arm Research

Arm Research Summit 2019

Abstractions and Specialization

Renegotiating Accelerator Abstractions Workshop

Matt Horsnell
Arm Research

Future Directions in Computer Architecture

University of Texas, Austin, November 11th 1977.

Recent developments in technology are making special-purpose solutions to information processing problems more attractive.

We ought to be able to off-load many special functions.

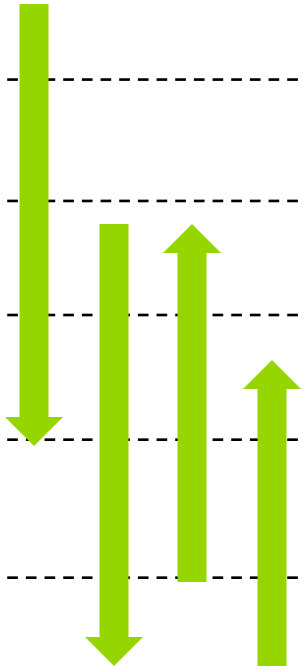
We should ask what architecture is best for a specific problem.

However, it deserves mention as a topic of future research. The goal of future research in high-level language and intermediate-level language architecture should be the design, implementation, and demonstration of practical high-level language machines and intermediate-level language machines.

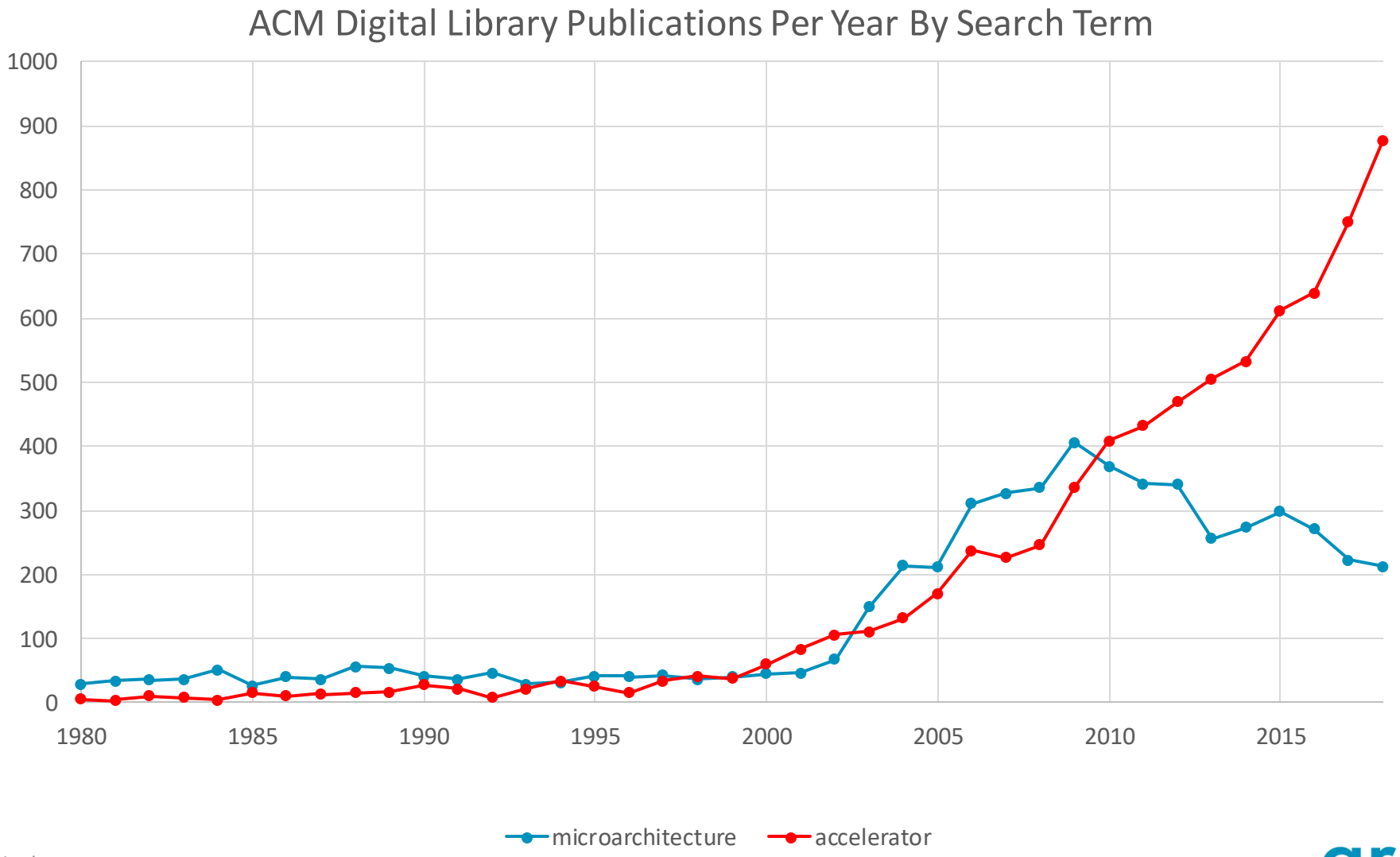
What's different now?

- Exponential growth in the scale of data
- Computational intensity needed to extract meaning (ML, Analytics)
- Impending end of Moore's Law
- Pressure on traditional layers of abstraction

Renegotiating Abstractions

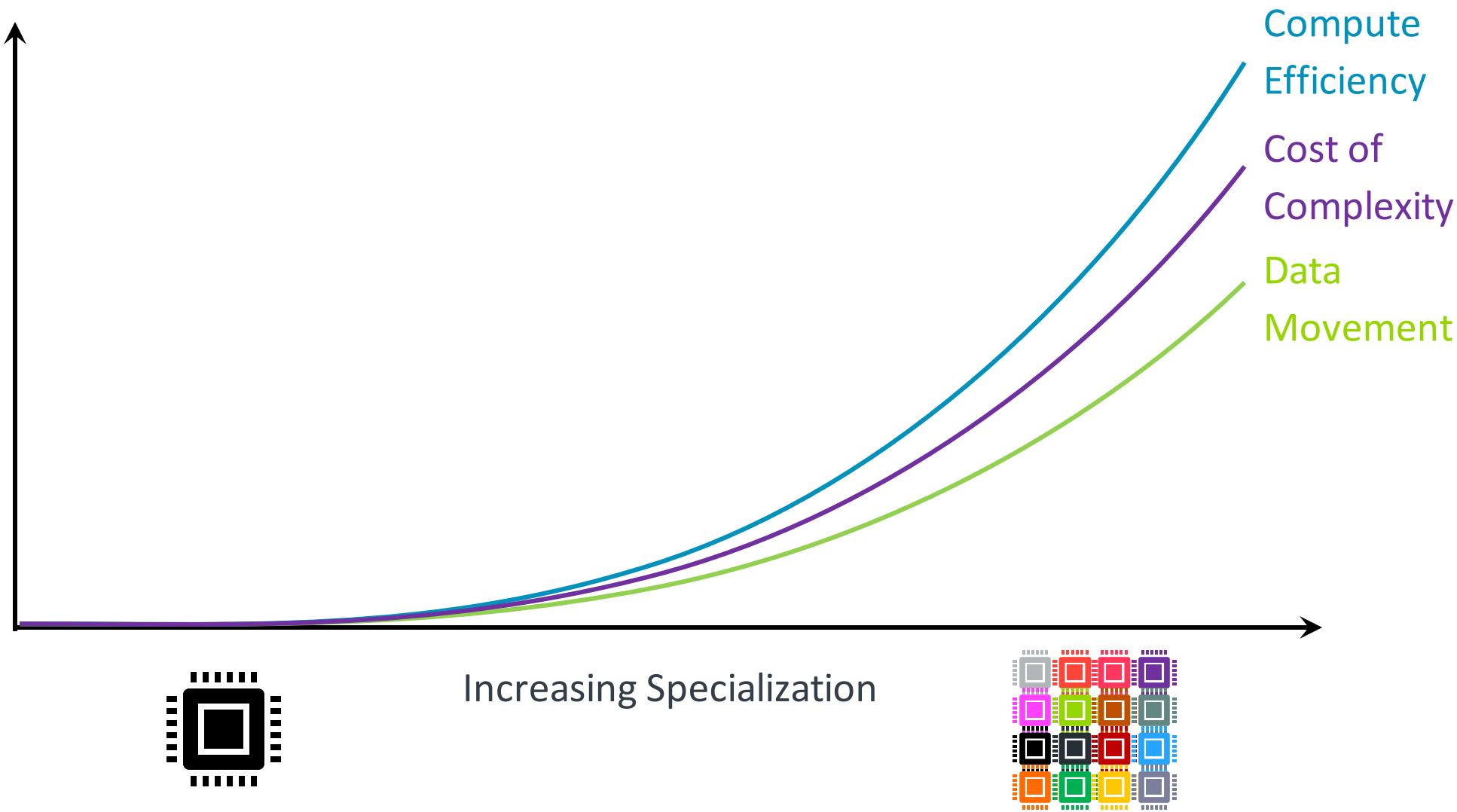
	Abstractions	Now	Future
Applications & Algorithms	Uniform Model		
Languages	Imperative, Shared Memory		
Compilers & IRs	Optimizations, Representations		
Execution Models & Runtimes	Orchestration, Management		
Architectures	Generalization, Specialization, System Composition		
Micro-architectures	Software Invisible		

Specialization – the “gold-rush”



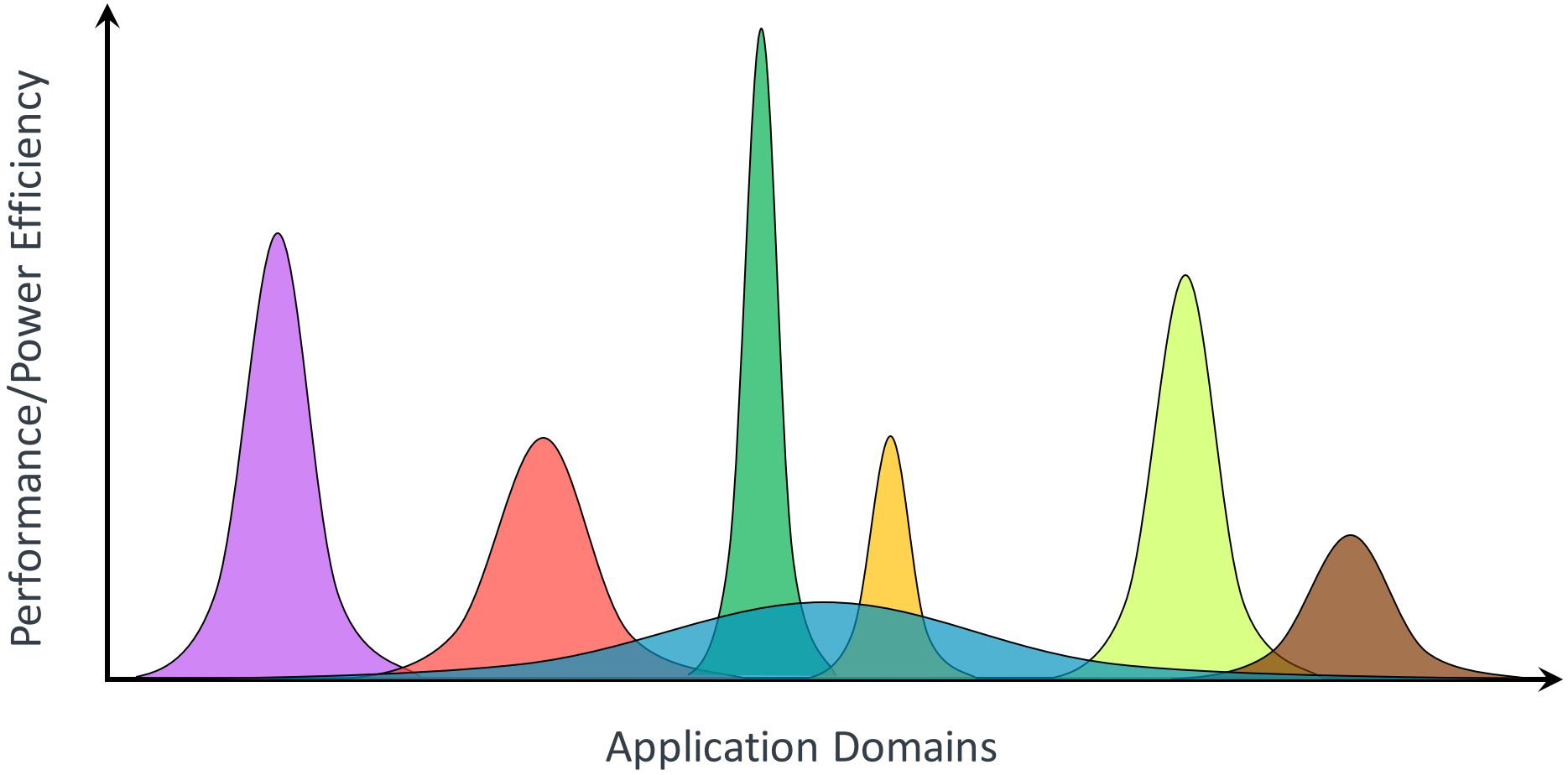
arm Research

Specialization – where is the sweet-spot?



arm Research

Specialization – coverage and convergence.



Adapted from “MultiAmdahl: Optimal Resource Allocation in Heterogeneous Architectures”, Yavits et. al

Specialization – where?

- In-core – specialized ISA, datatypes, data-paths and state.
- Tightly coupled – specialized “co-processor” interfaces and memory paths.
- Perhaps shared by multiple cores.
- Loosely coupled offload accelerator.
- Asymmetric cores – some having specialization some without.

- All valid points in the design space.
- Each has a different offload cost and changes the complexity of the hardware and software.

Some parting thoughts

- Accelerating an application is the tip of the iceberg.
- What do we do with messy applications – flows between domains?
- Do we think we can converge on a subset of general-specialized accelerators?
- Focused on the hardware and architecture aspects, but similar trade-offs and decisions apply in DSLs, compilers, frameworks, IRs.

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה