Heterogeneous Post-CMOS Technologies Meet Software

Jeronimo Castrillon

Chair for Compiler Construction, TU Dresden
jeronimo.castrillon@tu-dresden.de

Post Moore Interconnects Workshop
ISC High Performance
Frankfurt, Germany, June 23 2018
Context: Center for advancing electronics Dresden

- Large German Excellent Cluster
- **Goal:** “to explore new technologies for electronic information processing which overcome the limits of CMOS technology”

- Multiple participating organizations

- Multiple disciplines: Electrical Engineering, Computer Science, Materials, Chemistry, Physics, Biology
Cfaed Research Program (from 2012)
Cfaed sample technologies

Examples: Transistors, memory, interconnect and unconventional computing

Protein-based computing

Spin-orbit Racetracks

Reconfigurable transistors

Plasmonic waveguides


© J. Castrillon. ISC - Post-Moore, 2018
The evolution of systems: Extreme heterogeneity

- Inflection points
  - End of frequency scaling: Multicores
  - End of Dennard scaling and power density: heterogeneous systems
  - Physical limits of CMOS: Extreme heterogeneity (new materials, new paradigms)

- Many ideas (including those above): quantum, neuromorphic, protein-based, DNA storage, spin-orbittronics, organic, …
Software and heterogeneous systems

- Heterogeneous systems in niches for more than 30 years
  - Baseband processing (DSPs, hardwired accelerators)
  - Network processing units
  - Also doing approximate computing for a long time

Challenges
- Make it usable for a broader community for larger systems (also at the borders of computer science)
- Extreme heterogeneity: Still too much to understand

Challenges for SW systems

- **Orthogonalization of concerns**
  - Reduce the amount of rework if some part of the system changes
  - Separate core algorithm from their memory access (via abstractions)

- **Interoperability/interaction**
  - Interface components to talk to each other w/o knowing architectural properties
  - HW interfaces to provide safe non-OS-dependent interaction
  - Accelerators as first class citizens in systems
Challenges for SW systems (2)

- **Isolation**
  - Extreme heterogeneity means even more unexpected behavior in systems
  - Simple isolation mechanisms (capability-based) with hardware support to avoid expensive OS intervention

- **Abstractions**
  - Hide complexity from application developer (e.g. domain-specific languages)
  - Models and monitoring for automatic SW/HW adaptation
  - Fast and decentralized resource allocators complemented by sporadic global reorganizations
Investigating principles for a programming stack

Programming abstractions
- High-level: Domain-Specific Languages (DSLs)
- Lower-level: Dataflow execution models

Execution abstractions
- Application runtimes for adaptivity
- Micro-kernel based Oses

Models of machines and computation

Requires models: SW people require closer communication with technologists!
SW for extreme heterogeneity

- Working on principles for a programming stack
  - Programming abstractions
    - High-level: Domain-Specific Languages (DSL)
    - Lower-level: Dataflow execution models
  - Execution abstractions
    - Application runtimes for adaptivity
    - Micro-kernel based Oses
  - Models of machines and computation

- Requires models: SW people require closer communication with technologists!
HW Interfaces and Microkernels

- Data-Transfer Unit (DTU)
  - Unified interface for interoperability of heterogeneous components
  - HW-level isolation: access to external resources controlled by DTU
  - Simplifies management of heterogeneous components

N. Asmussen, et al., “M3: A Hardware/Operating-System Co-Design to Tame Heterogeneous Manycores”, ASPLOS’16
HW Interfaces and Microkernels

- Data-Transfer Unit (DTU)
  - Unified interface for interoperability
  - HW-level isolation
  - Simplified management

- M³: OS on top of DTU
  - Isolation: Kernel lets DTU enforce access/communication restrictions
  - Kernel is only responsible to establish communication channels
  - Interaction: components can directly communicate w/o OS intervention

Exotic HW can access system resources (isolated and low overhead)

N. Asmussen, et al., “M3: A Hardware/Operating-System Co-Design to Tame Heterogeneous Manycores”, ASPLOS’16
Domain-specific languages

- Higher-level algorithmic abstractions
  - More information makes it easier to optimize and adapt to
  - Examples: Tensor objects and operators, particle-based simulation

```
source =
type matrix : [mp np] &
type tensorIN : [np np np me] &
type tensorOUT : [mp mp mp me] &
var input A : matrix &
var input u : tensorIN &
var input output v : tensorOUT &
var input alpha : [] &
var input beta : [] &
v = alpha * (A # A # A # u .
[[5 8] [3 7] [1 6]]) + beta * v
```

Fortran embedding + JIT compilation
A. Susungi, et al., "Towards Compositional and Generative Tensor Optimizations" GPCE 17
N. A. Rink, et al., "CFDlang: High-level code generation for high-order methods in fluid dynamics", RWDSL 2018

Dataflow

- **Dataflow**: Formal execution semantics for transformations (compile and runtime)
- Used in the past for highly heterogeneous systems
  - Effort to describe abstractly the behavior and the interfacing of accelerators

Evaluation vehicle: Simulation (emulation)

- Huge effort in system simulation
- Mixture of technologies: sampling, trace-based, ...
- Extending: NVMain, Gem5, DRAMSys,…
  - Many collaborators
  - Need abstractions here as well!

Summary

- Cfaed: Center for advancing electronics Dresden
- Alternative technologies: Reconfigurable transistors, plasmonic waveguides, ...
- Scientific platform to start addressing software challenges

- Principles: Orthogonalization, interoperability, isolation and abstraction
  - Examples of OS and language research
  - Many works ahead!
References

M. Raitza, et al., "Exploiting Transistor-Level Reconfiguration to Optimize Combinational Circuits", DATE 2017


Parkin, US patents 6834005, 6898132.


Parkin, Scientific American (2009).


J. Castrillon, Tei-Wei Kuo, Heike E. Riel, Matthias Lieber, "Wildly Heterogeneous Post-CMOS Technologies Meet Software (Dagstuhl Seminar 17061)", In Dagstuhl Reports vol. 7, no. 2, pp. 1–22, Dagstuhl, Germany, Aug 2017

N. Asmussen, et al., “M3: A Hardware/Operating-System Co-Design to Tame Heterogeneous Manycores”, ASPLOS’16

A. Susungi, et al., "Towards Compositional and Generative Tensor Optimizations" GPCE 17

N. A. Rink, et al., "CFDlang: High-level code generation for high-order methods in fluid dynamics", RWDSL 2018


