arm Research Summit

Rethinking Boundaries through Hardware-Software Co-design for Productive Post-Moore Computing

Hosted by: Jonathan Beard, Jose Joao

17 September 2019

The panel

Panelists

- Tom Conte (Georgia Institute of Technology)
- Samira Khan (University of Virginia)
- Jon Masters (RedHat)
- Yale Patt (The University of Texas at Austin)

Rules

- Each panelist has 180 seconds to present a position.
- **Audience questions** - every response should be limited to 60 seconds.
- Each panelist has 60 seconds to present a summary.

The starter questions

Do we need to consider optimizations across the full hardware and software stack rather than focusing on isolated aspects of technology?

Is there a role for heterogeneity and specialization to complement or replace traditional homogeneous general-purpose systems?

Will revolutionary device and memory technologies offering extra performance be ready fast enough to beat evolution of existing technologies?

Does software have to finally become more efficient, now that it cannot rely on the “free ride” of Moore's Law?

How will software developers need to consider the productivity and usability of solutions that expose complexity to assess whether they are economically viable?

The panel

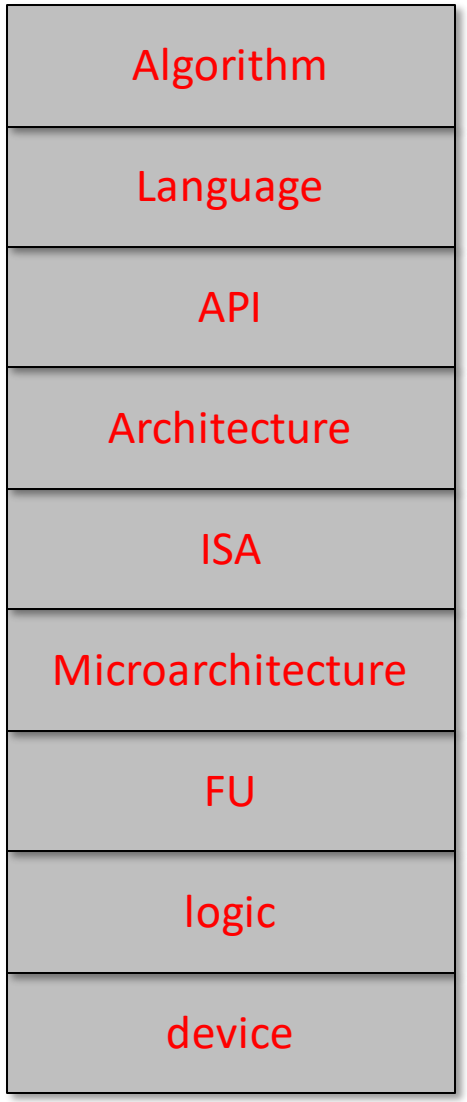
Panelists

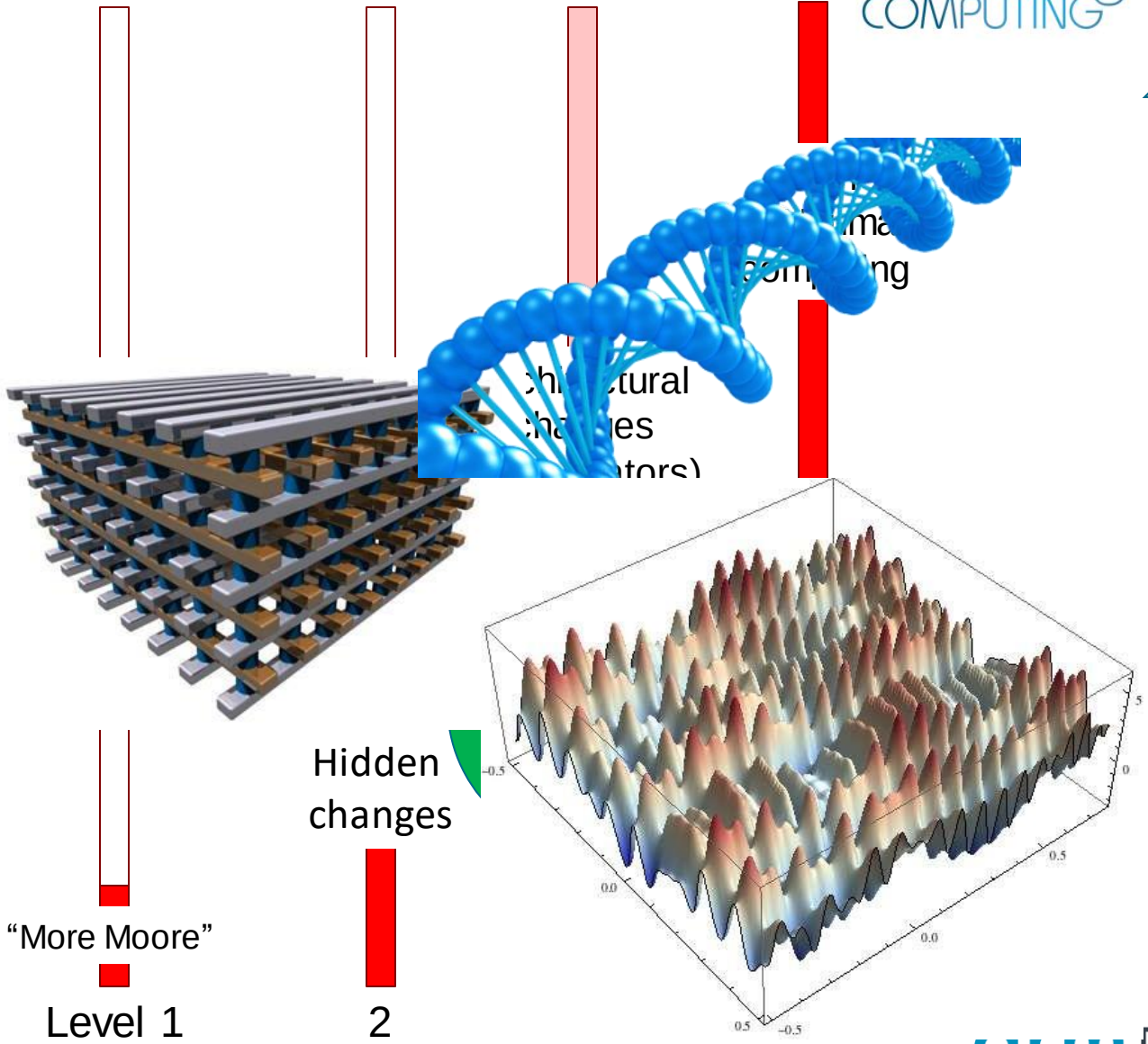
- Tom Conte (Georgia Institute of Technology)
- Samira Khan (University of Virginia)
- Jon Masters (RedHat)
- Yale Patt (The University of Texas at Austin)

Rules

- Each panelist has 180 seconds to present a position.
- **Audience questions** - every response should be limited to 60 seconds.
- Each panelist has 60 seconds to present a summary.

Potential Approaches vs. Disruption in Computing Stack





Rethinking Boundaries through Hardware-Software Co-design for Productive Post-Moore Computing

Samira Khan

My Background

My core is in architecture

2010-2015

Main Memory System
DRAM, NVM
Persistent Memory
Processing-in-Memory

2015-now

PMTest

XFTest

Debugging and testing tools for persistent memory applications

Software directed pre-execution for persistent memory

Janus

Profiling tool for PIM applications

PIMProfiler

arm Research

My Background

My core is in architecture

2010-2015

Main Memory System
DRAM, NVM
Persistent Memory
Processing-in-Memory

2015-now

PMTest

XFTest

Debugging and testing tools for persistent memory applications

Janus

Software directed pre-execution for persistent memory

PIMProfiler

Profiling tool for PIM applications

I design better software because I know how hardware works

Looking Backward: Lessons From Past

Hardware Transactional Memory

Has captivated architects for more than 20 years

Intel iAPX 432

My favorite coolest chip design

- **MYTH:** Cool hardware design is enough, everything will follow
- **REALITY:** Has to be ultra fast to justify new programming models (5X-10X)

- **MYTH:** Programmers want transparent hardware support
- **REALITY:** What programmers really want
  - Freedom and flexibility
  - Deterministic operations
  - Debugging support

Systems in The Future

Architects' Bubble

CPU

MEMORY

STORAGE

Ld/St

DRAM

FILE I/O

HDD

SPECIALIZED CORES

DRAM Layers

Logic Layer

Substrate

MEMORY WITH LOGIC

PERSISTENT MEM

SSD

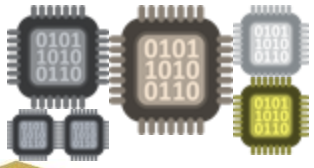
SMART SSD

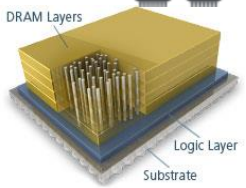
SMART NIC

BILLIONS OF EDGE DEVICES

Systems in The Future

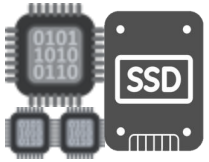
SPECIALIZED CORES



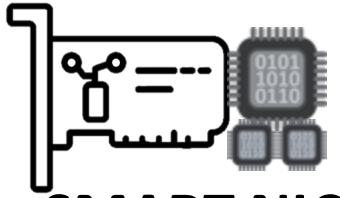


PERSISTENT MEM

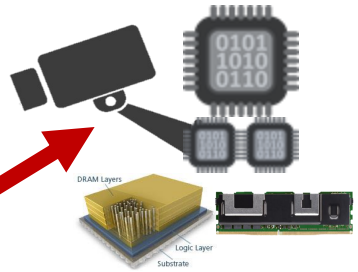
MEMORY WITH LOGIC

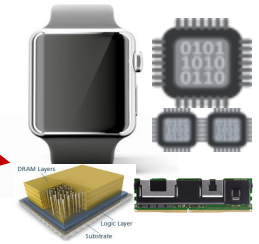


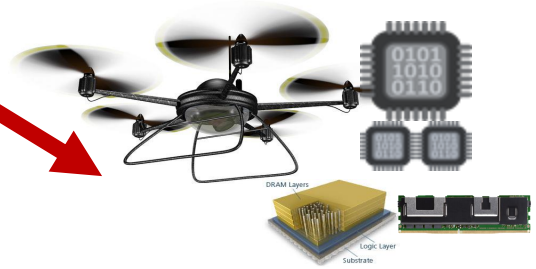
SMART SSD



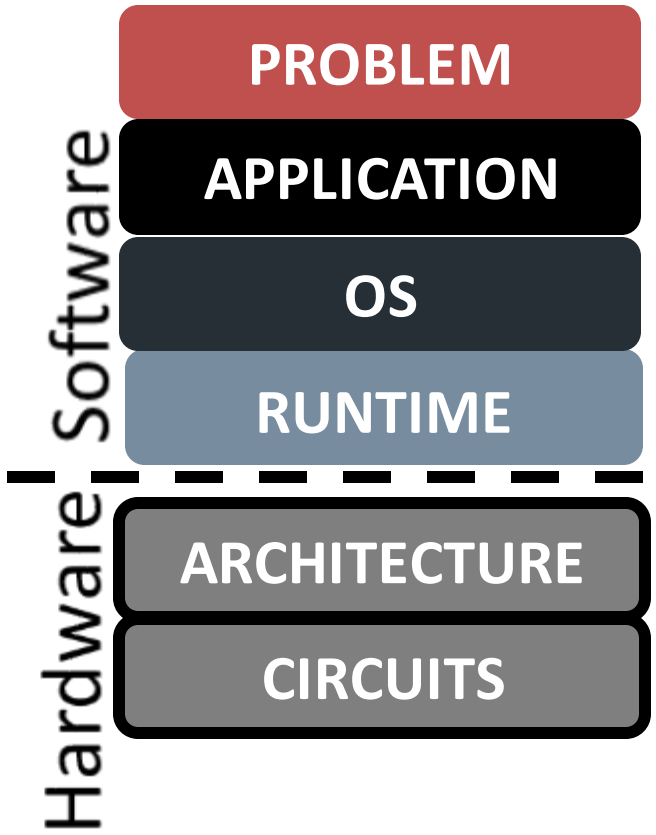
SMART NIC





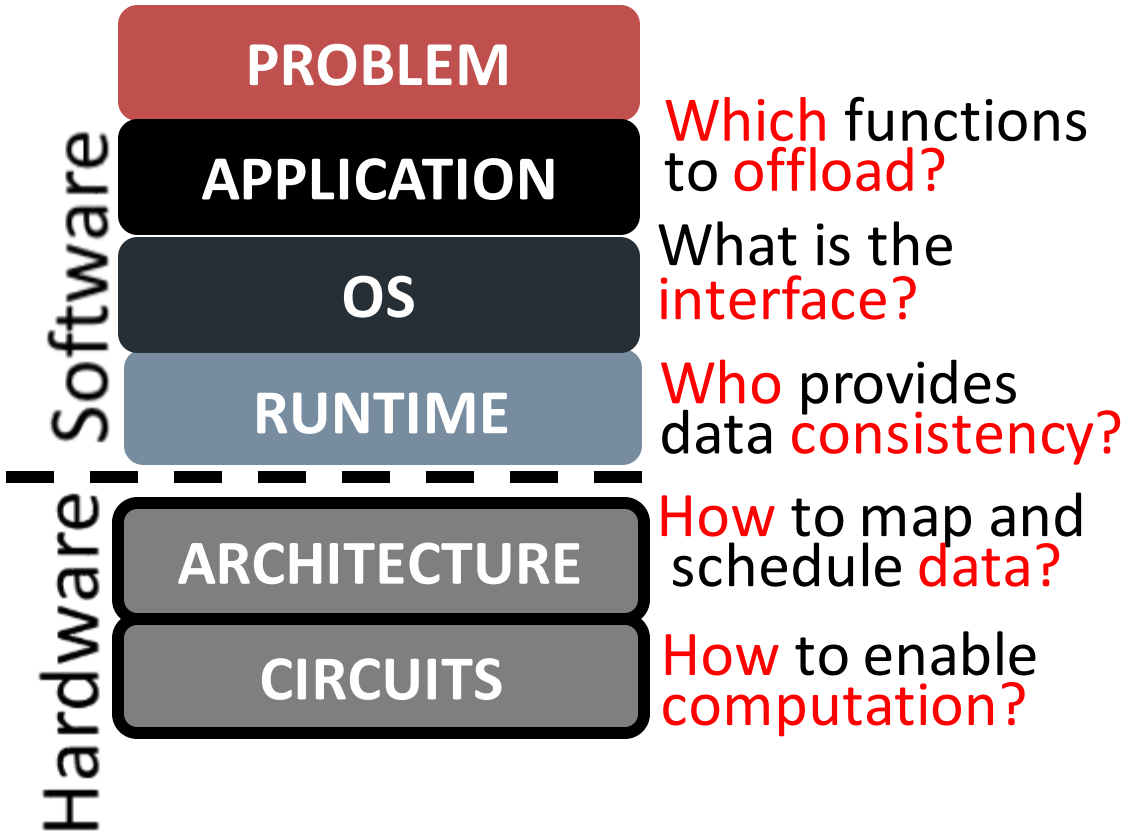


BILLIONS OF EDGE DEVICES



Need to build the proper software stack

Systems in The Future



- **MYTH:** Cool hardware design is enough, everything will follow
- **REALITY:** Has to be ultra fast to justify a new programming model (5X-10X)
- **MYTH:** Programmers want transparent hardware support
- **REALITY:** What programmers really want
  - Freedom and flexibility
  - Deterministic operations
  - Debugging support

Absolutely no way to make it programmable and easy to use without software-hardware collaborative design

Rethinking Boundaries through Hardware-Software Co-design for Productive Post-Moore Computing

Samira Khan

redhat.

Rethinking Boundaries through Hardware-Software Co-design

Jon Masters, Computer Architect, Red Hat, Inc.
jcm@redhat.com | @jonmasters

Arm Research Summit 2019

Status quo?

- **“Hardware” and “software” people don’t talk**
  - We created an “us” vs “them” collective mentality
  - We enjoy explicitly ignoring one another with intent
  - Software folks find hardware boring (and ignore it)
  - Hardware folks don’t have ways to engage ahead
  - Exceptions exist, but they are **NOT** the norm

- **“software” people: make friends with “hardware” people**
- **“hardware” people: make friends with “software” people**

This isn’t working

- Traditional “Enterprise” computing approach is many year hardware cycles
  - Vendor (maybe) takes some requirements input, hardware team builds design
  - Rely on abstractions (ISA, others) and build to the lowest common denominator
  - Reference platforms presented right at the end, software “enabled”, ship
  - Standardization is key to the existing approach (and to the next one)
- We need to evolve into a new co-design phase
  - Collaborate from the beginning to understand workloads
  - Remove unnecessary abstractions, iterate design together
  - Standardization is still important (e.g. CCIX/CXL), but where appropriate
  - Consider cloud-first and hybrid-cloud vs Enterprise

redhat.

Backup

Technologies that enable us

- The answer isn’t just “accelerators” but they are very important
- Standards enable us to have “just works” plugin acceleration
  - Too many standards: CXL, CCIX, GenZ, NVLink, CAPI, …
- Coherent accelerators make software enablement much easier
- Need to enable existing shipping OS/container/VMs, etc.

Panel: Rethinking Boundaries through Hardware-Software Co-design for Productive Post-Moore Computing

Yale Patt

The University of Texas at Austin

ARM Research Summit

17 September 2019

arm Research

Problem

Algorithm

Program

ISA (Instruction Set Arch)

Microarchitecture

Circuits

Electrons

Comments (Yale Patt)

- *The end of Moore's Law will raise the difficulty of designing future computer systems with steadily improving performance. Will our technologists be able to rise to the challenge?*

  I hope so. Never bet against engineering ingenuity.

- *Do we consider optimizations across the full hardware and software stack or focus on isolated aspects of technology?*

  The whole stack. Why would we limit ourselves

- *Is there a role for heterogeneity to complement or replace traditional homogeneous general-purpose systems?*

  Duh. I thought we finally agreed on that one years ago.

- *Will revolutionary technologies offering extra performance be ready fast enough to beat evolution of existing technologies?*

  We'll see. Existing technologies will evolve?

- *Does software have to finally become more efficient, now that it can not rely on the "free ride" of Moore's Law?*

  Huh? How about what microarchitecture has provided !!!!!

- *How will software developers consider the productivity, usability, and economics of solutions that expose complexity?*

  Education! Perhaps our most important challenge