Group 3: Architectural Design for Enhancing Programmability

Dean Tullsen, Josep Torrellas, Luis Ceze, Mark Hill, Onur Mutlu, Sampath Kannan, Sarita Adve, Satish Narayanasamy

Problem Statement

- Historically, we have attained sustained performance increases without asking for significant software changes. Continued performance scaling requires software and hardware changes to exploit parallelism. It is now much harder to get programmability, performance and correctness.

"Man-on-the-Moon" Goals

- Programming for parallel architectures as easy as it is now for sequential architectures
- Maintaining Moore's Law for performance (double the speedup every 2 years)
- No concurrency bugs

Research Issues

- Programming model
- Correctness
- Introspection
- Scalable Memory and Communication Fabric
- Resource management

Programming Model

- Vision: Co-evolve programming models and architectures for programmability, to rapidly attain correctness and performance.
- Specific research topics:
  - Programming model that allows:
    - Potentially express communication (e.g., producer-consumer, pipelined)
    - Hide/abstract asymmetries
  - Support for new language features, high-level languages, and safe languages
  - Understanding the hardware support for common models
    - Data-parallelism
    - Task parallelism
    - Functional
  - Supporting incremental optimization of an initial correct program implementation that has poor performance
  - Role of speculation (visible or not visible)
  - Redefining abstraction of HW/SW interface
    - Communication, locality, scheduling, synchronization

Correctness

- Vision: Architecture and programming model that increases the chances of having a correct program
- Specific research topics:
  - Extensive framework of tools for testing, debugging, performance monitoring, and code restructuring
  - Runtime operation of such tools
  - Hardware primitives to augment/support correctness tools (e.g., associate metadata with data)
  - Use the extra cores to improve the correctness
  - Schemes to proactively skip/stop-at defects
  - Application of machine-learning techniques for correctness
  - Find techniques that support both software debugging and hardware bring-up
  - Support for determinism when desired

Introspection

- Vision: Machine that collects and abstracts data that percolates up to the right level for analysis and adaptation
- Specific research topics:
  - Multiple models of interaction with the programmer. Passive (user not involved) or active (the user specifies hints).
  - Interaction hardware and runtime software
  - Enhancing monitoring hardware of critical performance/power events
  - HW/SW infrastructure and algorithms to mine data and identify bottlenecks and inefficiencies
  - SW/HW that adapts based on the learned information for
    - Performance and scalability
    - Energy efficiency
    - Correctness
  - Ability for the programmer to convey information to the hardware
  - Effective support for visualization

Scalable Memory and Communication Fabric

- Vision: Scalable memory and communication fabric that provides performance, scalability, power efficiency, and flexibility
- Specific research topics:
  - Flexible memory hierarchy
  - Adaptable designs for
    - Cache coherence
    - Memory consistency
    - Communication
  - Analyze the different needs for different users
  - Design with pay only what you use; lean and mean
  - Power proportional design

Resource Management

- Vision: architecture that supports flexible resource management and allocation, including isolation of software and hardware components for programmability, correctness and performance.
- Specific research topics:
  - Design to attain composable performance/power in a highly multiprogramming environment
  - Sandboxing parallel programs
  - Communication isolation between threads in the same program and across programs
  - Rethinking virtual memory and protection in concurrent systems
  - Application to systems software
  - Design for Quality of service
  - Scalable, transparent resource management, including energy

Why is this Transformative?

- Society has come to depend on substantial, continuous increase in performance. This research will allow us to continue in this path by harnessing parallel processing.