Computer Architecture Discussion
Random tidbits

John Shalf
Lawrence Berkeley National Laboratory

August 10, 2009
Charge for DARPA Software Study

• Targets
  – Embedded, departmental servers, exascale

• Challenges
  – Power
  – Concurrency
  – Resilience

• Requires re-engineering of software stack to match emerging constraints
  – Original design assumptions are no longer valid
  – New stack targets continuity from embedded to exascale
Focus Areas (Vivek’s Paper)

• Expressing parallelism and locality
  – Because energy cost of moving a bit and increasing inhomogeneity of machine architecture
  – Challenges: portable expression of parallelism, dynamic parallelism, and heterogeneity

• Managing Parallelism and Locality
  – Need bigger role for runtime
  – Runtime needs OS to get out of the way (but maintain safety)
  – Compiler needs to support more “late binding” to work more productively with runtime to delay key decisions

• Software/Hardware co-design
  – Algorithms and hardware need to evolve together based on iterative evaluation of their respective constraints

• Application Requirements
Times They are a Changing
(trends in computer architecture)

• Power is first order design constraint
• Clock Frequency Scaling Has Ended
  – Now we double cores every 18 months instead of doubling clock frequency
• Memory capacity per computational element will be decreasing
  – Also forces us towards strong scaling, even if you don’t want it
  – Requires constant memory footprint in face of exponential scaling
• Memory and communication bandwidth per peak FLOP decreasing
  – Old optimization target was to reduce flops (increase communication)
  – New optimization target is to reduce communication (increase FLOPs)
• Architectural diversity is increasing (architectural uncertainty)
  – Current languages are mis-matched with emerging machine models
  – Performance portability is more of a problem than ever
• Load imbalance is increasingly problematic with larger parallelism
• Reliability for largest-scale systems is likely going down
Power

Now a first-order design constraint
• Immediate need to add 8 MW to prepare for 2007 installs of new systems
• NLCF petascale system could require an additional 10 MW by 2008
• Need total of 40-50 MW for projected systems by 2011
• Numbers just for computers: add 75% for cooling
• Cooling will require 12,000 – 15,000 tons of chiller capacity

Cost estimates based on $0.05 kW/hr

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LBNL</td>
<td>43.70</td>
<td>50.23</td>
<td>53.43</td>
<td>57.51</td>
<td>58.20</td>
<td>56.40</td>
</tr>
<tr>
<td>ANL</td>
<td>44.92</td>
<td>53.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORNL</td>
<td>46.34</td>
<td>51.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNNL</td>
<td>49.82</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data taken from Energy Management System-4 (EMS4). EMS4 is the DOE corporate system for collecting energy information from the sites. EMS4 is a web-based system that collects energy consumption and cost information for all energy sources used at each DOE site. Information is entered into EMS4 by the site and reviewed at Headquarters for accuracy.
Power is an Industry Wide Problem
(2% of US power consumption and growing)

“Hiding in Plain Sight, Google Seeks More Power”,
by John Markoff, June 14, 2006

New Google Plant in The Dulles, Oregon,
from NYT, June 14, 2006

Relocate to Iceland?
HPC Power: It will only get worse

- Baltimore Sun Article on NSA system in Maryland (relocated to Utah)
  - Consuming 75MW and growing up to 15MW/year
  - Not enough power left for city of Baltimore!

- LBNL IJHPCA Study for ~1/5 Exaflop for Climate Science in 2008
  - Extrapolation of Blue Gene and AMD design trends
  - Estimate: 20 MW for BG and 179 MW for AMD

- DOE E3 Report
  - Extrapolation of existing design trends to exascale in 2016
  - Estimate: 130 MW

- DARPA Study
  - More detailed assessment of component technologies
  - Estimate: 20 MW just for memory alone, 60 MW aggregate extrapolated from current design trends

The current approach is not sustainable!
DARPA Exascale Study

• Commissioned by DARPA to explore the challenges for Exaflop computing

• Two model for future performance growth
  – Simplistic: ITRS roadmap; power for memory grows linear with #of chips; power for interconnect stays constant
  – Fully scaled: same as simplistic, but memory and router power grow with peak flops per chip
We won’t reach Exaflops with this approach

From Peter Kogge, DARPA Exascale Study
... and the power costs will still be staggering

From Peter Kogge,
DARPA Exascale Study
An Alternate “BG” Scenario With Similar Assumptions

Correlates well with BG/P #s

Full
Simplistic

Exa flops
Peta flops

1.E+10
1.E+09
1.E+08
1.E+07
1.E+06
1.E+05
1.E+04
1.E+03
1/1/00 1/1/04 1/1/08 1/1/12 1/1/16 1/1/20

GFIops

Top 10 Rmax
Rpeak Leading Edge
Exascale Goal
Evolutionary Light Fully Scaled
Evolutionary Light Simplistically Scaled
Where does the power go?

- **Serial-performance oriented design**
  - Ignore Little’s Law \( C = BW \times Lat \) latency stalls
  - Speculative execution (ILP, speculative exec)
  - Dynamic power dissipation

- **Wasted data movement**
  - Cache-coherency lets data move all over the place
  - PRAM abstract machine model

- **Off-chip I/O’s increasing**
  - Telegraph equation for wires
  - \( \text{energy} = \text{bitrate} \times \text{Length}^2 / \text{cross-section area} \)

- **Memory**: core storage array, inefficient logic, off-chip I/O BW

- **Other**: (cooling, power supplies, storage technology trends)
Note on Power Calculations

• Some of the power calculations are based on equal-weighting power contribution (not based on “goodput”)

• As such, 0.01 bytes/flop assumes power is equal between memory and FPU
  – not based on throughput
  – Ratio could be revised if a different balance would result in desired throughput

• *Notion of “balance” must take total system power & cost as a constraint (Amdahl ratios do not do this)*
CPU

Multiple ways of looking at it
Views of CPU Constraints

• Market Driven View
• Transistor-level view
• Complexity/Practicality View
• Reliability
• Market view
Intel HPC Market Overview

- High End Systems (>$1M)
- Most/all Top 500 systems
- Custom SW & ISV apps
- Technology risk takers & early adopters

**Volume Market**
- Mainly capacity; <=~150 nodes
- Mostly clusters; >50% & growing
- Higher % of ISV apps
- Fast growth from commercial HPC; Oil & Gas, Financial services, Pharma, Aerospace, etc.

Total market >$10.0B in 2006
Forecast >$15.5B in 2011

HPC is built with of pyramid investment model

IDC Segment

<table>
<thead>
<tr>
<th>System Size</th>
<th>2005</th>
<th>2010</th>
<th>CAGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$250K-$1M</td>
<td>$1.9B</td>
<td>$3.4B</td>
<td>11.8%</td>
</tr>
<tr>
<td>$50K-$250K</td>
<td>$2.9B</td>
<td>$4.9B</td>
<td>10.7%</td>
</tr>
<tr>
<td>0-$50K</td>
<td>$2.2B</td>
<td>$3.4B</td>
<td>9.6%</td>
</tr>
</tbody>
</table>

ASC/OASCR Collabs
Dec 11, 2008
Processor Technology Trend

- **1990s** - R&D computing hardware dominated by desktop/COTS
  - Had to learn how to use COTS technology for HPC
- **2010** - R&D investments moving rapidly to consumer electronics/ embedded processing
  - Must learn how to leverage embedded processor technology for future HPC systems
Consumer Electronics has Replaced PCs as the Dominant Market Force in CPU Design!!

Netbooks based on Intel Atom embedded processor is the fastest growing portion of “laptop” market.

Apple Introduces Cell Phone (iPhone)
Technology Continuity for A Sustainable Hardware Ecosystem

Scalable building blocks for a compelling environment at all scales
• Transistor view
Traditional Sources of Performance Improvement are Flat-Lining

- **New Constraints**
  - 15 years of exponential clock rate growth has ended
- **But Moore’s Law continues!**
  - How do we use all of those transistors to keep performance increasing at historical rates?
  - Industry Response: #cores per chip doubles every 18 months instead of clock frequency!
  - *Is this a good idea, or is it completely brain-dead?*
  - *Has industry run out of ideas?*

Figure courtesy of Kunle Olukotun, Lance Hammond, Herb Sutter, and Burton Smith
CMOS Scaling

**SCALING:**
- Voltage: \( V/\alpha \)
- Oxide: \( t_{ox}/\alpha \)
- Wire width: \( W/\alpha \)
- Gate width: \( L/\alpha \)
- Diffusion: \( x_d/\alpha \)
- Substrate: \( \alpha N_A \)

**RESULTS:**
- Higher Density: \( \sim \alpha^2 \)
- Higher Speed: \( \sim \alpha \)
- Power/ckt: \( \sim 1/\alpha^2 \)
- Power Density: \( \sim \text{Cons} \)

What is Happening Now?

- **Moore’s Law**
  - Silicon lithography will improve by 2x every 18 months
  - Double the number of transistors per chip every 18mo.

- **CMOS Power**
  
  Total Power = \( V^2 \cdot f \cdot C_{\text{active power}} + V \cdot I_{\text{leakage}} \)
  - As we reduce feature size Capacitance \( (C) \) decreases proportionally to transistor size
  - Enables increase of clock frequency \( (f) \) proportionally to Moore’s law lithography improvements, with same power use
  - This is called “Fixed Voltage Clock Frequency Scaling” *(Borkar `99)*

- **Since ~90nm**
  - \( V^2 \cdot f \cdot C \approx V \cdot I_{\text{leakage}} \)
  - Can no longer take advantage of frequency scaling because passive power \( (V \cdot I_{\text{leakage}}) \) dominates
  - Result is recent clock-frequency stall reflected in Patterson Graph at right

SPEC Int benchmark performance since 1978 from Patterson & Hennessy Vol 4.
What is Happening Now?

- **Moore’s Law**
  - Silicon lithography will improve by 2x every 18 months
  - Double the number of transistors per chip every 18mo.

- **CMOS Power**
  - Total Power = $V^2 \cdot f \cdot C + V \cdot I_{\text{leakage}}$
  - As we reduce feature size Capacitance ($C$) decreases proportionally to transistor size
  - Enables increase of clock frequency ($f$) proportionally to Moore’s law lithography improvements, with same power use
  - This is called “Fixed Voltage Clock Frequency Scaling” (Borkar ’99)

- **Since ~90nm**
  - $V^2 \cdot f \cdot C \sim V \cdot I_{\text{leakage}}$
  - Can no longer take advantage of frequency scaling because passive power ($V \cdot I_{\text{leakage}}$) dominates
  - Result is recent clock-frequency stall reflected in Patterson Graph at right

SPEC Int benchmark performance since 1978 from Patterson & Hennessy Vol 4.
Off-State Leakage Current ($I_{OFF}$)

- The leakage current specification sets the lower limit for the threshold voltage $V_{TH}$.

The inverse slope is defined to be the subthreshold swing, $S$.

- The leakage current specification sets the lower limit for the threshold voltage $V_{TH}$. 

![Graph showing log $I_{DS}$ vs. $V_{GS}$ with $V_{TH}$ and leakage current, $I_{OFF}$ highlighted.](image-url)
$V_{TH}$ Design Trade-Off

- Low $V_{TH}$ is desirable for high ON current:
  \[ I_{ON} \propto (V_{DD} - V_{TH})^\eta \quad 1 < \eta < 2 \]
  where $V_{DD}$ is the power-supply voltage

...but high $V_{TH}$ is needed for low OFF current
Historical Voltage Scaling

- Since $V_{TH}$ cannot be scaled down aggressively, the power-supply voltage ($V_{DD}$) has not been scaled down in proportion to the MOSFET channel length:

![Graph showing $V_{DD}$ and $V_{TH}$ over technology generations.]

Source: P. Packan (Intel), 2007 IEDM Short Course
Power Density Scaling – NOT!

Power Density vs. CMOS Scaling

Source: B. Meyerson (IBM) Semico Conf., January 2004

Power Density Prediction circa 2000

- Sun’s Surface
- Rocket Nozzle
- Nuclear Reactor
- Hot Plate
- Pentium® proc

Source: S. Borkar (Intel)

Photo of a Virtex IV in Standby Mode

courtesy L. Pileggi
Minimizing Operation Energy

CMOS Energy per Operation

\[ E_{\text{total}} = E_{\text{dynamic}} + E_{\text{leakage}} = \alpha L_d C V_{\text{dd}}^2 + L_d I_{\text{OFF}} V_{dd} t_{\text{delay}} \]

\[ t_{\text{delay}} = \frac{L_d C V_{\text{dd}}}{2 I_{\text{ON}}} \]

→ CMOS has a fundamental lower limit in energy per operation, due to subthreshold leakage.
Computing performance is now limited by power dissipation. This has forced the move to parallelism as principal means of increasing performance without increasing energy per operation.
The Need for a Better Switch

- When each core operates at the minimum energy, increasing performance requires more power.
Nano-Electronics Science and Technology Thrust

- Advanced materials, structures, and transistor designs to enable dramatic improvements in energy efficiency and performance via high degrees of parallelism and 3-D integration.

**Multi-gate MOSFETS:**
- advanced channel materials
- ferro-electric gate dielectrics

**Vertical Nanowire Structures:**

**Tunnel MOSFET (TFET) Design:**

**Measured Tri-Gate I-V**

**Measured TFET I-V**

**Energy-Performance Comparison**
(30-stage fanout-4 inverter chains)
• Practical Considerations simpler CPUs
Non-Recurring Engineering Costs are Rising

- Productivity is not keeping up with Moore’s law
- New ASICs are $20M and more!
- Only few markets can justify custom ASICs

Die complexity increases 58% per year

Productivity increases 21% per year

Source: Flynn, ASAP’05
The Scaling of ASIC Designs

Source: SIA Roadmap, 2001

Ofer Sacham
Stanford
Hardware: What are the problems?
(Lessons from the Berkeley View)

• Current Hardware/Lithography Constraints
  – Power limits leading edge chip designs
    • Intel Tejas Pentium 4 cancelled due to power issues
  – Yield on leading edge processes dropping dramatically
    • IBM quotes yields of 10 – 20% on 8-processor Cell
  – Design/validation leading edge chip is becoming unmanageable
    • Verification teams > design teams on leading edge processors

• Solution: Small Is Beautiful
  – Simpler (5- to 9-stage pipelined) CPU cores
    • Small cores not much slower than large cores
  – Parallel is energy efficient path to performance: \( CV^2F \)
    • Lower threshold and supply voltages lowers energy per op
  – Redundant processors can improve chip yield
    • Cisco Metro 188 CPUs + 4 spares; Sun Niagara sells 6 or 8 CPUs
  – Small, regular processing elements easier to verify
Low-Power Design Principles

- Cubic power improvement with lower clock rate due to $V^2F$
- Slower clock rates enable use of simpler cores
- Simpler cores use less area (lower leakage) and reduce cost
- Tailor design to application to REDUCE WASTE

Move towards throughput oriented design
Low-Power Design Principles

- **Power5 (server)**
  - 120W@1900MHz
  - Baseline

- **Intel Core2 sc (laptop)**:
  - 15W@1000MHz
  - 4x more FLOPs/watt than baseline

- **Intel Atom (handhelds)**
  - 0.625W@800MHz
  - 80x more

- **Tensilica XTensa DP (Moto Razor)**:
  - 0.09W@600MHz
  - 400x more (80x-120x sustained)
Low Power Design Principles

- **Power5 (server)**
  - 120W@1900MHz
  - Baseline

- **Intel Core2 sc (laptop)**:
  - 15W@1000MHz
  - 4x more FLOPs/watt than baseline

- **Intel Atom (handhelds)**
  - 0.625W@800MHz
  - 80x more

- **Tensilica XTensa DP (Moto Razor)**:
  - 0.09W@600MHz
  - 400x more (80x-100x sustained)

Even if each simple core is 1/4th as computationally efficient as complex core, you can fit hundreds of them on a single chip and still be 100x more power efficient.
• Fault tolerance
Fault Tolerance/Resilience

• Fault characteristics are defined by the market
  – Faults proportional to # sockets (not # cores) and silicon surface area
  – Energy efficient approach (with many cores) expose less surface area and fewer sockets with our approach
  – Industry demands fixed FIT rate per node (but HPC wants 10x every 3 years, which requires more nodes)

• Hard Errors
  – Spare cores in design (Cisco Metro)
  – SoC design (fewer components and fewer sockets)
  – Use solder (not sockets)

• Soft Errors
  – ECC for memory and caches
  – On-board NVRAM controller for localized checkpoint
    Checkpoint to neighbor for rollback
Reliability

Figure 2. Failures in billions of hours of operation\textsuperscript{2-6}

- 100% Vulnerable, 100% AVF
- 20% Vulnerable, 100% AVF
- 100% Vulnerable, 10% AVF
- 20% Vulnerable, 10% AVF
- IBM Goal
Specialization

Accumulation of ISA Cruft
Peel Back the Historical Growth of Instruction Sets \textit{(accretion of cruft)}

Chris Rowen: Tensilica Inc.

Traditional Processor Family

- System Interface
- Input/Output Wires
- Data Streaming Ports
- Slave DMA Access
- MP Split Transaction
- Block Data Bus
- Bus Bridges
- Special-Purpose DSP
- 16b GP DSP
- Superscalar
- 24b Audio
- Image multimedia
- Packet processing
- Encryption
- Computation Instruction Set
- RTL
- Processor Control
- Memory Systems
- Coherent Caches
- Tightly Coupled Memories
- Inst Cache
- Debug
- Interrupts
- Timers
- Data Cache
- Write-back Cache
- Inst Cache
- Memory Protection
- MMU
- Secure Rings
- Time per variant: years

Configurable Processor Family

- System Interface
- Input/Output Wires
- Data Streaming Ports
- Slave DMA Access
- MP Split Transaction
- Block Data Bus
- Bus Bridges
- Special-Purpose DSP
- 16b GP DSP
- Superscalar
- 24b Audio
- Image multimedia
- Packet processing
- Encryption
- Computation Instruction Set
- Processor Control
- Memory Systems
- Coherent Caches
- Tightly Coupled Memories
- Inst Cache
- Debug
- Interrupts
- Timers
- Data Cache
- Write-back Cache
- Inst Cache
- Memory Protection
- MMU
- Secure Rings
- Time per variant: days

Area = silicon cost and power
A Short List of x86 Opcodes that Science Applications Don’t Need!

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Size</th>
<th>Flags</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>16</td>
<td>32</td>
<td>Accumulator Add</td>
</tr>
<tr>
<td>ADC</td>
<td>16</td>
<td>32</td>
<td>Accumulator Add with Carry</td>
</tr>
<tr>
<td>SUB</td>
<td>16</td>
<td>32</td>
<td>Subtract</td>
</tr>
<tr>
<td>SBB</td>
<td>16</td>
<td>32</td>
<td>Subtract with Borrow</td>
</tr>
<tr>
<td>SVC</td>
<td>16</td>
<td>32</td>
<td>System Call</td>
</tr>
<tr>
<td>VMX</td>
<td>16</td>
<td>32</td>
<td>VM Exit</td>
</tr>
<tr>
<td>INVD</td>
<td>16</td>
<td>32</td>
<td>Invalidate Data Cache</td>
</tr>
</tbody>
</table>

Note: The table lists opcodes that are generally not used in scientific applications due to their uncommon use cases or potential for inefficiency.
More Wasted Opcodes

• We only need 80 out of the nearly 300 ASM instructions in the x86 instruction set!
• Still have all of the 8087 and 8088 instructions!
• Wide SIMD Doesn’t Make Sense with Small Cores
• Neither does Cache Coherence
• Neither does HW Divide or Sqrt for loops
  • Creates pipeline bubbles
• Better to unroll it across the loops (like IBM MASS libraries)
• Move TLB to memory interface because its still too huge
Our ability to sense, collect, generate and calculate on data is growing faster than our ability to access, manage and even “store” that data

- Memory density is doubling every three years; processor logic is every two
- Storage costs (dollars/Mbyte) are dropping gradually compared to logic costs

Source: David Turek, IBM
Memory Power Trends

• “Manycore” design point improves computational efficiency dramatically
  – Can fit hundreds to thousands of cores per chip
  – Increases computational throughput by 20-100x with fixed power envelope
  – Opportunity for 100x efficiency improvements for datacenters (2% of US Power consumption) and HPC (hundreds of megawatts per center)
  – But increasing memory bandwidth proportional to computational throughput causes memory power to dominate

• Need to bring memory interface power consumption to parity with optimized CPU core design point
  – Otherwise, further improvements to computational efficiency will not realize any benefit
Memory Power Evolution (is getting a lot worse)
Manycore Improves Efficiency of CPU But Pushes Power Problem to Memory

Minas & Ellison

Comparison of Server Power (40nm)
1Gbit DDR3 Architecture

• DRAM power is dominated by
  – Sense amp power (amplifying cell voltages)
  – DDR memory interface bus power
1Gbit DDR3 Architecture

- DRAM power is dominated by
  - Sense amp power (amplifying cell voltages)
  - DDR memory interface bus power

Dean Klein
Micron
Exascale: The Power of Memory

Assumptions

Cell Voltage  
1.2 V

Cell Capacitance  
25 fF

Bitline Capacitance  
75 fF

Memory System Bandwidth  
1 EB/sec

Simplified Results:

Energy/bit  
36 fJ

Total Memory Cell Power  
288 KW

With Bitline  
1150 KW

With 512X Over-Fetch  
590 MW

And this is before we consider technology to move the data from DRAM to CPU!
Nano-Mechanics Science and Technology

- Materials and processes to enable ultra-low-energy, high-density, high-speed embedded nano-electro-mechanical non-volatile memory.

**NEMory array**

**Measured NEMory cell characteristics**

- $L_{\text{beam}} = 886 \text{ nm}$, $V_{BL} = 0 \text{ V}$
- $W_{\text{beam}} = 53 \text{ nm}$, $V_{RWL} = 0 \text{ V}$
- $t_{\text{beam}} = 100 \text{ nm}$
- $t_{\text{gap,1,2}} = 30 \text{ nm}$
- $V_{\text{BL-WWL}}$ vs $\Delta C_{BL-WW}$

<table>
<thead>
<tr>
<th></th>
<th>NEMory</th>
<th>NOR Flash</th>
<th>Phase-Change Memory</th>
<th>Ionic Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage mechanism</td>
<td>mechanical gap-closing actuator</td>
<td>charge on floating gate</td>
<td>reversible material phase change</td>
<td>ion transport and redox reaction</td>
</tr>
<tr>
<td>Cell area</td>
<td>$6 \sim 12 \text{ F}^2$</td>
<td>$10 \text{ F}^2$</td>
<td>$4.8 \text{ F}^2$</td>
<td>$5 \sim 10 \text{ F}^2$</td>
</tr>
<tr>
<td>Program/erase time</td>
<td>$0.9 \text{ ns} / 0.3 \text{ ns}$</td>
<td>$1 \mu\text{s} / 10 \text{ ms}$</td>
<td>$50 \text{ ns} / 120 \text{ ns}$</td>
<td>$&lt; 20 \text{ ns}$</td>
</tr>
<tr>
<td>Read time</td>
<td>$&gt;1.5 \text{ ns}$</td>
<td>$10 \text{ ns}$</td>
<td>$60 \text{ ns}$</td>
<td>$&lt;10 \text{ ns}$</td>
</tr>
<tr>
<td>Program/erase voltage</td>
<td>$1.5 \text{ V}$</td>
<td>$12 \text{ V}$</td>
<td>$3 \text{ V}$</td>
<td>$&lt;0.5 \text{ V}$</td>
</tr>
<tr>
<td>Read voltage</td>
<td>$3 \text{ V}$</td>
<td>$2 \text{ V}$</td>
<td>$3 \text{ V}$</td>
<td>$&lt;0.2 \text{ V}$</td>
</tr>
<tr>
<td>Program/erase energy</td>
<td>$3 \times 10^{-17} \text{ J/bit}$</td>
<td>$10^{-14} \text{ J/bit}$</td>
<td>$5 \times 10^{-12} \text{ J/bit}$</td>
<td>$10^{-15} \text{ J/bit}$</td>
</tr>
</tbody>
</table>
Interconnects

The cost of moving a bit from point A to point B
Interconnect Cost
(Scalable Topologies)

• Fully-connected networks scale superlinearly in cost, but perform the best
• Limited-connectivity networks scale linearly in cost, but introduce new problems
Interconnect Design Considerations for Message Passing Applications

- Application studies provide insight to requirements for Interconnects (both on-chip and off-chip)
  - On-chip interconnect is 2D planar (crossbar won’t scale!)
  - Sparse connectivity for most apps.; crossbar is overkill
  - No single best topology
  - Most point-to-point message exhibit sparse topology + often bandwidth bound
  - Collectives tiny and primarily latency bound
- Ultimately, need to be aware of the on-chip interconnect topology in addition to the off-chip topology
  - Adaptive topology interconnects (HFAST)
  - Intelligent task migration?
Energy Consumed by Moving Data

• Moving data consumes substantial energy

• Need more explicit control of data movement to economize memory movement
  – E.g. use explicitly software managed memory

• Need a better abstraction for dealing with software managed memory
The problem with Wires: Energy to move data proportional to distance

• **Wire cost to move a bit:**
  - energy = bitrate * Length$^2$ / cross-section area
  - On-Chip (1cm): ~1pJ/bit, 100Tb/s
  - On-Module (5cm): ~2-5pJ/bit, 10Tb/s
  - On-Board (20cm): ~10pJ/bit, 1Tb/s
  - Intra-rack (1m): ~10-15pJ/bit, 1Tb/s
  - Inter-cabinet (2-50m): 15-30pJ/bit, 5-10Tb/s aggregate

• **To move a bit with optics:** target ~1-2pJ/bit for all distance scales

Photonics requires no redrive and passive switch little power

Copper requires to signal amplification even for on-chip connections

Keren Bergman
Switches, Routers, and Networks-on-Chip (example)

On-chip photonics enable ultrahigh-bandwidth, low-power communications for both on- and off-chip signaling, allowing the maximized performance for chip-scale parallel processing systems.

**photonic switching device:** 2×2 switch composed of 2 waveguides, 2 micro-ring resonators, and a crossing

**photonic routing subsystem:** integrated photonic and electronic devices providing multi-wavelength, non-blocking, low-power photonic routing

**CMP system vision:** 3D stack with dedicated communications plane (top layer) housing a photonic NoC

Micro-Ring Switches
- 375 fJ/transition possible
- 400 µW ON-state static power possible
- 250Gbit/s throughput bandwidth demonstrated

Keren Bergman
targeting million-way parallelism changes the selection of algorithms
Technology Trends are Affecting Algorithm Requirements

• Parallel computing has thrived on weak-scaling for past 15 years

• Flat CPU performance increases emphasis on strong-scaling

• Algorithm Requirements will change accordingly
  – Concurrency will increase proportional to system scale (every 18 months)
  – Timestepping algorithms will be increasingly driven towards implicit or semi-implicit stepping schemes
  – Multiphysics/multiscale problems increasingly rely on spatially adaptive approaches such as Berger-Oliger AMR
  – Strong scaling will push applications towards smaller messages sizes – requiring lighter-weight messaging (weak point of MPI)
Where to Find 12 Orders in 10 years? (for simulations of ITER)

David Keyes, Columbia U.

Hardware: 3

- 1.5 orders: increased processor speed and efficiency
- 1.5 orders: increased concurrency
- 1 order: higher-order discretizations
  - Same accuracy can be achieved with many fewer elements

Software: 9

- 4 orders: adaptive gridding
  - Zones requiring refinement are <1% of ITER volume and resolution requirements away from them are ~10^2 less severe
- 3 orders: implicit solvers
  - Mode growth time 9 orders longer than Alfven-limited CFL
OS
Role of OS at Exascale

- Started discussion in “Runtimes” group
  - Why?

- Nearly every important policy decision that would be made by the runtime is mediated by the OS
  - The OS is slow to respond (privilege changes are slow)
  - Doesn’t even allow us to control the key policies required for execution models
  - The OS wasn’t designed for manycore
Old OS Assumptions are Bogus

• Assumes limited number of CPUs that must be shared
  – *Old CW*: time-multiplexing (context switching and cache pollution!)
  – *New CW*: spatial partitioning

• Greedy allocation of finite I/O device interfaces (eg. 100 cores go after the network interface simultaneously)
  – *Old CW*: First process to acquire lock gets device (resource/lock contention! Nondeterm delay!)
  – *New CW*: QoS management for symmetric device access

• Background task handling via threads and signals
  – *Old CW*: Interrupts and threads (time-multiplexing) (inefficient!)
  – *New CW*: side-cores dedicated to DMA and async I/O
Old OS Assumptions are Bogus

• Fault Isolation
  – Old CW: CPU failure --> Kernel Panic (will happen with increasing frequency in future silicon!)
  – New CW: CPU failure --> Partition Restart (partitioned device drivers)

• Inter-Processor Communication
  – Old CW: invoked for ANY interprocessor communication or scheduling
  – New CW: direct HW access mediated by hypervisor

• Parallel Interrupt Dispatch
  – Old CW: invoked for ANY interprocessor communication or scheduling
  – New CW: direct HW access mediated by hypervisor
  – dedicated to DMA and async I/O
Example of OS Misbehavior

- Scalable I/O for massively concurrent systems requires modification of default OS policies
  - Many issues with coordinating access to disk within node (on chip or CMP)
  - Mutex locks and greedy resource allocation policies will not do! (its like rugby where device == the ball) could solve with app-aware scheduling
  - OS will need to devote more attention to QoS for cores competing for finite resource

<table>
<thead>
<tr>
<th>nTasks</th>
<th>I/O Rate 16 Tasks/node</th>
<th>I/O Rate 8 tasks per node</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>-</td>
<td>131 Mbytes/sec</td>
</tr>
<tr>
<td>16</td>
<td>7 Mbytes/sec</td>
<td>139 Mbytes/sec</td>
</tr>
<tr>
<td>32</td>
<td>11 Mbytes/sec</td>
<td>217 Mbytes/sec</td>
</tr>
<tr>
<td>64</td>
<td>11 Mbytes/sec</td>
<td>318 Mbytes/sec</td>
</tr>
<tr>
<td>128</td>
<td>25 Mbytes/sec</td>
<td>471 Mbytes/sec</td>
</tr>
</tbody>
</table>
Intersection with Execution Models

- **OS prevents emergence of novel/scalable execution models**
  - Example: Scheduler is privileged (outside of control of the Application)
  - OS also controls locality management policy (use laborious process pinning to fix it)

- **Need to give more control to application or runtime environment to manage scheduling**
  - However, need to grant that control without compromising security and fault isolation
  - With thousands of cores, there are new opportunities to maintain security and fault isolation via physical (spatial) partitions

- **We cannot do research into novel execution models without at least a provisional thin-OS as a research platform**
  - Impractical to hack directly on the Linux OS scheduler (experiments often are inconclusive)
  - In-effect ceding more policy control over to the OS and
Spatial Partitioning

- Spatial Partition: group of processors acting within hardware boundary
  - Boundaries are “hard”, communication between partitions controlled
  - Anything goes within partition
- Each Partition receives a vector of resources
  - Some number of dedicated processors
  - Some set of dedicated resources (exclusive access)
    - Complete access to certain hardware devices
    - Dedicated raw storage partition
  - Some guaranteed fraction of other resources (QoS guarantee):
    - Memory bandwidth, Network bandwidth
    - Fractional services from other partitions
Deconstructing the OS
(already examples in embedded space)

- Normal Components split into pieces
  - Device drivers (Security/Reliability)
  - Network Services (Performance)
    - TCP/IP stack
    - Firewall
    - Virus Checking
    - Intrusion Detection
  - Persistent Storage (Performance, Security, Reliability)
  - Monitoring services
    - Performance counters
    - Introspection
  - Identity/Environment services (Security)
    - Biometric, GPS, Possession Tracking
- Applications Given Larger Partitions
  - Freedom to use resources arbitrarily
Inhibitors to Change

• Replacing the OS tends to be an all-or-nothing proposition
  – Modern OS’s have accreted years of capabilities
  – Bad track record (remember PINK?)

• However, we will make zero progress on alternative execution models and associated runtimes if we don’t have a minimal research substrate
  – Impractical, resource intensive, and inconclusive to keep hacking on the Linux kernel (scheduler etc..) to perform these experiments