Architecture and Performance of the TILE-Gx72 Manycore Processor

Matthew Mattina
CTO
Tilera

Hot Interconnects 2013
Agenda

• Overview

• iMesh Architecture

• Performance Results
TILE-Gx72™: At-a-Glance

- Tile Array: 72x 64-bit cores
  - Tile = core + 256KB L2 cache + iMesh interface
- 1.2GHz, TSMC 40nm HPM
  - 75W TDP
- mPIPE™: Wirespeed programmable packet processing and load balancing engine
  - Dynamic flow affinity
- 8x 10GbE XAUI ports
  - Feeds packets into the mPIPE, 120MPPS
- MiCA™: Cryptographic and RSA Engines
  - 44K keys per second, zero core resources
- 6x PCIe ports
  - 96 Gbps of dedicated PCIe and SR-IOV support
- 4x DDR3 controllers @ 1600MT/sec
  - > 50GB/s main memory BW
- Standard SMP Linux, C/C++, Java, gdb,...
- World’s Highest Single Chip CoreMark™ score

© 2013 Tilera Corporation.
Markets and Solutions

Heterogeneous Devices
• TILE-Gx72 data plane + x86 control plane
  – Application Delivery Controllers
  – Firewalls
  – Server adaptor cards
  – Emerging SDN/NFV

Homogeneous Devices
• TILE-Gx72 data and control plane
  – Single-chip VPN/Router/Firewall
  – IDS/IPS Appliances
  – Video Transcoding Bridge
Agenda

• Overview

• iMesh Architecture

• Performance Results
iMesh: Scaling to 72 cores and beyond

- iMesh: Multiple mesh networks and cache coherence protocol interconnecting all on-die components

- Single shared global physical address space; direct cache access; full HW cache coherence

- Scalable: more tiles = more interconnect bandwidth and more shared cache BW

- Three protocol classes, three physical meshes, three different widths: keeps iMesh router simple and fast
iMesh: Physical and Link Layer

- Each iMesh interface is 5x5 xbar, connects 4 neighbors + core, point-to-point wires, low power
- Fast arbiter: single cycle hop, same clock as core
- Source-based, wormhole routing, route headers coded for high-speed
- Link level FIFOs and per-hop flow control
- Handplaced M7/M8 routed over L2 cache minimizes wiring channel area
- Regular design accelerates time to market: build/verify one Tile, then replicate

<table>
<thead>
<tr>
<th>Network Name</th>
<th>Link Width, Bisection BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDN</td>
<td>128b, 346 GB/s</td>
</tr>
<tr>
<td>RDN</td>
<td>112b, 302 GB/s</td>
</tr>
<tr>
<td>QDN</td>
<td>64b, 173 GB/s</td>
</tr>
</tbody>
</table>
iMesh: Caching and Coherence

- Every Tile contains private 32KB L1 I and D caches and 256KB L2 cache
- L2 caches private L2 lines and global “L3” lines
- Distributed coherence directory tracks sharers, invalidates shared copies on writes
- Flexible cache line distribution
iMesh Protocol - Remote Write Hit with Sharer

**Tile 0**

<table>
<thead>
<tr>
<th>Directory</th>
<th>TAG</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0x0</td>
</tr>
</tbody>
</table>

**Tile 1**

<table>
<thead>
<tr>
<th>Directory</th>
<th>TAG</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0x0</td>
</tr>
</tbody>
</table>

**Tile 2**

<table>
<thead>
<tr>
<th>Directory</th>
<th>TAG</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 1</td>
<td></td>
<td>0x0</td>
</tr>
</tbody>
</table>

Store V, 0x1

V (a virtual address) → TLB → P (a physical address), Home Tile coordinates

© 2013 Tilera Corporation.
iMesh Protocol - Remote Write Hit with Sharer

Tile 0

<table>
<thead>
<tr>
<th>Directory</th>
<th>TAG</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TAG(P) 0x1</td>
</tr>
</tbody>
</table>

Store V, 0x1

V (a virtual address) → TLB → P (a physical address), Home Tile coordinates

Cache controller writes-through to home on SDN

Tile 1

<table>
<thead>
<tr>
<th>Directory</th>
<th>TAG</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TAG(P) 0x0</td>
</tr>
</tbody>
</table>

Data invalidated!

Tile 2

<table>
<thead>
<tr>
<th>Directory</th>
<th>TAG</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 1</td>
<td></td>
<td>TAG(P) 0x0</td>
</tr>
</tbody>
</table>

Write is processed by home cache and sharer list is inspected

Tile 2

<table>
<thead>
<tr>
<th>Directory</th>
<th>TAG</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TAG(P) 0x0</td>
</tr>
</tbody>
</table>

Inval sent to sharer on QDN

Tile 1

<table>
<thead>
<tr>
<th>Directory</th>
<th>TAG</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TAG(P) 0x0</td>
</tr>
</tbody>
</table>

Local copy is updated

Dir

© 2013 Tilera Corporation.
iMesh Protocol - Remote Write Hit with Sharer

**Tile 0**

- **Directory**
- **TAG**
- **DATA**

<table>
<thead>
<tr>
<th>Tag</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAG(P)</td>
<td>0x1</td>
</tr>
</tbody>
</table>

*Cache controller writes-through to home on SDN*

**Tile 1**

- **Directory**
- **TAG**
- **DATA**

**Local copy is updated**

**Tile 2**

- **Directory**
- **TAG**
- **DATA**

<table>
<thead>
<tr>
<th>Tag</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAG(P)</td>
<td>0x0</td>
</tr>
</tbody>
</table>

*Data invalidated!*

**Tile 3**

**Tile 4**

*Write is processed by home cache and sharer list is inspected*

**Tile 5**

*Inval sent to sharer on QDN*

**Tile 6**

*Inval ack sent over RDN*

**Tile 7**

*Inval sent to sharer on QDN*

**Store V, 0x1**

- **V (a virtual address)**
- **TLB**
- **P (a physical address), Home Tile coordinates**
iMesh Protocol - Remote Write Hit with Sharer

- **Tile 0**
  - Store V, 0x1
  - V (a virtual address) ➔ TLB ➔ P (a physical address), Home Tile coordinates
  - Cache controller writes-through to home on SDN
  - Local copy is updated

- **Tile 1**
  - Write ack sent back to Tile 0 via RDN
  - Data invalidated!

- **Tile 2**
  - Inval ack sent over RDN
  - Inval sent to sharer on QDN
  - Write is processed by home cache and sharer list is inspected

- **Tile 3**
  - Write ack sent back to Tile 0 via RDN

- **Tile 4**
  - Tile 1 is removed from sharers and new data is written to cache!
iMesh at the Movies
Agenda

• Overview

• iMesh Architecture

• Performance Results and Summary
### Performance Results

**TCP Throughput**

- **Graph:** Line chart with cores on the x-axis and Gbps on the y-axis.
- **Note:** 30x performance improvement.

**H.264 Video Encode**

- **Graph:** Line chart with cores on the x-axis and Fps on the y-axis.
- **Note:** 65x performance improvement.

### Application Performance

<table>
<thead>
<tr>
<th>Application</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP Throughput</td>
<td>80Gbps, 30 cores</td>
</tr>
<tr>
<td>h.264 1080p encode</td>
<td>22 channels, 72 cores</td>
</tr>
<tr>
<td>Suricata IDS, policy based rule set</td>
<td>13Gbps, 72 cores</td>
</tr>
<tr>
<td>Packet forwarding with netfilter</td>
<td>80Gbps, 40 cores</td>
</tr>
</tbody>
</table>

---

1. 512 connections, 1500B packets, “echo” server, full-duplex performance
2. “Toys_and_calendar” video sequence, 3Mbps, baseline profile
3. 677 rules, typical traffic
The iMesh interconnect and cache coherence protocol enables Tilera’s Gx family of Manycore processors to efficiently scale performance from the TILE-Gx9 to the TILE-Gx72.
Thank You!