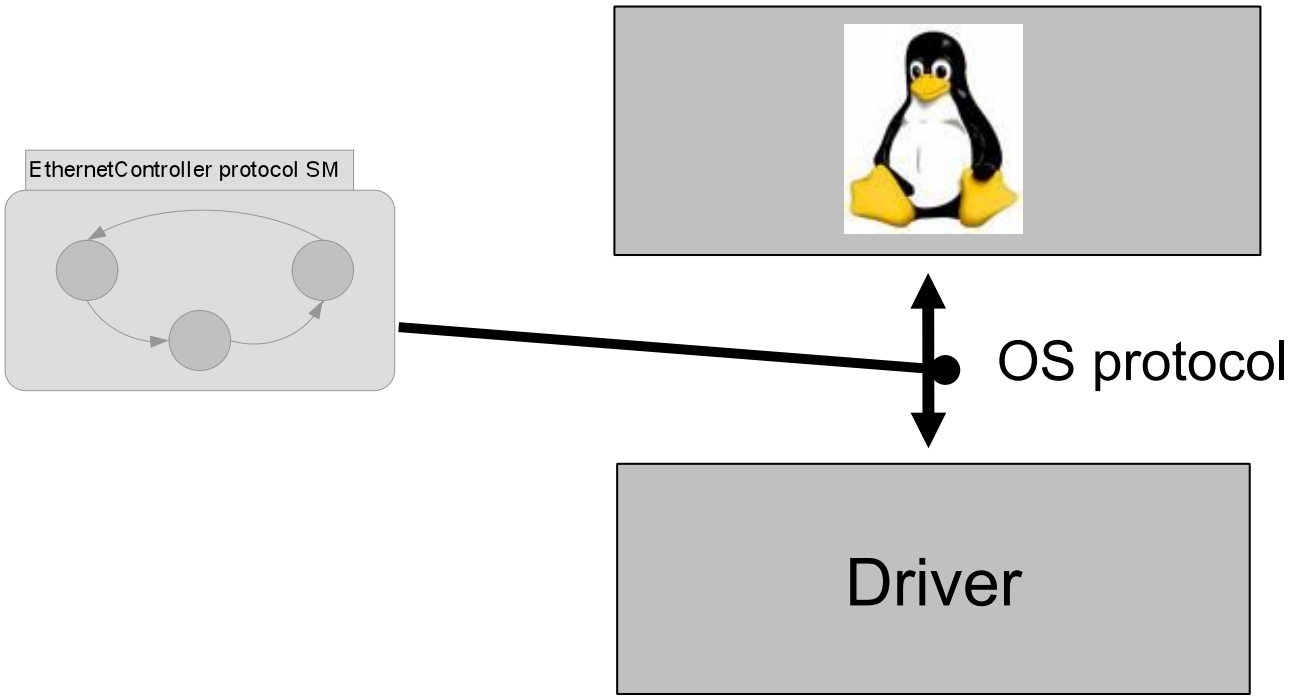
Modeling Driver Protocols with State Machines



Runtime Failure Detection



Runtime Failure Detection



Evaluation

Evaluation

NICTA

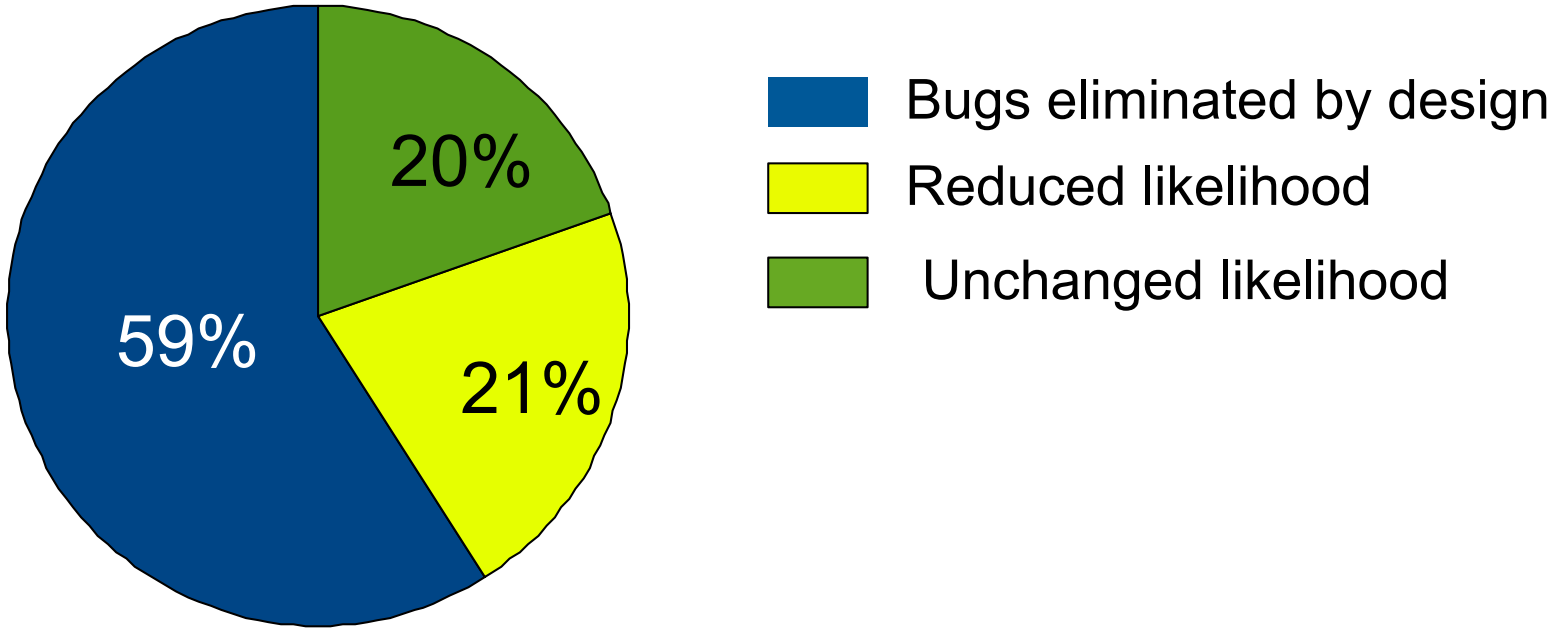
How effective is Dingo in reducing driver bugs?

- Evaluation methodology: artificially injected **61** bugs found in similar Linux drivers into Dingo drivers

Evaluation

How effective is Dingo in reducing driver bugs?

- Evaluation methodology: artificially injected **61** bugs found in similar Linux drivers into Dingo drivers

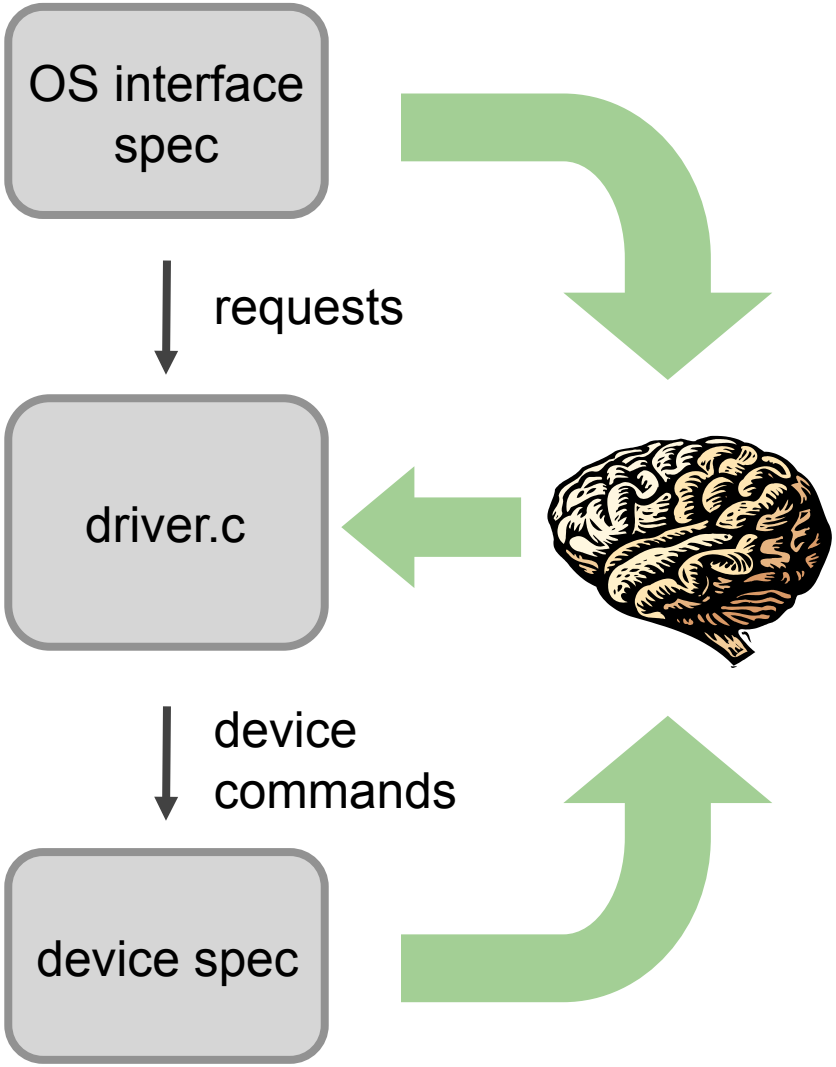


Summary

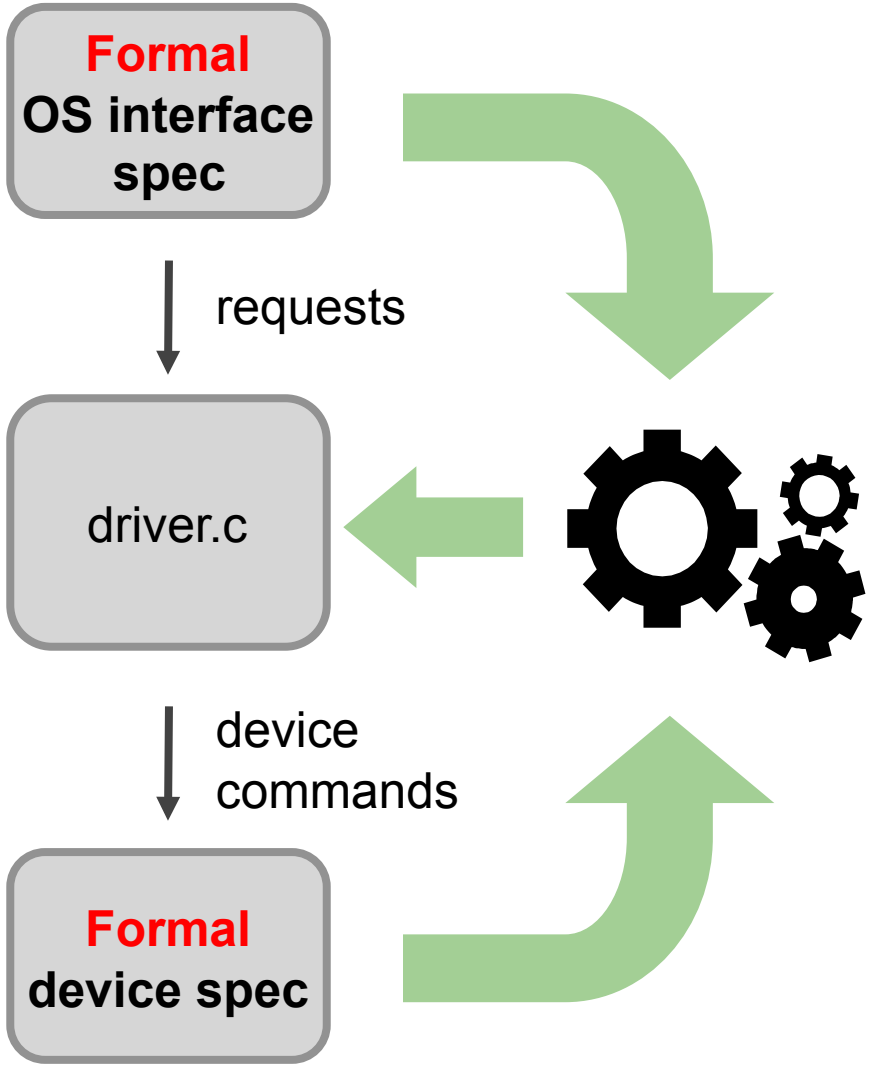
- 40% of driver bugs are caused by the complexity of the OS interface
- Dingo significantly reduces these bugs through an improved design of this interface
- These improvements are implemented in an existing operating system without sacrificing the performance
- Dingo drivers can co-exist with legacy drivers
- Working on pushing Dingo support into Linux mainstream

Automatic Generation of Device Drivers

Conventional Driver Development



Driver Synthesis: High-Level View



Advantages

- Separation of concerns
  - Know one thing well
- Reuse
  - Specify once, synthesise many

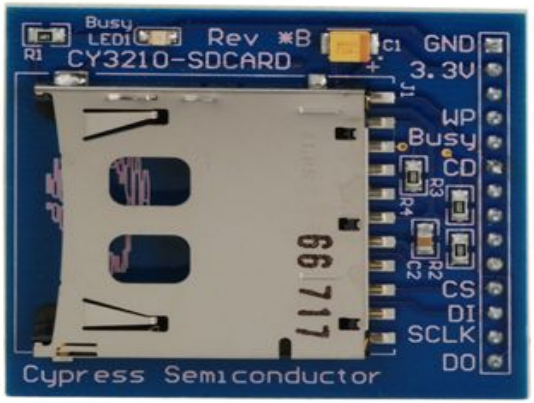
Synthesising Real Drivers

Synthesis algorithm adapted from game theory

- Issue: The state explosion problem
  - **Problem:** The product state space can be huge
  - **Solution:** Explore the product state space incrementally
- Issue: Dealing with data
  - **Problem:** Enumerating all variable assignments is infeasible
  - **Solution:** Manipulate data symbolically

Results

- Successfully synthesised drivers for real devices:
  - Asix AX88772 USB-to-Ethernet adapter
    - Linux
  - Ricoh R5C822 SD host controller
    - Linux
    - FreeBSD



Results

	USB-to-Ethernet	SD
OS interface spec	309 loc	641 loc
Device spec	463 loc	653 loc
Synthesised driver	2620 loc	4667 loc
Linux driver	1200 loc	1174 loc

Results

NICTA

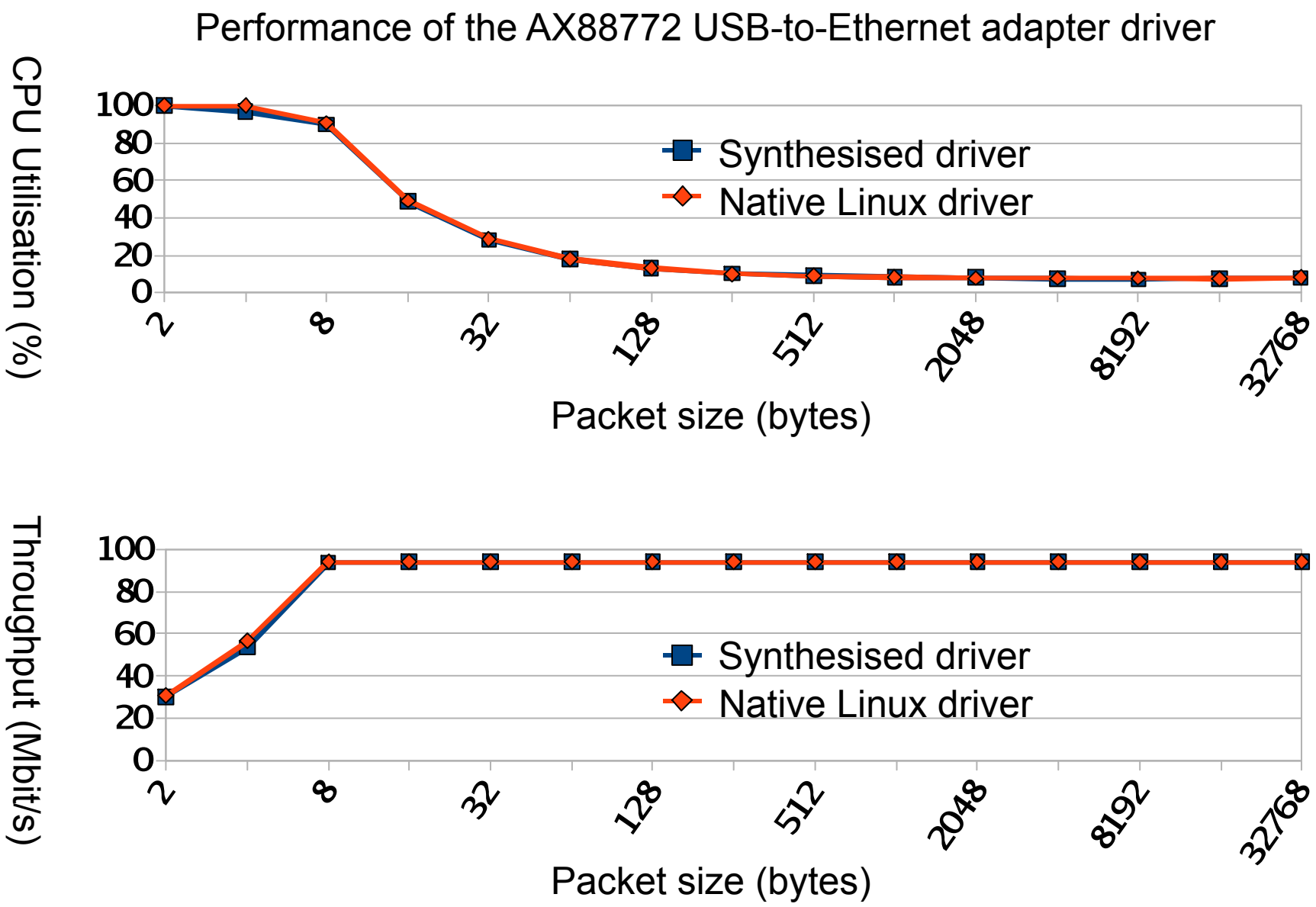
	USB-to-Ethernet	SD
OS interface spec	309 loc	641 loc
Device spec	463 loc	653 loc
Synthesised driver	2620 loc	4667 loc
Linux driver	1200 loc	1174 loc

Results

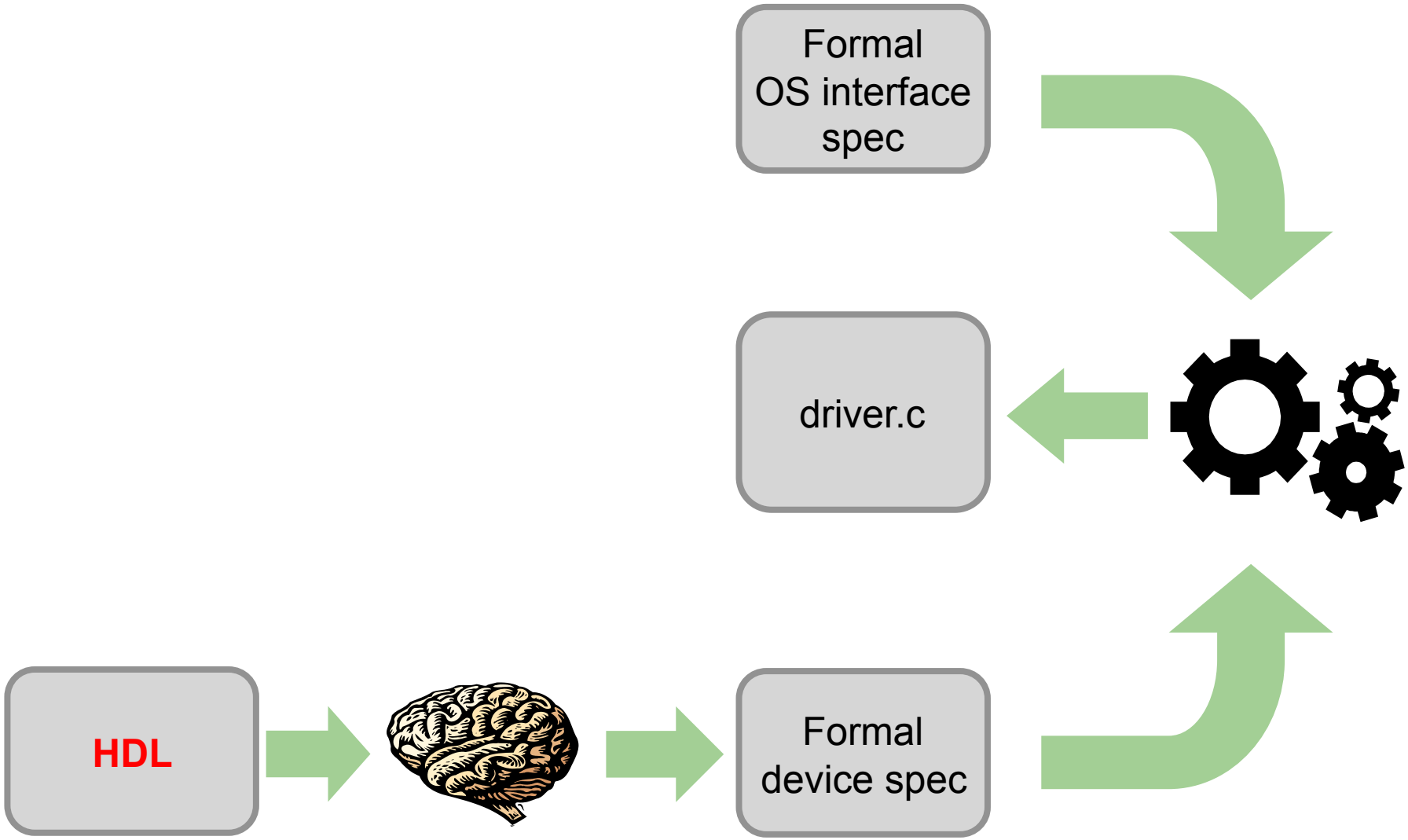
NICTA

	USB-to-Ethernet	SD
OS interface spec	309 loc	641 loc
Device spec	463 loc	653 loc
Synthesised driver	2620 loc	4667 loc
Linux driver	1200 loc	1174 loc

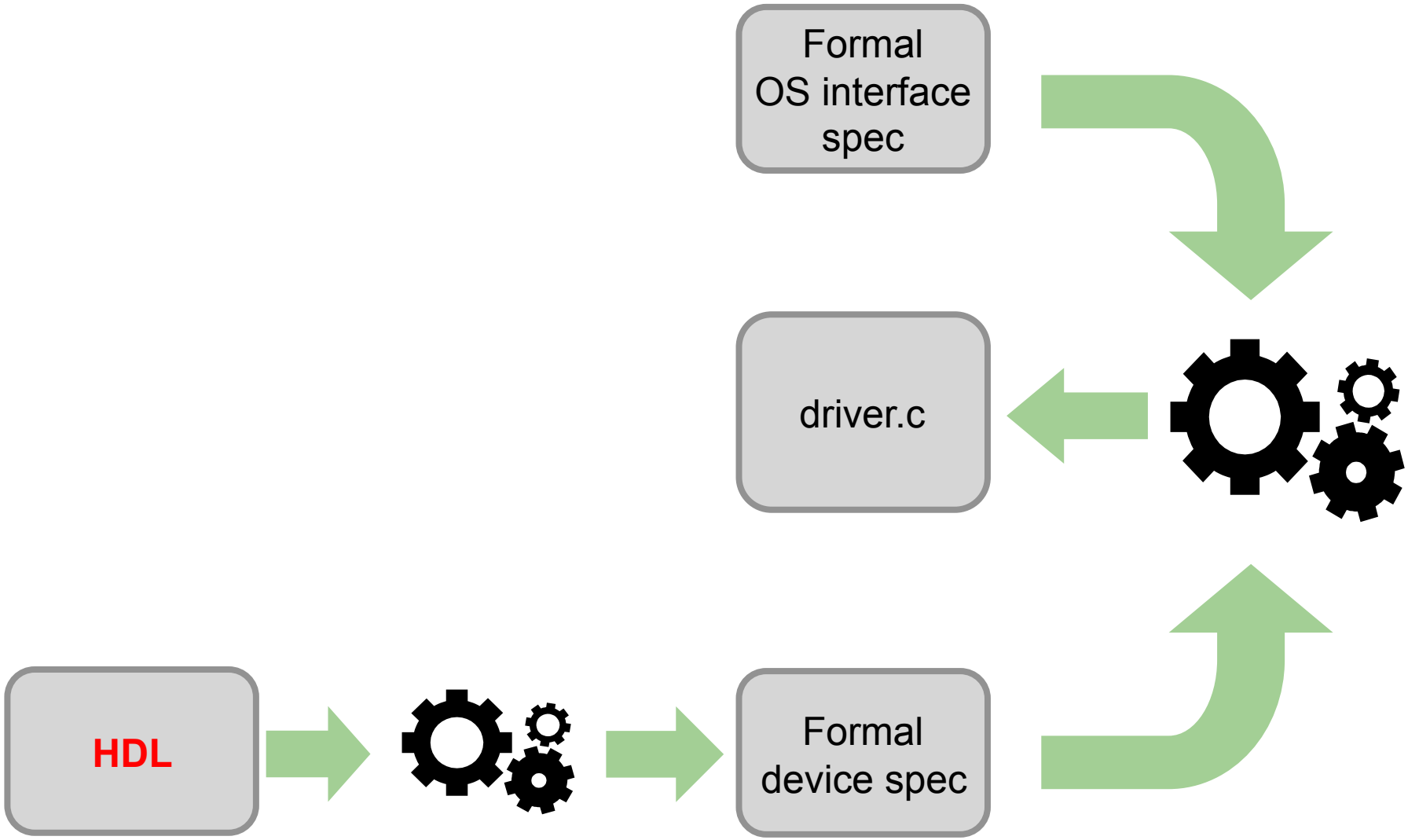
Performance



Future work



Future work



Conclusions

- Driver synthesis is possible
  - Device experts provide device specs
  - OS experts provide OS specs
  - **Termite does the rest**
- Still work-in-progress
  - Addressing current limitations
  - Driver synthesis from HDL
- Details:

  Ryzhyk, Chubb, Kuz, Heiser, 4th EuroSys, Apr 2009

  Ryzhyk, Chubb, Kuz, Le Sueur, Heiser, 22nd SOSP, Oct 2009