Lecture 04
Interconnect Overhead

Specific topics include a short review of logic scaling, the impact of technology scaling on interconnect, how interconnect scaling impacts the current solution to problems associated with logic scaling (multi-core architectures), and information processing “tokens”
NW topologies

Dally Paper Slides

Preferred NW configurations

<table>
<thead>
<tr>
<th>Configuration</th>
<th>$H$</th>
<th>$t_r$</th>
<th>$B_C$</th>
<th>$w$</th>
<th>$B_B$</th>
<th>$T_c$</th>
<th>$T_s$</th>
<th>$T_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh</td>
<td>$6\frac{1}{4}$</td>
<td>2</td>
<td>16</td>
<td>192</td>
<td>3,072</td>
<td>5.3</td>
<td>3</td>
<td>17.8</td>
</tr>
<tr>
<td>MeshX2</td>
<td>$6\frac{1}{4}$</td>
<td>2</td>
<td>32</td>
<td>192</td>
<td>6,144</td>
<td>5.3</td>
<td>3</td>
<td>17.8</td>
</tr>
<tr>
<td>Torus</td>
<td>5</td>
<td>2</td>
<td>32</td>
<td>288</td>
<td>9,216</td>
<td>4.0</td>
<td>2</td>
<td>14.0</td>
</tr>
<tr>
<td>CMesh</td>
<td>$3\frac{1}{8}$</td>
<td>3</td>
<td>16</td>
<td>288</td>
<td>4,608</td>
<td>2.1</td>
<td>2</td>
<td>11.5</td>
</tr>
<tr>
<td>CMeshX2</td>
<td>$3\frac{1}{8}$</td>
<td>3</td>
<td>32</td>
<td>288</td>
<td>9,216</td>
<td>2.1</td>
<td>2</td>
<td>11.5</td>
</tr>
<tr>
<td>FTree</td>
<td>$4\frac{3}{8}$</td>
<td>2</td>
<td>64</td>
<td>144</td>
<td>9,216</td>
<td>4.4</td>
<td>4</td>
<td>13.1</td>
</tr>
<tr>
<td>FClos</td>
<td>$4\frac{3}{8}$</td>
<td>2</td>
<td>32</td>
<td>144</td>
<td>4,608</td>
<td>3.5</td>
<td>4</td>
<td>12.2</td>
</tr>
</tbody>
</table>
(a) Completion Time by Pattern

(b) Chip Area

(c) Network Power Dissipation

(d) Area-Delay Metric
(e) Energy Delay Metric

Figure 11: Workload Packet Latency Distribution for Uniform Random Traffic Pattern

Figure 12: Offered Latency for CMeshX2 Network

Amdahl’s Law Slides
Impediments to Parallel Performance

- **Contention for access to shared resources**
  - i.e. multiple accesses to limited # of memory banks may dominate system scalability

- **Programming languages, environments, & methods:**
  - Need simple semantics that can expose computational properties to be exploited by large-scale architectures

- **Algorithms**
  - What if you write good code for a 4-core chip, and then get an 8-core chip?

- **Cache coherency**
  - P1 writes, P2 can read
    - Protocols can enable $ coherency but add overhead

Overhead where no actual processing is done.

Recent multi-core die photos
(Route packets, not wires?)

- Likely to see HW support for parallel processor configurations:
  - Coherency
  - On-chip IC NWs

Overhead where no actual processing is done.

Impediments to Parallel Performance

- **Latency**
  - Is already a major source of performance degradation
  - Architecture charged with hiding local latency
    - (that’s why we talked about registers & caches)
  - Hiding global latency is also task of programmer
    - (i.e. manual resource allocation)

- **Today:**
  - access to DRAM in 100s of CCs
  - round trip remote access in 1000s of CCs
  - multiple clock cycles to cross chip or to communicate from core-to-core
    - Not “free”

Recent multi-core die photos
(Route packets, not wires?)

- Takes advantage of 8 voltage and 28 frequency islands to allow independent DVFS of cores and mesh. As performance scales, the processor dissipates between 25 W and 125 W. 567 mm² processor on 45 nm CMOS integrates 40 IA-32 cores and 4 DDR3 channels in a 2D-mesh network. Cores communicate through message passing using 384 KB of on-die shared memory. Fine-grain power management

Impediments to Parallel Performance

- All ‘ed items also affect Fraction\text{parallelizable}
  - (and hence speedup)

  \[
  \text{Speedup} = \frac{1}{[1 - \text{Fraction}_{\text{parallelizable}}] + \frac{\text{Fraction}_{\text{parallelizable}}}{N}}
  \]
Multi-core only as good as algorithms that use it