Hardware–Software Co-Design:
Not Just a Cliché

Adrian Sampson
James Bornholt
Luis Ceze

University of Washington

SNAPL 2015

(not to scale)

free lunch

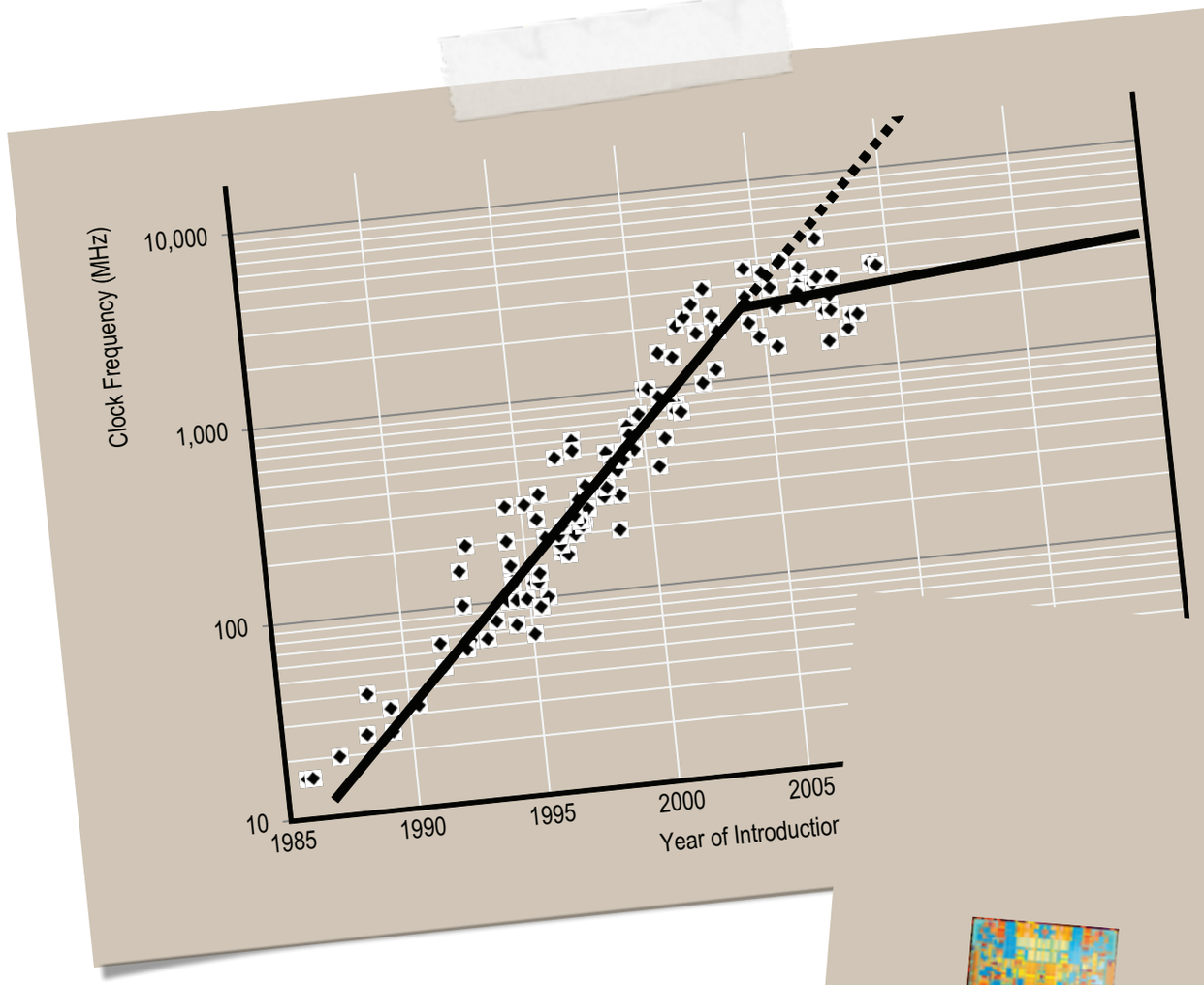
time immemorial

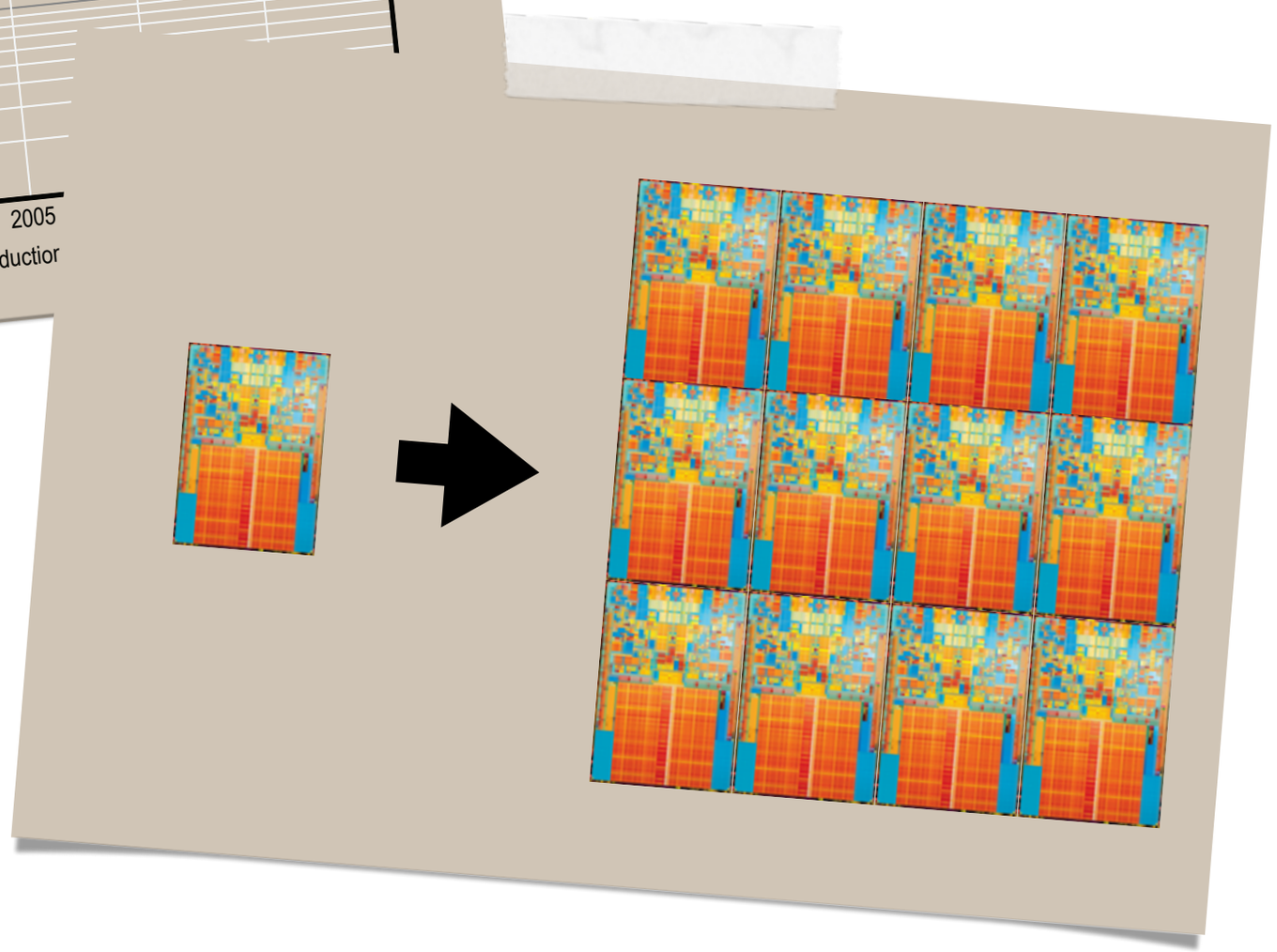
2005

2015

exponential single-threaded performance scaling!

(not to scale)


Clock Frequency (MHz)
10,000
1,000
100
10
1985
1990
1995
2000
2005
Year of Introductior



free lunch

multicore era

time immemorial

2005

2015

we'll scale the number of cores instead

The multicore transition was a **stopgap,** not a panacea.

free lunch **multicore era** **who knows?**

time immemorial

2005

2015

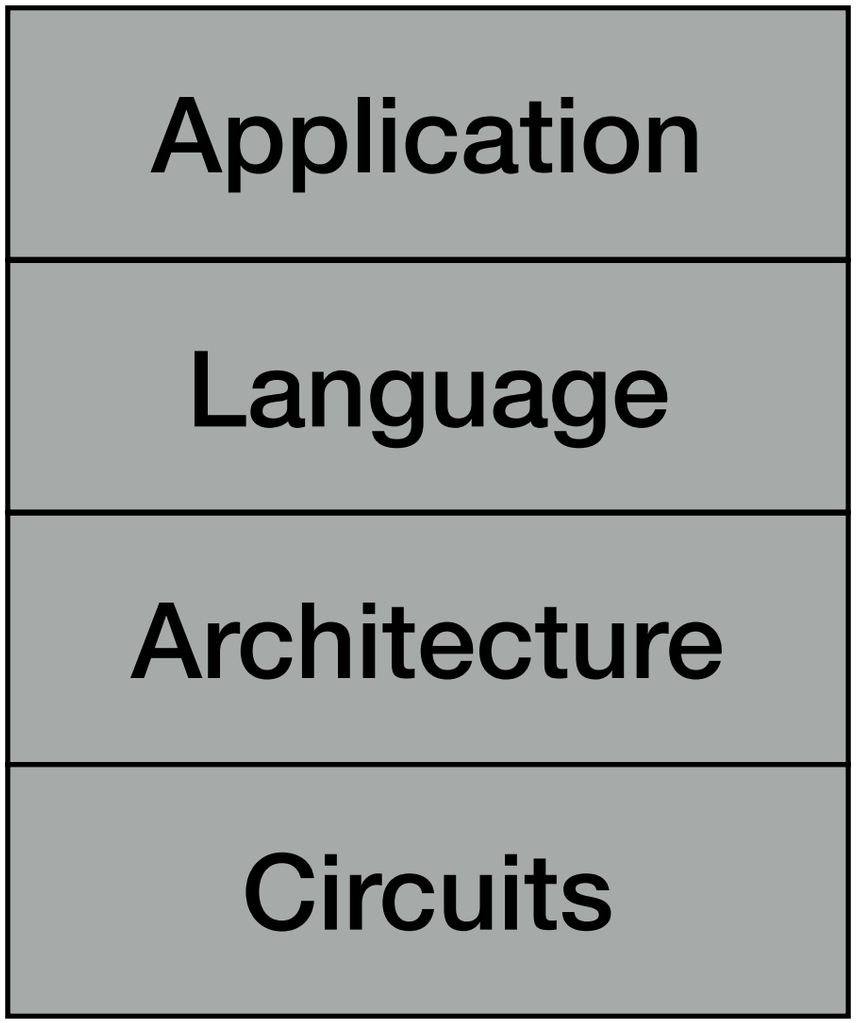
?

?

?

?

?


Application
Language
Architecture
Circuits

Application

Language

hardware–software abstraction boundary

parallelism

data movement

Architecture

Circuits

guard bands

energy costs

Application

Language

parallelism

data movement

hardware-software abstraction boundary

guard bands

energy costs

Architecture

Circuits

lessons learned from

Approximate Computing

New Opportunities

for hardware–software co-design

lessons learned from

Approximate Computing

New Opportunities

for hardware–software co-design

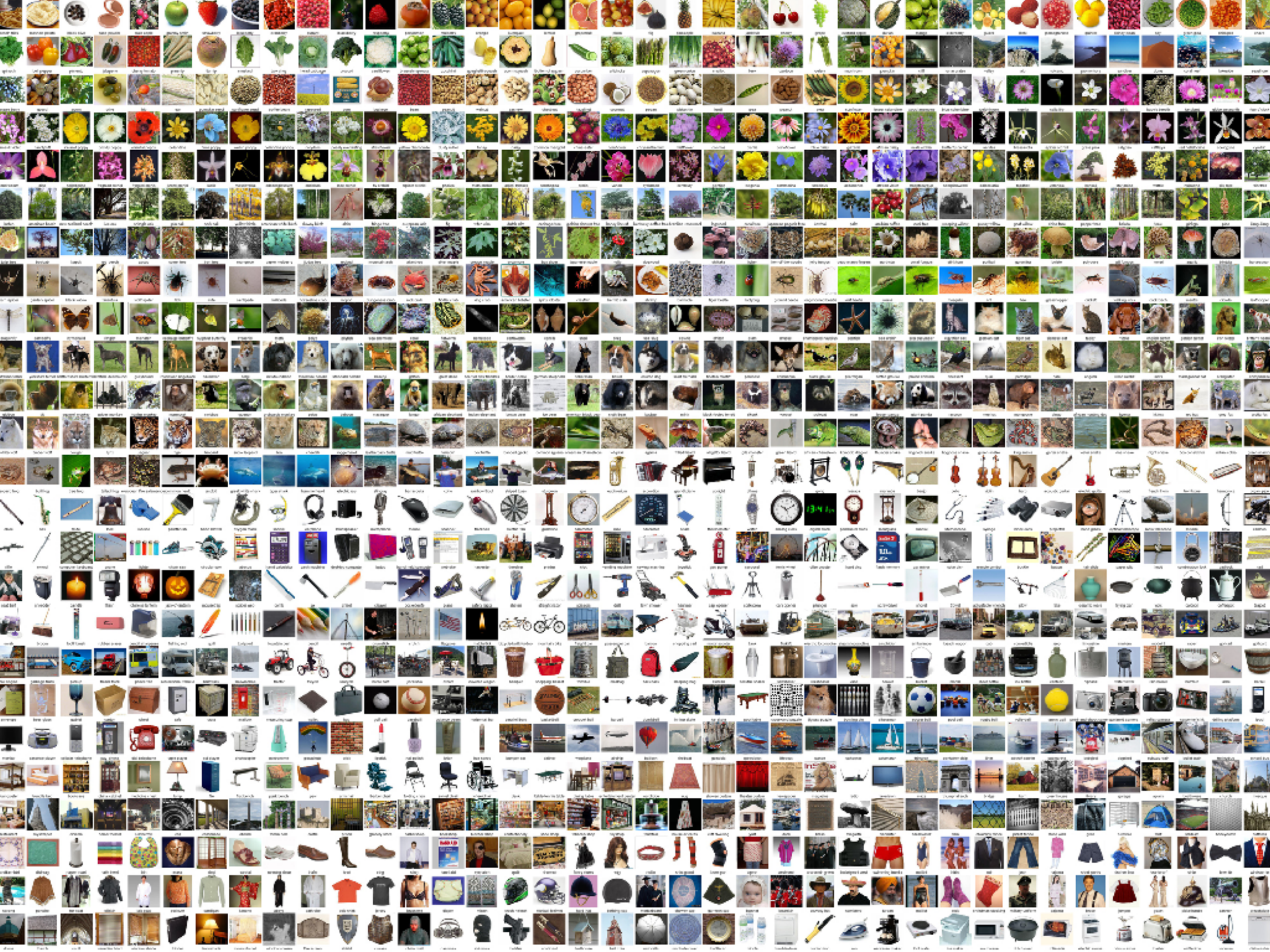
Application

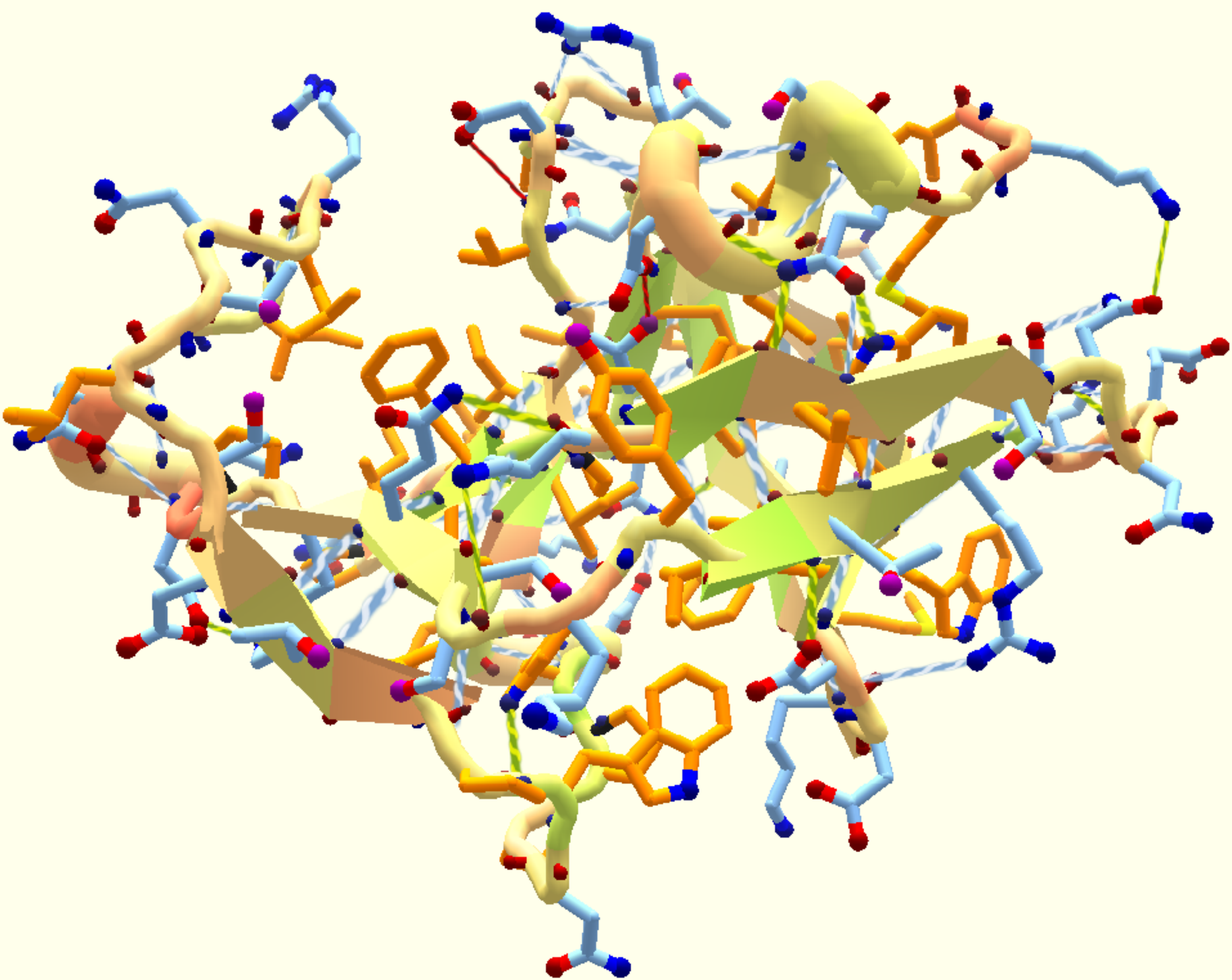
Language

new abstractions for incorrectness

Architecture

Circuits





type systems

debuggers

Application

Language

probabilistic guarantees

auto-tuning

new abstractions for incorrectness

flaky functional units

lossy cache compression

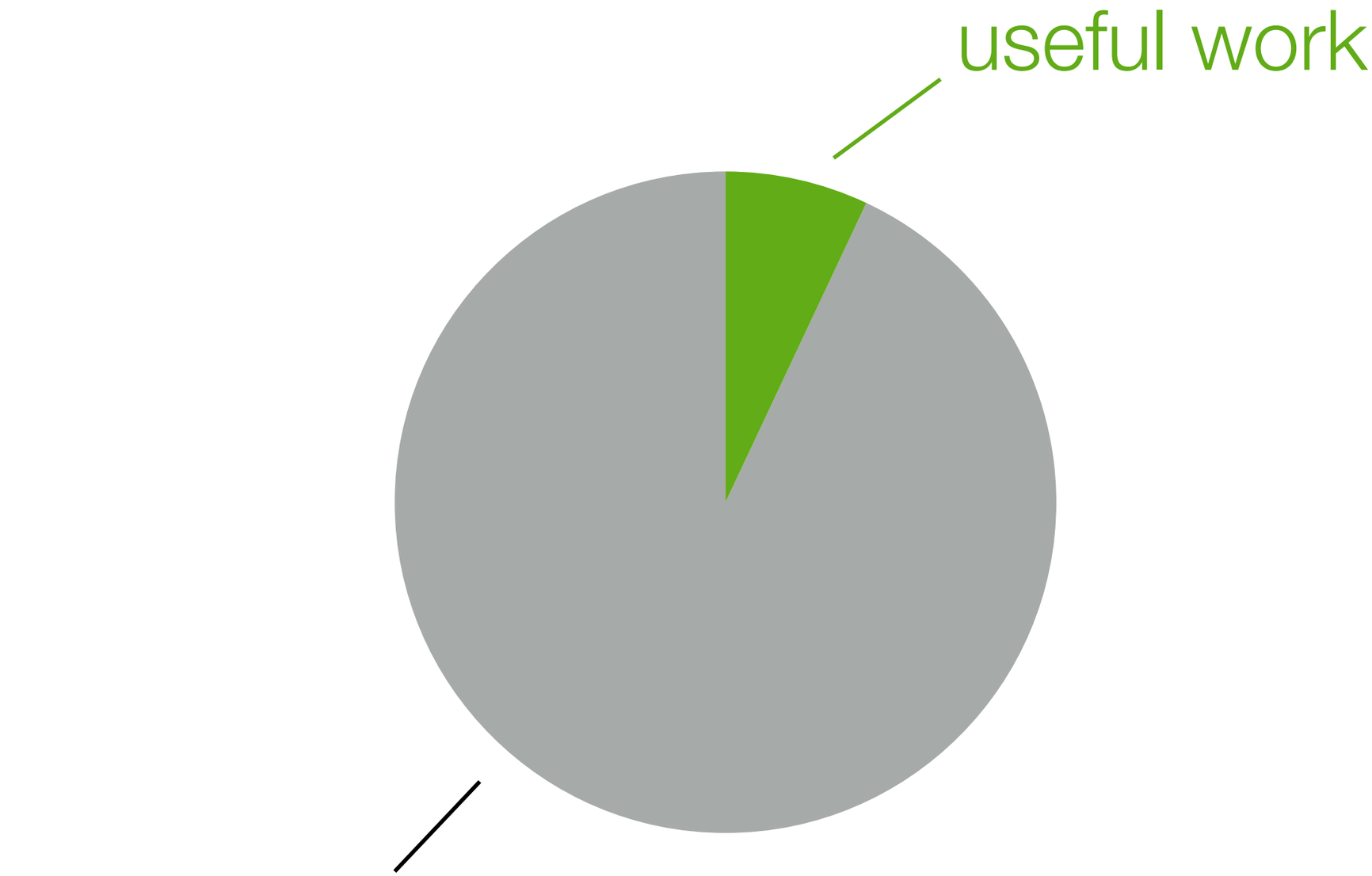
Architecture

Circuits

neural acceleration

drowsy SRAMs

The von Neumann curse



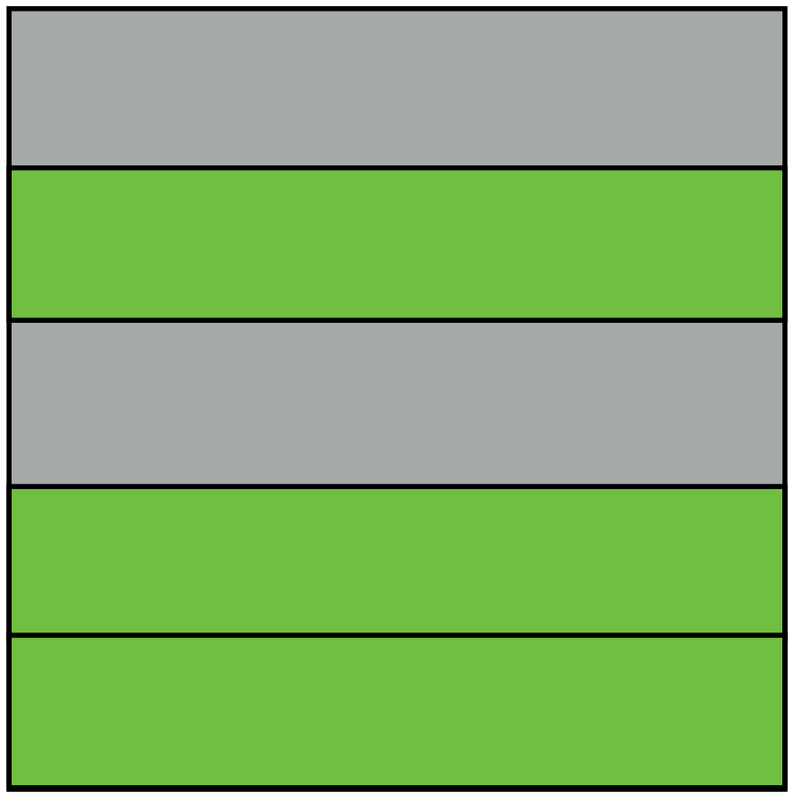
Hardware design costs sanity & well-being

Thierry Moreau,
FPGA design champion

[Moreau et al.; HPCA 2015]

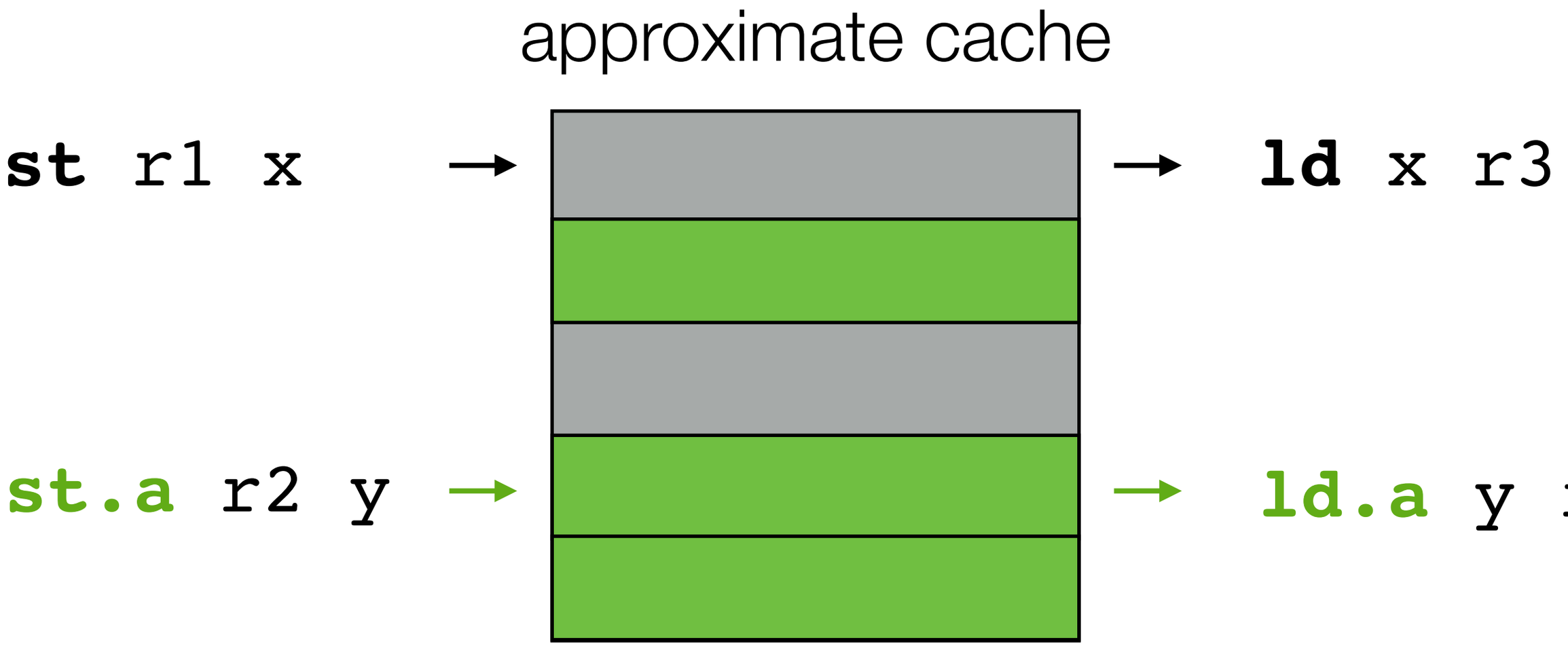
Trust your compiler

approximate cache



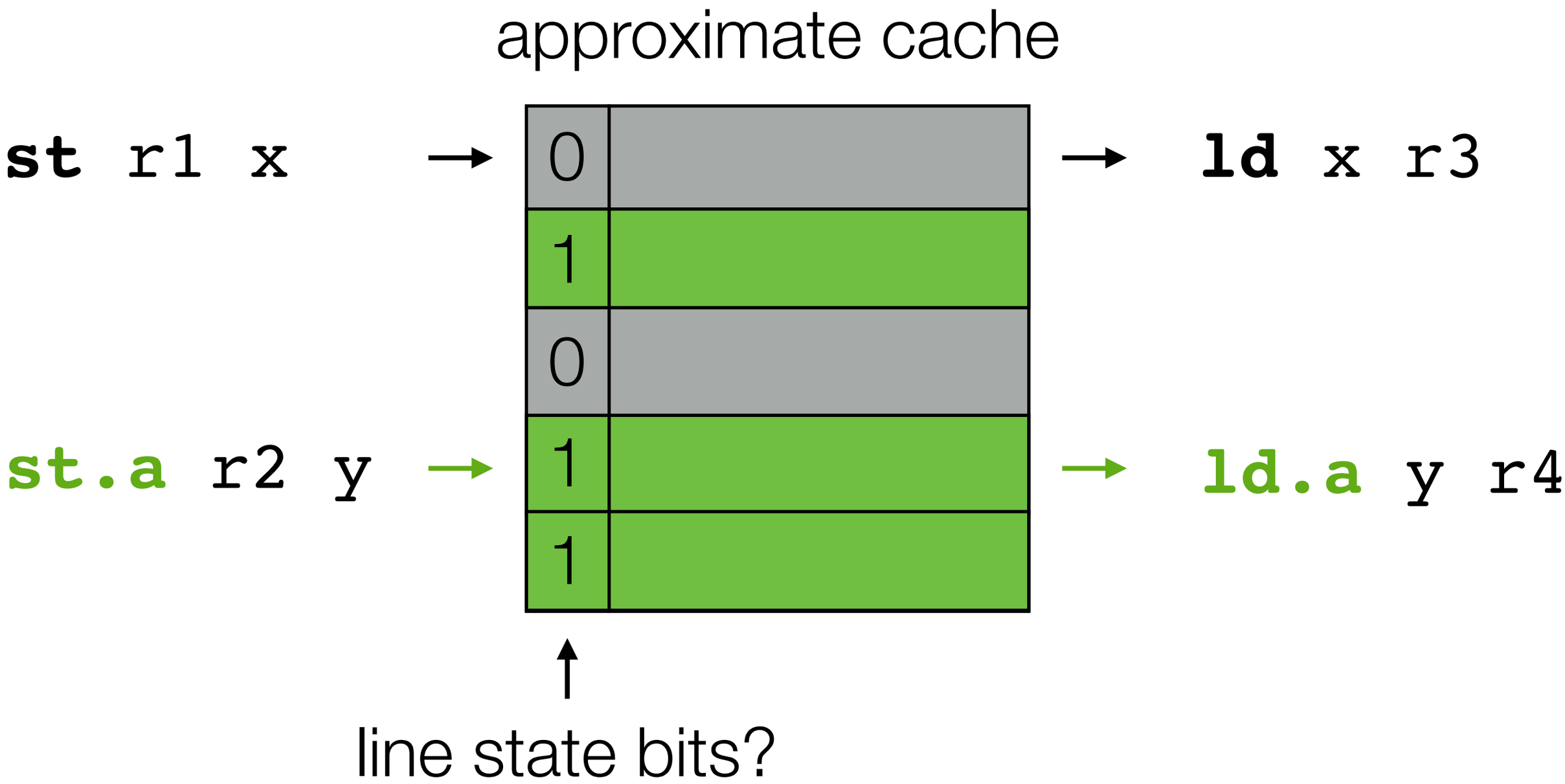
[Esmaeilzadeh, Sampson, Ceze, Burger; ASPLOS 2012]

Trust your compiler



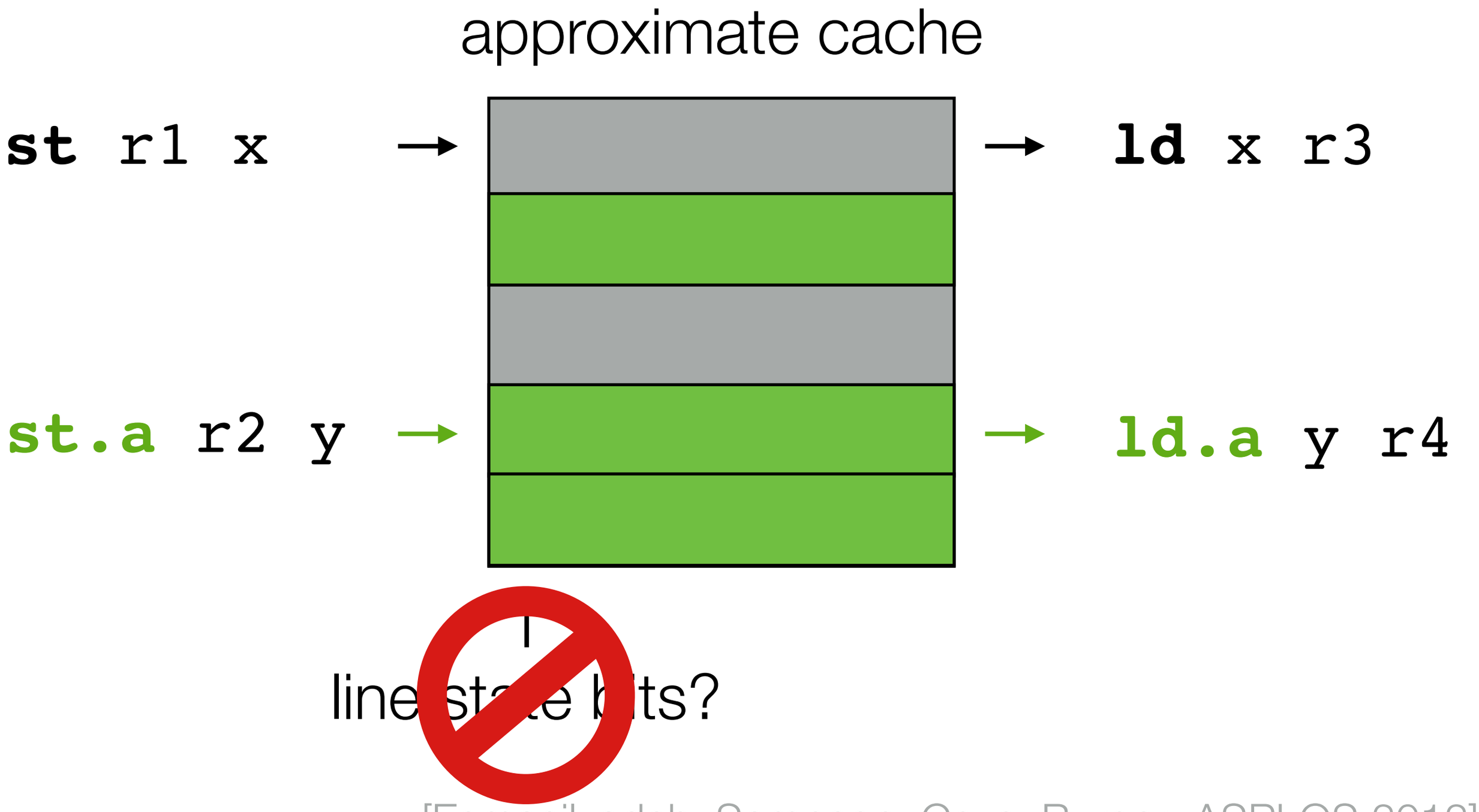
[Esmaeilzadeh, Sampson, Ceze, Burger; ASPLOS 2012]

Trust your compiler



[Esmaeilzadeh, Sampson, Ceze, Burger; ASPLOS 2012]

Trust your compiler



[Esmaeilzadeh, Sampson, Ceze, Burger; ASPLOS 2012]

lessons learned from

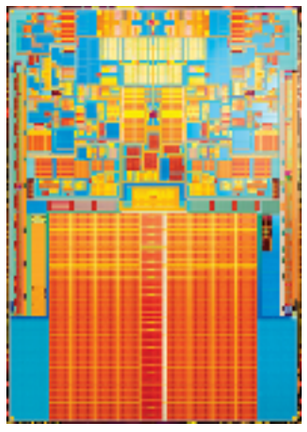
Approximate Computing

New Opportunities

for hardware–software co-design

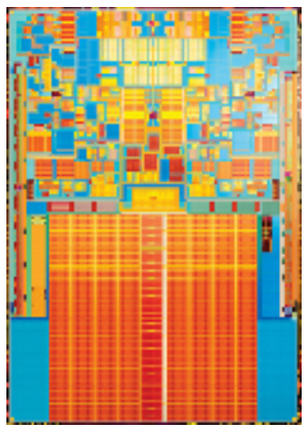
More hardware flexibility that humans can actually program

More hardware flexibility that humans can actually program



FPGA

More hardware flexibility that humans can actually program



FPGA

explicit data movement

explicit memory blocks

explicit physical routing

explicit clock frequency

explicit ILP

explicit numeric bit width

More hardware flexibility that humans can actually program

A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services

Andrew Putnam Adrian M. Caulfield Eric S. Chung Derek Chiou[1]
Kypros Constantinides[2] John Demme[3] Hadi Esmaeilzadeh[4] Jeremy Fowers
Gopi Prashanth Gopal Jan Gray Michael Haselman Scott Hauck[5] Stephen Heil
Amir Hormati[6] Joo-Young Kim Sitaram Lanka James Larus[7] Eric Peterson
Simon Pope Aaron Smith Jason Thong Phillip Yi Xiao Doug Burger

Microsoft

Abstract

Datacenter workloads demand high computational capabilities, flexibility, power efficiency, and low cost. It is challenging to improve all of these factors simultaneously. To advance datacenter capabilities beyond what commodity server designs can provide, we have designed and built a composable, reconfigurable fabric to accelerate portions of large-scale software services. Each instantiation of the fabric consists of a 6x8 2-D torus of high-end Stratix V FPGAs embedded into a half-rack of 48 machines. One FPGA is placed into each server, accessible through PCIe, and wired directly to other FPGAs with pairs of 10 Gb SAS cables.

In this paper, we describe a medium-scale deployment of this fabric on a bed of 1,632 servers, and measure its efficacy in accelerating the Bing web search engine. We describe the requirements and architecture of the system, detail the

desirable to reduce management issues and to provide a consistent platform that applications can rely on. Second, datacenter services evolve extremely rapidly, making non-programmable hardware features impractical. Thus, datacenter providers are faced with a conundrum: they need continued improvements in performance and efficiency, but cannot obtain those improvements from general-purpose systems.

Reconfigurable chips, such as Field Programmable Gate Arrays (FPGAs), offer the potential for flexible acceleration of many workloads. However, as of this writing, FPGAs have not been widely deployed as compute accelerators in either datacenter infrastructure or in client devices. One challenge traditionally associated with FPGAs is the need to fit the accelerated function into the available reconfigurable area. One could virtualize the FPGA by reconfiguring it at run-time to support more functions than could fit into a single device. However, current reconfiguration times for standard FPGAs

More hardware flexibility that humans can actually program

A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services

Andrew Putnam Adrian M. Caulfield Eric S. Chung Derek Chiou[1]
Kypros Constantinides[2] John Demme[3] Hadi Esmaeilzadeh[4] Jeremy Fowers
Gopi Prashanth Gopal Jan Gray Michael Haselman Scott Hauck[5] Stephen Heil
Amir Hormati[6] Joo-Young Kim Sitaram Lanka James Larus[7] Eric Peterson
Simon Pope Aaron Smith Jason Thong Phillip Yi Xiao Doug Burger

Microsoft

23 authors!

Abstract

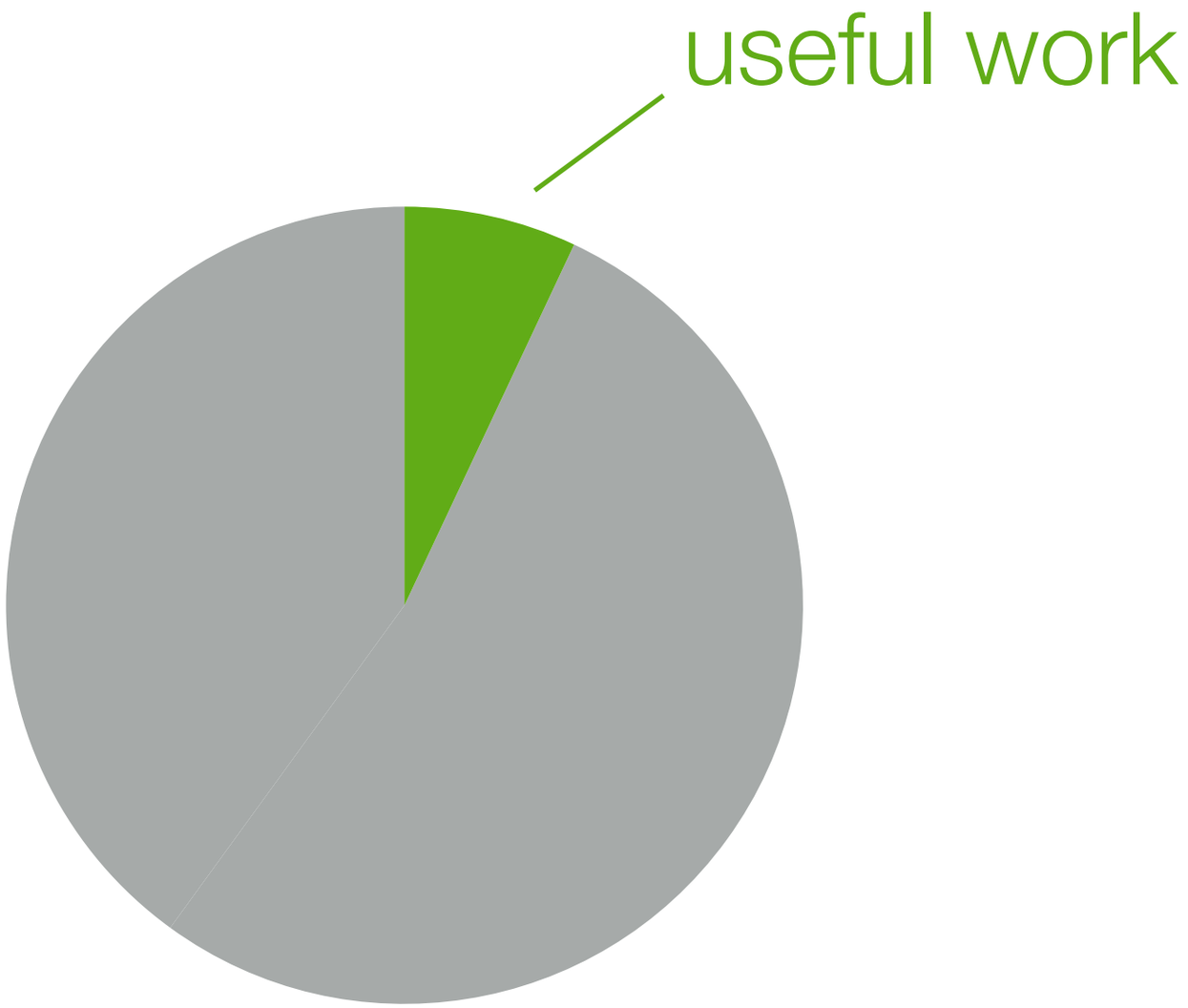
Datacenter workloads demand high computational capabilities, flexibility, power efficiency, and low cost. It is challenging to improve all of these factors simultaneously. To advance datacenter capabilities beyond what commodity server designs can provide, we have designed and built a composable, reconfigurable fabric to accelerate portions of large-scale software services. Each instantiation of the fabric consists of a 6x8 2-D torus of high-end Stratix V FPGAs embedded into a half-rack of 48 machines. One FPGA is placed into each server, accessible through PCIe, and wired directly to other FPGAs with pairs of 10 Gb SAS cables.

In this paper, we describe a medium-scale deployment of this fabric on a bed of 1,632 servers, and measure its efficacy in accelerating the Bing web search engine. We describe the requirements and architecture of the system, detail the

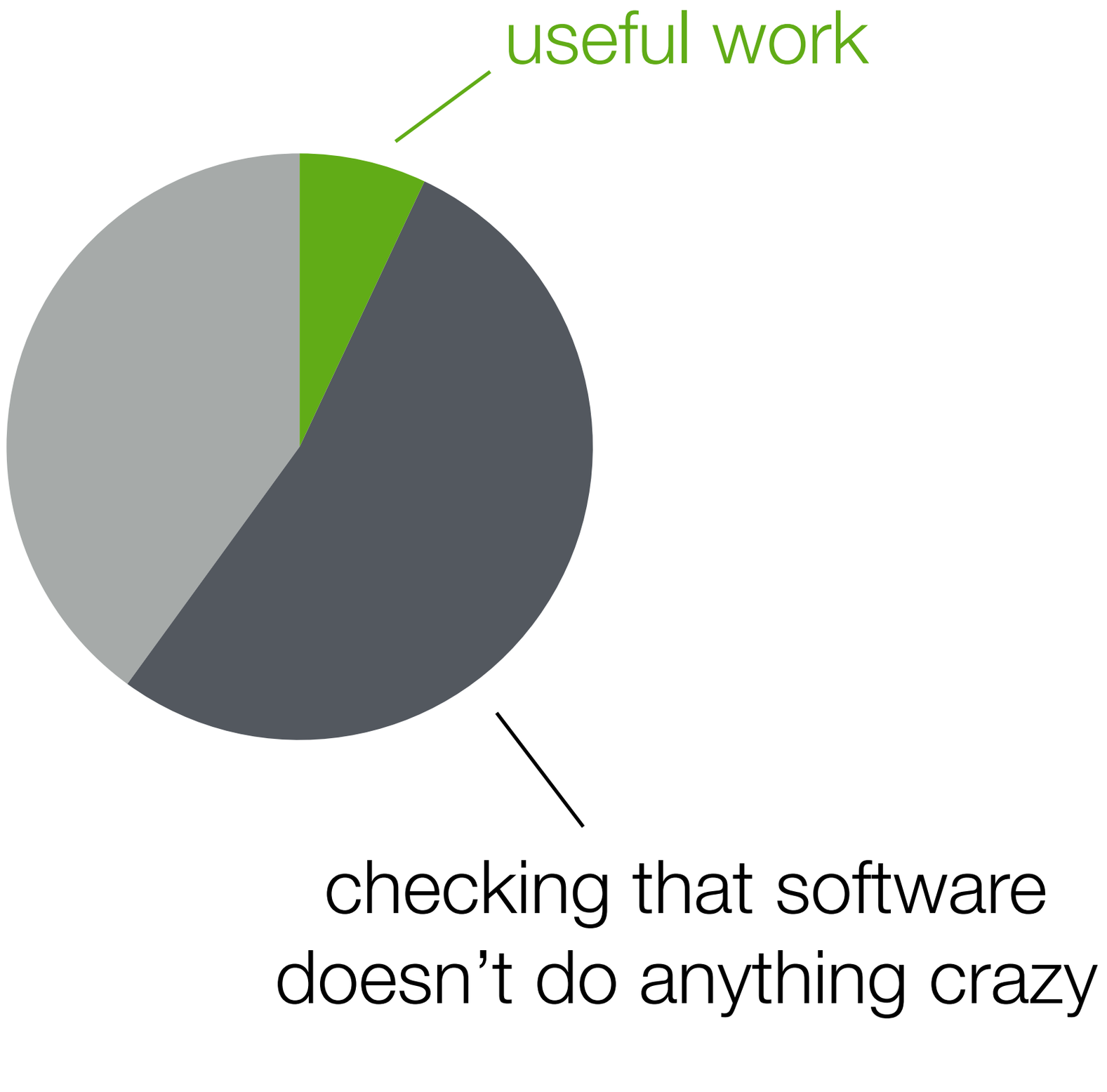
able to reduce management issues and to provide a consistent platform that applications can rely on. Second, datacenter services evolve extremely rapidly, making non-programmable hardware features impractical. Thus, datacenter providers are faced with a conundrum: they need continued improvements in performance and efficiency, but cannot obtain those improvements from general-purpose systems.

Reconfigurable chips, such as Field Programmable Gate Arrays (FPGAs), offer the potential for flexible acceleration of many workloads. However, as of this writing, FPGAs have not been widely deployed as compute accelerators in either datacenter infrastructure or in client devices. One challenge traditionally associated with FPGAs is the need to fit the accelerated function into the available reconfigurable area. One could virtualize the FPGA by reconfiguring it at run-time to support more functions than could fit into a single device. However, current reconfiguration times for standard FPGAs

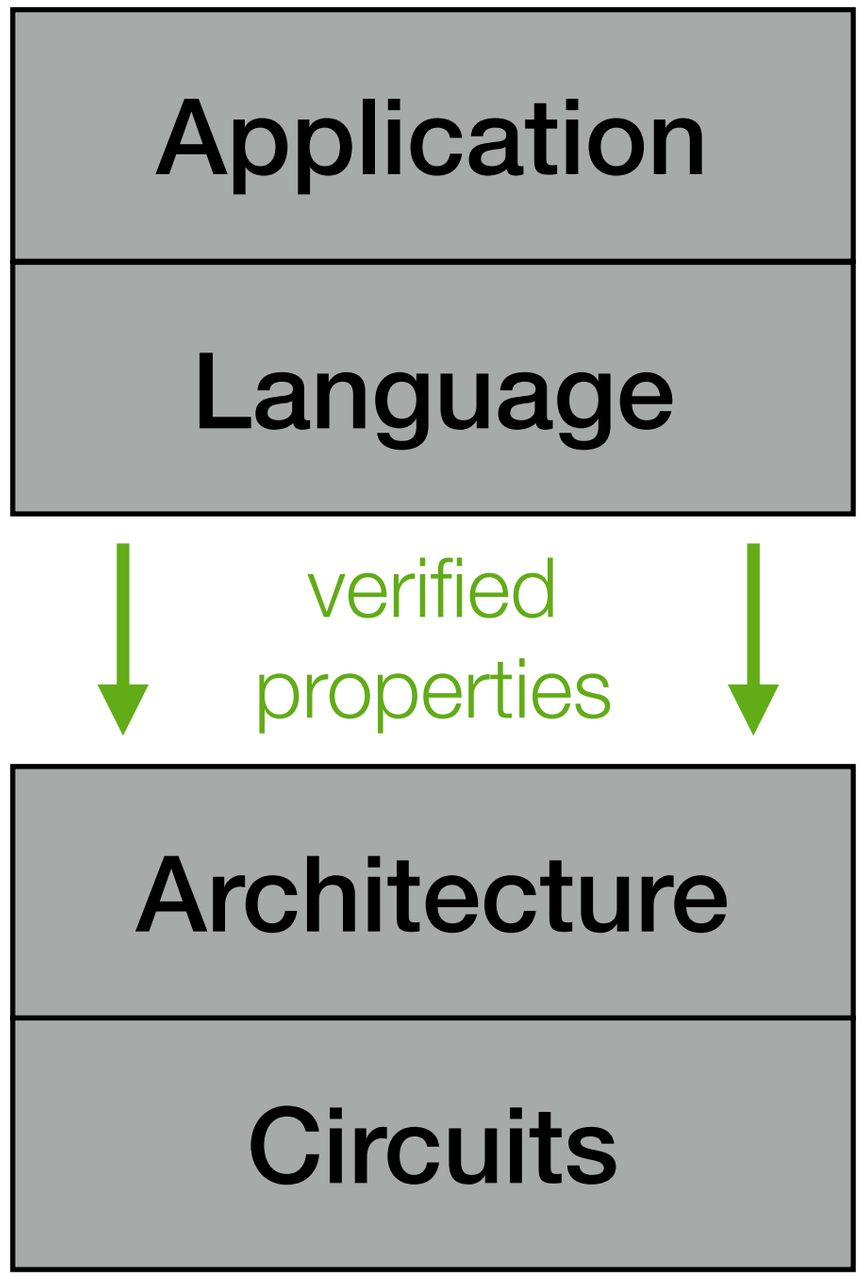
Trust, but formally verify



Trust, but formally verify



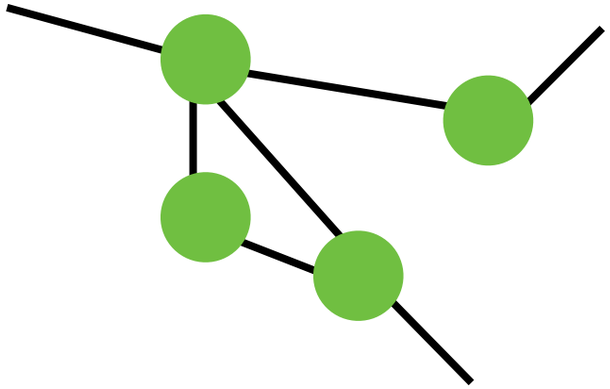
Trust, but formally verify



e.g., [Hunt and Larus; OSR April 2007]

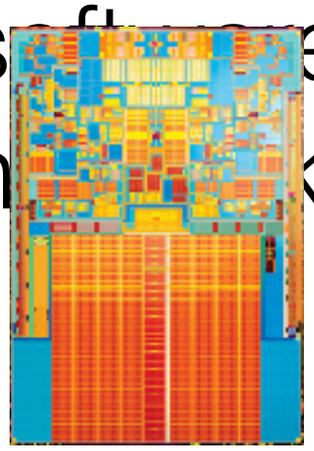
Hardware beyond core computation

power supply
& battery



software-defined networking

CPU



FPGA

GPU

accelerators

new memory
technologies

mobile display
& backlight

free lunch

multicore era

the era
of language
co-design?

time
immemorial

2005

2015