Architectural Directions for the Post Exascale Decade
Future system directions.

This talk focus on architecture and microarchitecture opportunities primarily.

- The importance of addressing memory
- Low level techniques to move information cheaper
- Program model enabling for lower energy
- System concepts resulting from new architectures and technologies.
Traditional DIMM Memory bandwidth not keeping up

Based on Top500 data (www.top500.org)
Memory is changing how we architect systems

<table>
<thead>
<tr>
<th>Capacity</th>
<th>32 GB DRAM DIMMS</th>
<th>High BW Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth</td>
<td>16 GB/s</td>
<td>250 GB/s</td>
</tr>
<tr>
<td>BW/Capacity</td>
<td>½ to 1</td>
<td>30 to 60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DRAM DIMMS</th>
<th>High BW Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost/Capacity</td>
<td>1x</td>
</tr>
<tr>
<td>Cost/BW</td>
<td>1x</td>
</tr>
<tr>
<td>Power/bit</td>
<td>1x</td>
</tr>
</tbody>
</table>

There is little reason over time to utilize DRAM DIMMS. High BW memory is constrained to reside inside package.

Need to...
- Get enough capacity to support applications
- Develop correct integrated fabric to support
Repartitioning of the design

This is how we will be seeing systems built in the near future.
The CPU is typically logically partitioned into subdomains for tasks.

The size of the hardware based coherence region is larger than most MPI based jobs require.

While there is a small amount of power associated with coherency, it is not the main concern here.
Repartitioning of the design

Smaller CPU die has no first order impact on programmability or user models.

It does require OS innovation.
Repartitioning of the design

By repartitioning the way build nodes...

We can have all DRAM integrated into nodes allowing for 15x more memory BW.

The memory capacity per core does not need to be sacrificed (any more than with traditional DRAM)

The biggest hurdle is to recognize that adding core adds power which is not general purpose performance as other resources can not be similarly increased
New memory technologies will play a critical role in the future

RAS: New technologies will allow for very cost effective fast failure recovery. This will be augmented by increasing fault detection coverage leading to systems with high system availability for user cycles.

IO and workflow: New technologies will allow for large storage like capacity with BW closer to memory to be integrated into systems. This will allow for data analytics at scale and HPC to share a common infrastructure.

Path to better memory: DRAM technology will likely be pushed into a smaller fraction of the system as new technologies transition from capacity/$ focus to also having sufficient BW/capacity.
Energy efficiency ... yes it will be the problem

A repartitioned compute will have very good energy savings. But we are still left with a scaling challenge over time.

Many revolutionary technologies are being looked at as alternatives to CMOS transistors. They all have a long way to go.

Historically we have continued to innovate and integrate to get better energy efficiency. This will not end.

Looking at some possibilities here...
Encoding data for energy efficiency

We encode data for resiliency, can we do the same for energy? This adds wires but we can reduce the switching rate for wires in aggregate with this.

4 wires (16 states) encoded onto 8 wires shown... No loss in frequency but requires more area.
Data encoding for lower energy

Switching energy scaling for data transmission with optimal encoding

**Based on analytic model**
Informational approach to lower energy

A simple example provides food for thought...

Energy is needed to move information...

P = Cv^2 f (per wire)
E = Cv^2 (frequency independent)

Only way to improve is to reduce capacitance or voltage?
Utilizing time for encoding information

Total energy is equivalent to one wire changing
And… Cap is actually an average assuming neighboring tracks so there is additional gain.
Informational approach to lower energy

**Based on analytic model**
Exploit the “unexploited architectural dimension” of HPC.

Take a lesson from MPI...
Success comes from simplicity of constraining same code to run on each node.

Concept : Force highly predictive execution model through..

• Execution fundamentally repetitive on long time scale.
• Compiler and PM enforce repetitive structure. (push complexity of execution into preprocessing)
• Hardware optimized for this type of operation. (yes exceptions tolerated)
  -- time step based calculations
  -- ensemble generation.
Future system directions. (conclusion)

- Memory technologies will allow for at least a 10x -100x step but we need to rethink the memory system to exploit this.
- Power will be the problem and systems will likely push another 2x-4x. Beyond that energy costs will dominate.
- There are multiple ways to get an additional 2-4x improvement in energy efficiency through architecture.
- Exploitation of predictable dependencies could offer high single thread performance in a constrained program model.

Bottom line…. Can see path to 10 EF/s and likely 100 EF/s
Legal Disclaimers

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at https://www-ssl.intel.com/content/www/us/en/high-performance-computing/path-to-aurora.html.

Intel, the Intel logo, Xeon, Intel Xeon Phi and Intel Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States or other countries.

*Other names and brands may be claimed as the property of others.

© 2015 Intel Corporation