

Working Group 3:
Custom-Based Architectures

Chair: Peter Kogge

Vice Chair: Thomas Sterling

WG3 – Architecture: Custom based Charter

- Charter
 - Identify opportunities and challenges for innovative HEC system architectures, including alternative execution models, support mechanisms, local element and system structures, and system engineering factors to accelerate rate of sustained performance gain (time to solution), performance to cost, programmability, and robustness. Establish a roadmap of advanced-concept alternative architectures likely to deliver dramatic improvements to user applications through the end of the decade. Specify those critical developments achievable through custom design necessary to realize their potential.
- Chair
 - Peter Kogge, Notre Dame
- Vice-Chair
 - Thomas Sterling, California Institute of Technology & Jet Propulsion Laboratory

WG3 – Architecture: Custom based Guidelines and Questions

- Present driver requirements and opportunities for innovative architectures demanding custom design
- Identify key research opportunities in advanced concepts for HEC architecture
- Determine research and development challenges to promising HEC architecture strategies. Project brief roadmap of potential developments and impact through the end of the decade.
- Specify impact and requirements of future architectures on system software and programming environments.
- Example topics:
 - System-on-a-chip (SOC), Processor-in-memory (PIM), streaming, vectors, multithreading, smart networks, execution models, efficiency factors, resource management, memory consistency, synchronization

Working Group Participants

- Duncan Buell, U. So. Carolina
- George Cotter, NSA
- William Dally, Stanford Un.
- James Davenport, BNL
- Jack Dennis, MIT
- Mootaz Elnozahy, IBM
- Bill Feiereisen, LANL
- Michael Henesey, SRC Computers
- David Fuller, JNIC
- David Kahaner, ATIP
- Peter Kogge, U. Notre Dame
- Norm Kreisman, DOE
- Grant Miller, NCO
- Jose Munoz, NNSA
- Steve Scott, Cray
- Vason Srini, UC Berkeley
- Thomas Sterling, Caltech/JPL
- Gus Uht, U. RI
- Keith Underwood, SNL
- John Wawrzynek, UC Berkeley

Charter (from Charge)

- **Identify opportunities & challenges** for innovative HEC system architectures, including
 - alternative execution models,
 - support mechanisms,
 - local element and system structures,
 - and system engineering factors**to accelerate**
 - rate of sustained performance gain (time to solution),
 - performance to cost,
 - programmability,
 - and robustness.
- **Establish roadmap** of advanced-concept alternative architectures likely to deliver dramatic improvements to user applications through the end of the decade.
- **Specify those critical developments** achievable through custom design necessary to realize their potential.

Original Guidelines and Questions

- ***Present driver requirements*** and opportunities for innovative architectures demanding custom design
- ***Identify key research opportunities*** in advanced concepts for HEC architecture
- ***Determine research and development challenges*** to promising HEC architecture strategies.
- ***Project brief roadmap*** of potential developments and impact through the end of the decade.
- ***Specify impact and requirements*** of future architectures on system software and programming environments.
- ***(new) What role*** should/do universities play in developments in this area

Outline

- What is Custom Architecture (CA)
- Endgame Objectives, Benefits, & Challenges
- Fundamental Opportunities Delivered by CA
- Road Map
- Summary Findings
- Difficult fundamental challenges
- Roles of Universities

What Is Custom Architecture?

- Major components designed explicitly and system balanced for support of scalable, highly parallel HEC systems
- Exploits performance opportunities afforded by device technologies through innovative structures
- Addresses sources of performance degradation (inefficiencies) through specialty hardware and software mechanisms
- Enable higher HEC programming productivity through enhanced execution models
- Should incorporate COTS components where useful without sacrifice of performance

Endgame Objectives

- Enable solution of
 - Problems we can't solve now
 - And larger versions of ones we can solve now
- Base economic model: provides 10 – 100X ops/Lifecycle \$ AT SCALE
 - Vs inefficiencies of COTS
- Significant reduction in real cost of programming
 - Focus on sustained performance, not peak

Strategic Benefits

- Promotes architecture diversity
- Performance: ops & bandwidth over COTS
 - Peak: 10X – 100X through FPU proliferation
 - Memory bandwidth 10X-100X through network and signaling technology
 - Focus on sustainable
- High Efficiency
 - Dynamic latency hiding
 - High system bandwidth and low latency
 - Low overhead
- Enhanced Programmability
 - Reduced barriers to performance tuning
 - Enables use of programming models that simplify programming and eliminate sources of errors
- Scalability
 - Exploits parallelism at all levels of parallelism
- Cost, size, and power
 - High compute density

Challenges To Custom

- Small market and limited opportunity to exploit economy of scale
- Development lead time
- Incompatibility with standard ISAs
- Difficulty of porting legacy codes
- Training of users in new execution models
- Unproven in the field
- Need to develop new software infrastructure
- Less frequent technology refresh
- Lack of vendor interest in leading edge small volumes

Fundamental Technical Opportunities Enabled by CA

- Enhanced Locality – Increasing Computation/Communication Demand
- Exceptional global bandwidth
- Architectures that enable utilization of global bandwidth
- Execution models that enable compiler/programmer to use the above

Enhanced Locality – Increasing Computation/Communication Demand

Mechanisms

- Spatial computation via reconfigurable logic
- Streams that capture physical locality by observing temporal locality
- Vectors – scalability and locality microarchitecture enhancements
- PIM – capture spatial locality via high bandwidth local memory (low latency)
- Deep and explicit register & memory hierarchies
 - With software management of hierarchies

Technologies

- Chip stacking to increase local B/W

Providing Exceptional Global Bandwidth

Mechanisms:

- High radix networks
- Non-blocking, bufferless topologies
- Hardware congestion control
- Compiler scheduled routing

Technologies:

- High speed signaling (system-oriented)
 - Optical, electrical, heterogeneous (e.g. VCSEL)
- Optical switching & routing
- High bandwidth memory device, high density

Notes:

- Routing & flow control are nearing optimal

Architectures that Enable Use of Global Bandwidth

Note: This addresses providing the traffic stream to utilize the enhanced network

- Stream and Vectors
- Multi-threading (SMT)
- Global shared memory (a communication overhead reducer)
- Low overhead message passing
- Augmenting microprocessors to enhance additional requests (T3E, Impulse)
- Prefetch mechanisms

Execution Models

Note: A good model should:

- Expose parallelism to compiler & system s/w
- Provide explicit performance cost model for key operations
- Not constrain ability to achieve high performance
- Ease of programming
- Spatial direct mapped hardware
- Resource flow
- Streams
- Flat vs Dist. Memory (UMA/NUMA vs M.P.)
- New memory semantics
- CAF and UPC, first good step
- Low overhead synchronization mechanisms
- PIM-enabled: Traveling threads, message-driven, active pages, ...

Roadmap: When to Expect CA Deployment

- 5 Years or less
 - Must have relatively mature support s/w (and/or “friendly users”)
- 5-10 years
 - Still open research issues in tools & system s/w
 - Approaching 10 years if requires mind set change in applications programmers
- 10-15 years:
 - After 2015 all that’s left in silicon is architecture

Roadmap - 5 Year Period

- Significant research prototype examples
 - Berkeley Emulation Engine: \$0.4M/TF by 2004 on Immersed Boundary method codes
 - QCDOC: \$1M/TF by 2004
 - Merrimac Streaming: \$40K/TF by 2006
 - Note: several companies are developing custom architecture roadmaps

Roadmap - 5 Years or Less

Technologies Ready for Insertion

- High bandwidth network technology can be inserted
 - No software changes
- SMT: will be ubiquitous within 5 years
 - But will vendors emphasize single thread performance in lieu of supporting increased parallelism
- Spatial direct mapped approach

Roadmap - 5 to 10 Years

- All prior prototypes could be expanded to reach PF sustained at competitive recurring \$
- Industry is targeting sustained Petaflops
 - If properly funded
- Need to encourage transfer of research results
- Virtually all of prior technology opportunities will be deployable
 - Drastic changes to programming will limit adoption

Roadmap: 10-15 Years

- Silicon scaling at sunset
 - Circuit, packaging, architecture, and software opportunities remain
- Need to start looking now at architectures that mesh with end of silicon roadmap and non-silicon technologies
 - Continue exponential scaling of performance
 - Radically different timing/RAS considerations
 - Spin out: how to use faulty silicon

Findings

- Significant CA-driven opportunities for enhanced Performance/Programmability
 - 10-100X potential above COTS at the same time
- Multiple, CA-driven innovations identified for near & medium term
 - Near term: multiple proof of concept
 - Medium term: deployment @ petaflops scale
- Above potential will not materialize in current funding culture

Findings (2)

- No one side of the community can realize opportunities of future Custom Architecture:
 - Strong peer-peer partnering needed between industry, national labs, & academia
 - Restart pipeline of HEC & parallel-oriented grad students & faculty
- Creativity in system S/W & programming environments must support, track, & reflect creativity in HEC architecture

Findings (3)

- Need to start now preparing for end of Moore's Law and transition into new technologies
 - If done right, potential for significant trickle back to silicon

Fundamentally Difficult Challenges

Technical

- Newer applications for HEC
- OS geared specifically to highly scaled systems
- How to design HEC for upgradable
- High Latency, low bandwidth ratios of memory chips and systems
- File systems
- Reliability with unreliable components at large scale
- Fundamentally parallel ISAs

Fundamentally Difficult Challenges

Cultural

- Instilling change into programming model
- Software inertia
- How should HEC be viewed
 - As a service vs product
- I/O, SAN, Storage systems for HEC
- How to define requirements

Universities As A Critical Resource

- Provide innovative concepts and long term vision
- Provide students
- Keeps the research pipeline full
- Good at early simulations and prototype tools
- Students no longer commonly exposed to massive parallelism
- Parallel computing architecture students in significant decline, as well as those interested in HEC
- Difficult to roll leading edge chips but only place for 1st generation prototypes of novel concepts
- Don't do well at attacking the hard problems of moving beyond 1st prototype, or productizing
- Soft money makes it hard to keep teams together