



TILERA[®]

Architecture and Performance of the TILE-Gx72 Manycore Processor

Matthew Mattina
CTO
Tilera

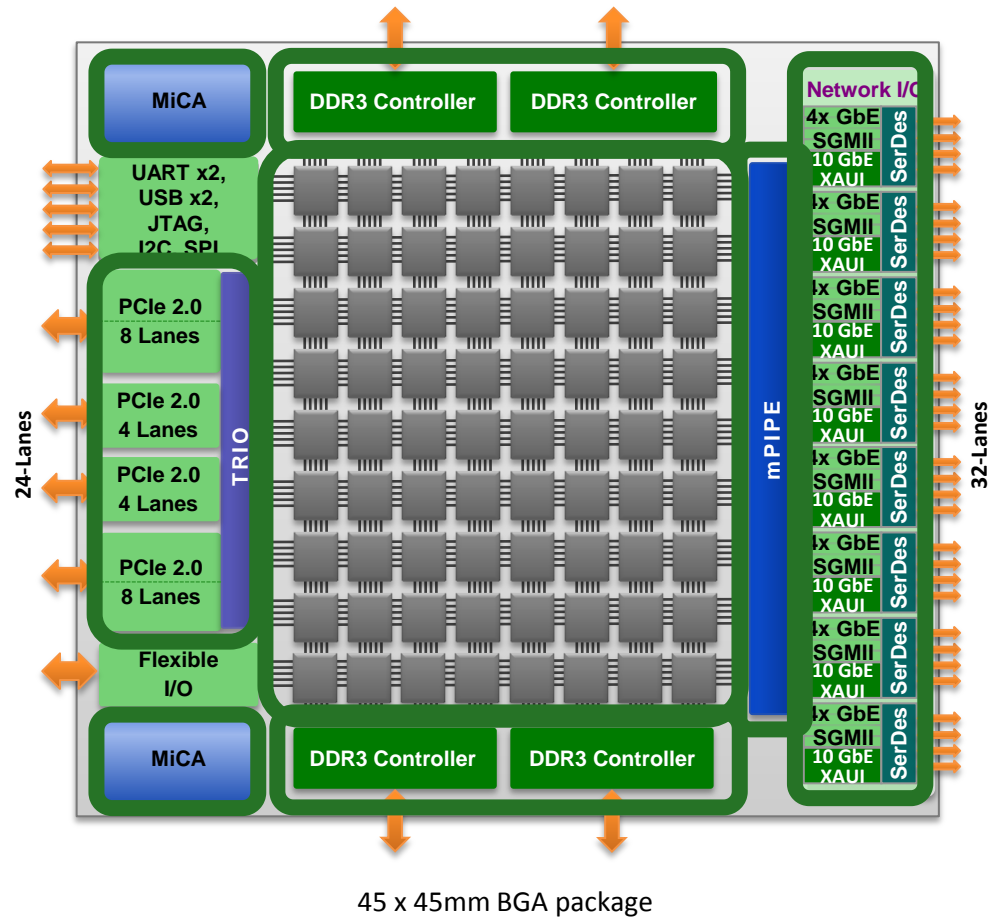
Hot Interconnects 2013

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



Markets and Solutions

Heterogeneous Devices

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



Homogeneous Devices

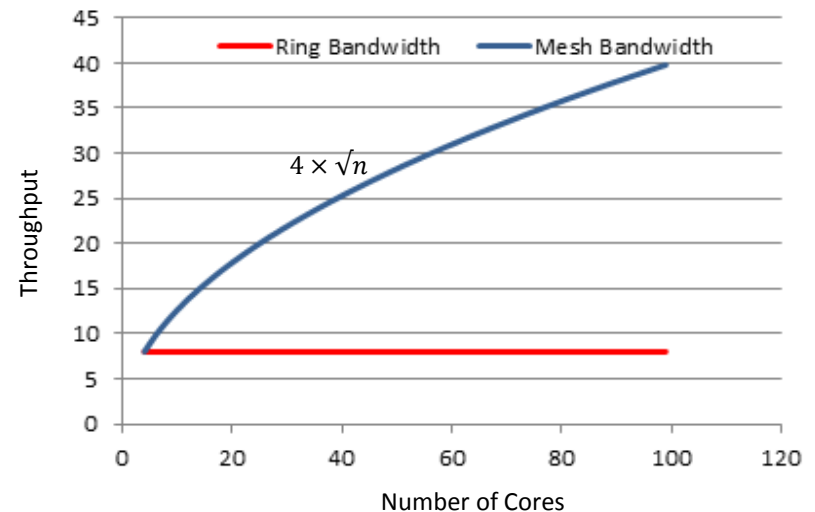
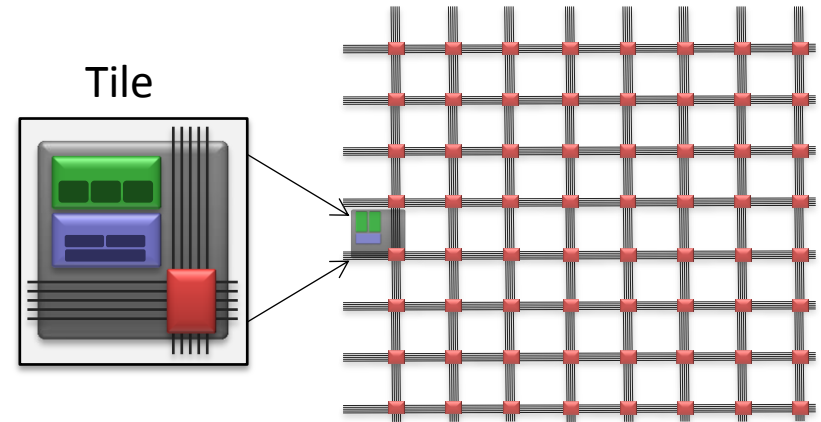
- TILE-Gx72 data and control plane
 - Single-chip VPN/Router/Firewall
 - IDS/IPS Appliances
 - Video Transcoding Bridge

Agenda

- Overview
- iMesh Architecture
- Performance Results

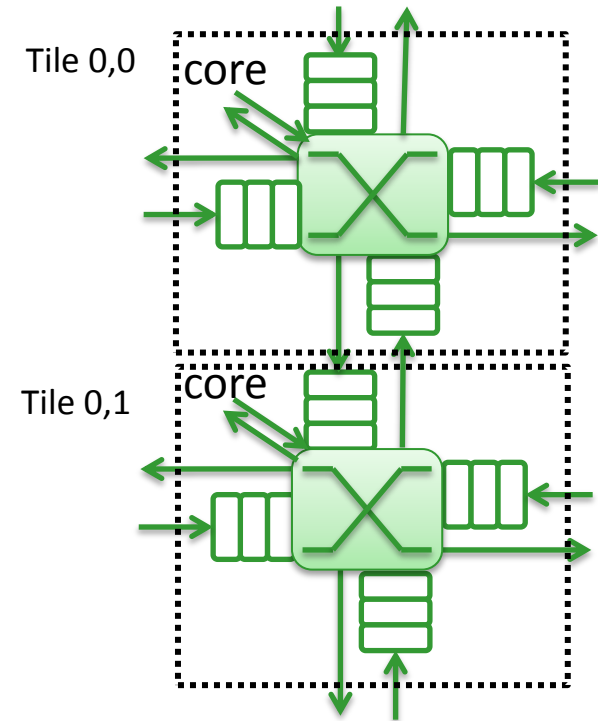
iMesh: Scaling to 72 cores and beyond

- 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

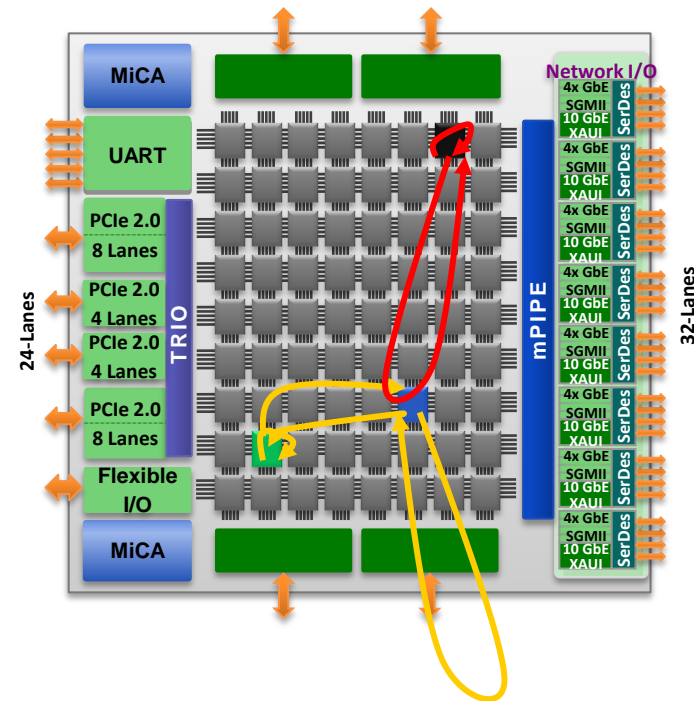
- Each iMesh interface is 5x5 xbar, connects 4 neighbors + core, point-to-point wires, low power
- Fast arbiter: single cycle hop, same clock as core
- Source-based, wormhole routing, route headers coded for high-speed
- Link level FIFOs and per-hop flow control
- Handplaced M7/M8 routed over L2 cache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



| Network Name | Link Width, Bisection BW |
|--------------|--------------------------|
| SDN | 128b, 346 GB/s |
| RDN | 112b, 302 GB/s |
| QDN | 64b, 173 GB/s |

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

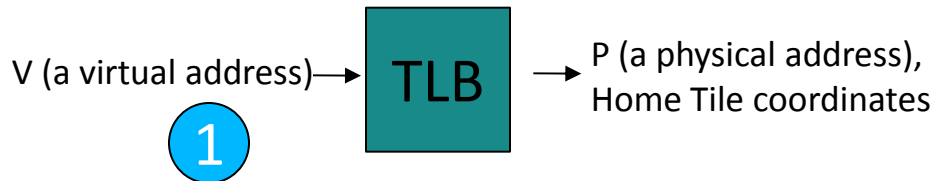
Tile 0

| Directory | TAG | DATA |
|-----------|--------|------|
| | | |
| | TAG(P) | 0x0 |

Tile 1

| Directory | TAG | DATA |
|-----------|--------|------|
| | | |
| | TAG(P) | 0x0 |

Store V, 0x1

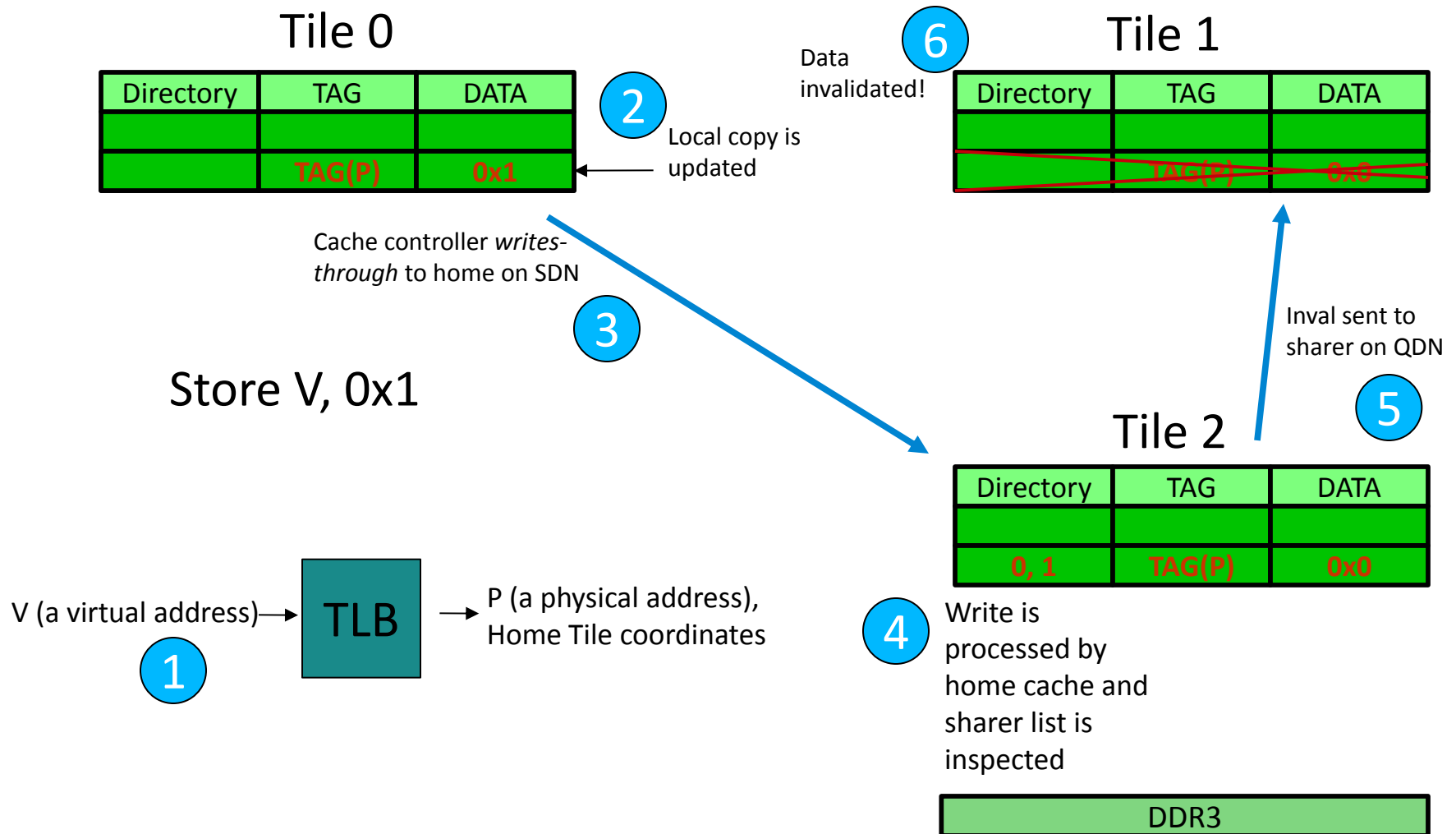


Tile 2

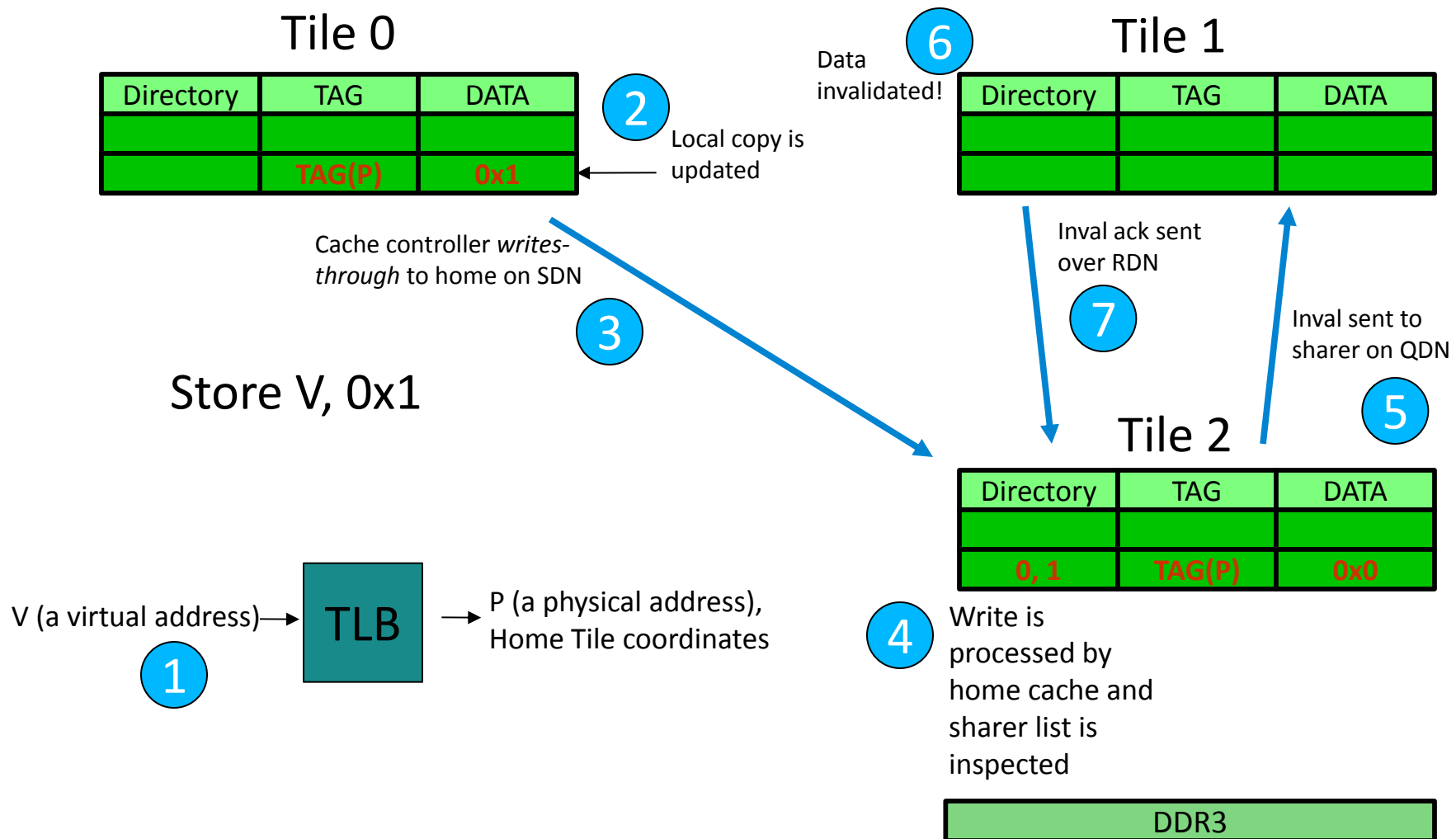
| Directory | TAG | DATA |
|-----------|--------|------|
| | | |
| 0, 1 | TAG(P) | 0x0 |



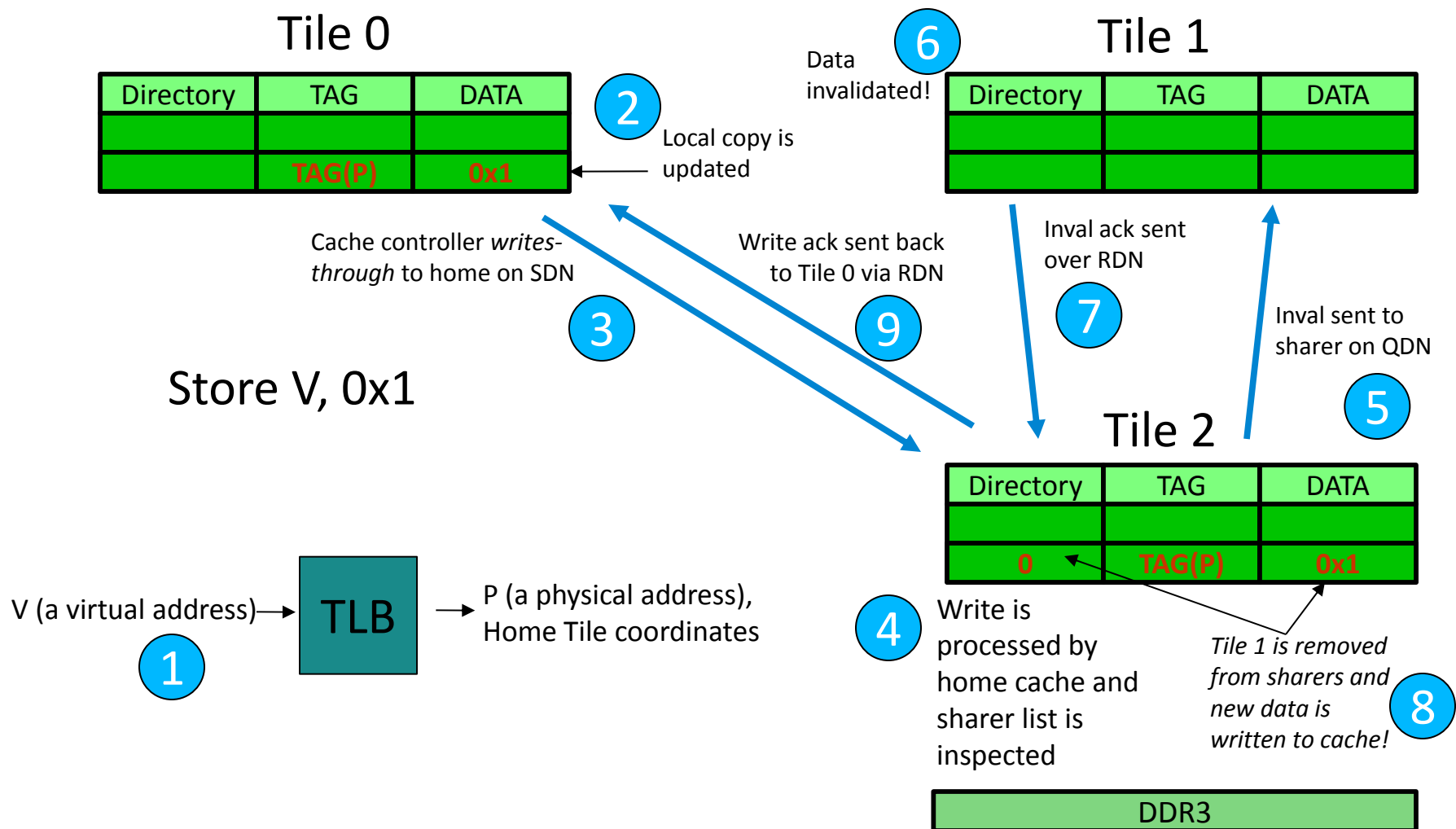
iMesh Protocol - Remote Write Hit with Sharer



iMesh Protocol - Remote Write Hit with Sharer

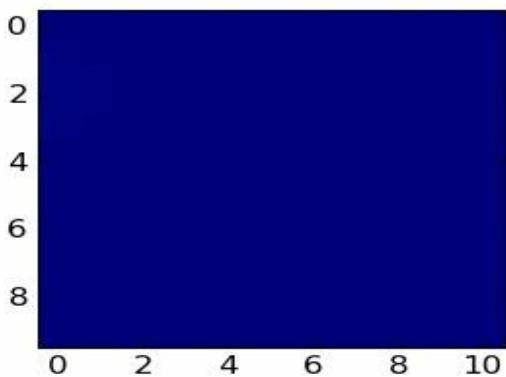


iMesh Protocol - Remote Write Hit with Sharer

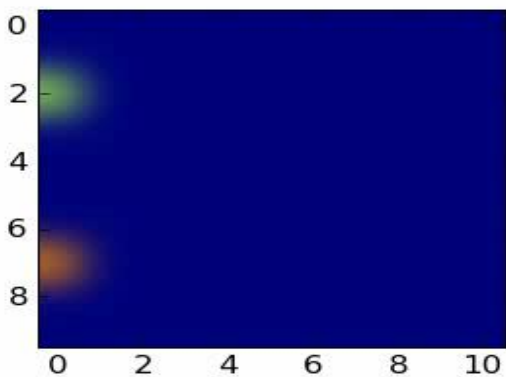
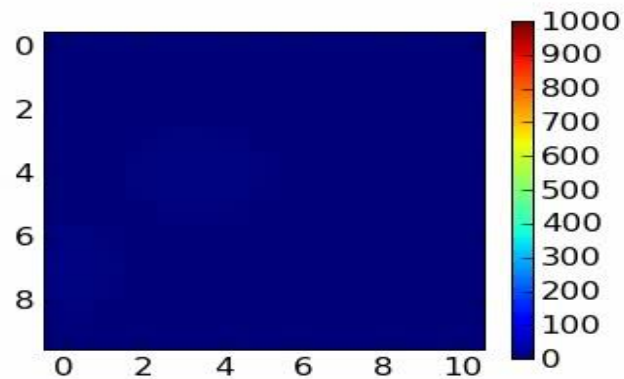


iMesh at the Movies

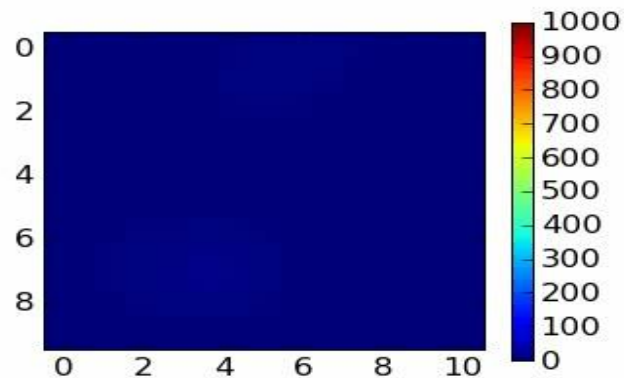
NP Arb



NP Arb



RR Arb, XY Route



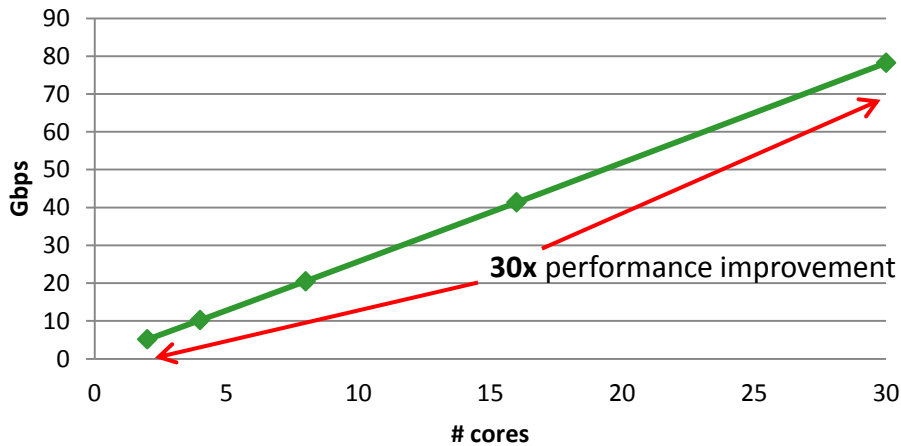
NP Arb

Agenda

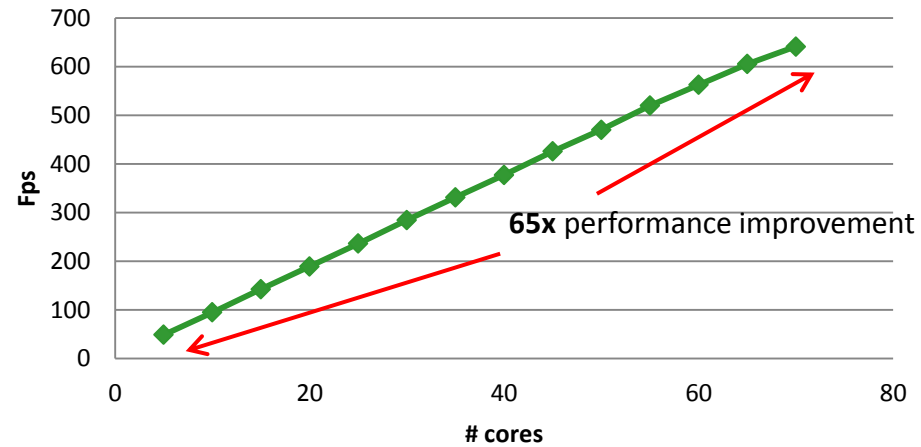
- Overview
- iMesh Architecture
- Performance Results and Summary

Performance Results

TCP Throughput¹



H.264 Video Encode²



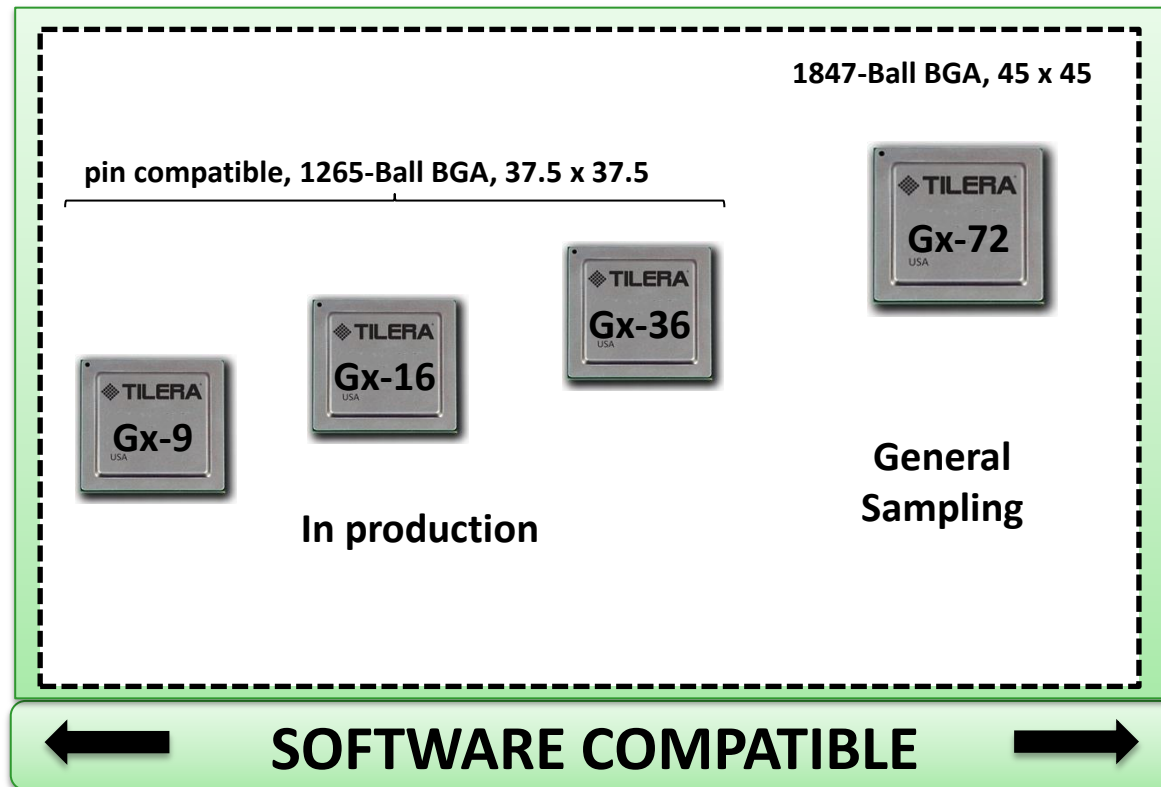
| Application | Performance |
|--|-----------------------|
| TCP Throughput | 80Gbps, 30 cores |
| h.264 1080p encode | 22 channels, 72 cores |
| Suricata IDS, policy based rule set ³ | 13Gbps, 72 cores |
| Packet forwarding with netfilter | 80Gbps, 40 cores |

¹ 512 connections, 1500B packets, "echo" server, full-duplex performance ² "Toys_and_calendar" video sequence, 3Mbps, baseline profile

³ 677 rules, typical traffic

Summary

The iMesh interconnect and cache coherence protocol enables Tiler's Gx family of Manycore processors to efficiently scale performance from the TILE-Gx9 to the TILE-Gx72





Thank You!