Interconnects and Software for Real-Time Embedded Systems

Ken Cain
Mercury Computer Systems
kcain@mc.com

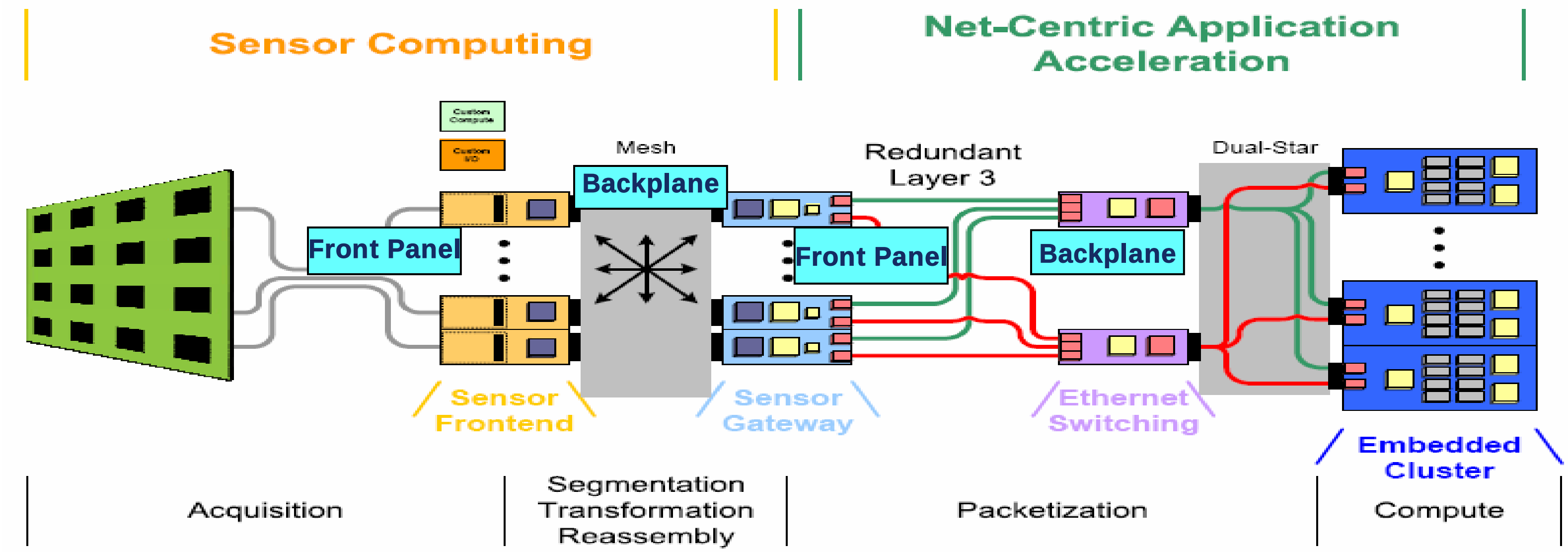
Outline

- **Real-time embedded hardware environments**
- **Current interconnects**
  - Embedded RapidIO characteristics
  - Server iWARP, IB with performance comparison to RapidIO
- **Embedded DMA stack, OS, middleware considerations**
- **Interconnect trends**
- **OpenFabrics software considerations for embedded**
- **Conclusions**

Real-Time Embedded Environments

 www.mc.com

Typical Embedded System Architecture



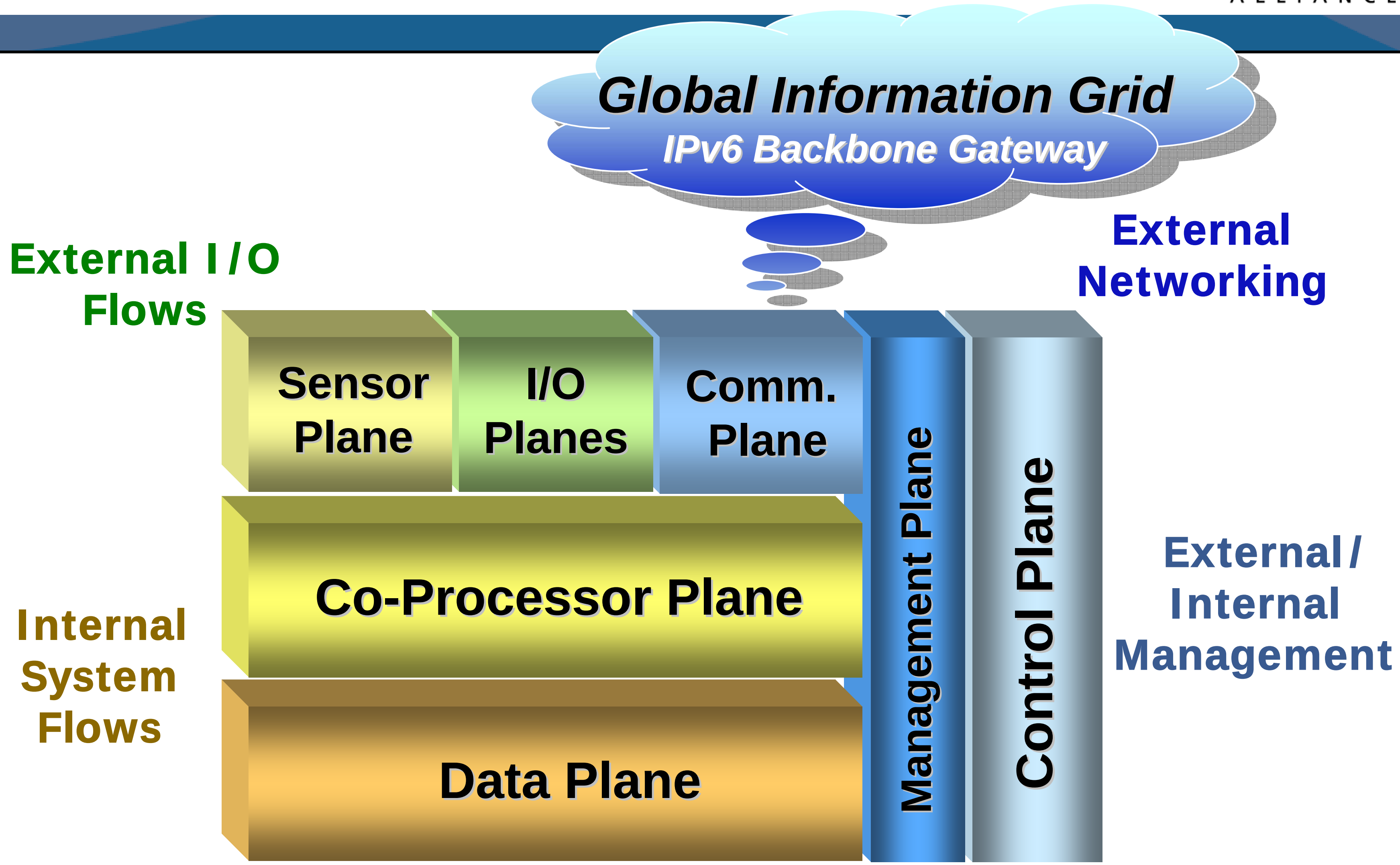
RADAR, sonar, video, etc. sensors
sensor compute modules (FPGA)
A to D converters, RF tuners

VMEbus (VITA 41, 46/48) 3U, 6U boards
single-board computer (SBC) modules
DSP modules (PPC / x86 multicore, GPU)
Switched-fabric module (e.g., sRIO)

- **Significant size, weight, power constraints (not servers!)**
- **Compute, fabric efficiency → smaller or more capable systems**

 www.mc.com

Mercury's Communications Model



RapidIO

Current Embedded Fabric: RapidIO

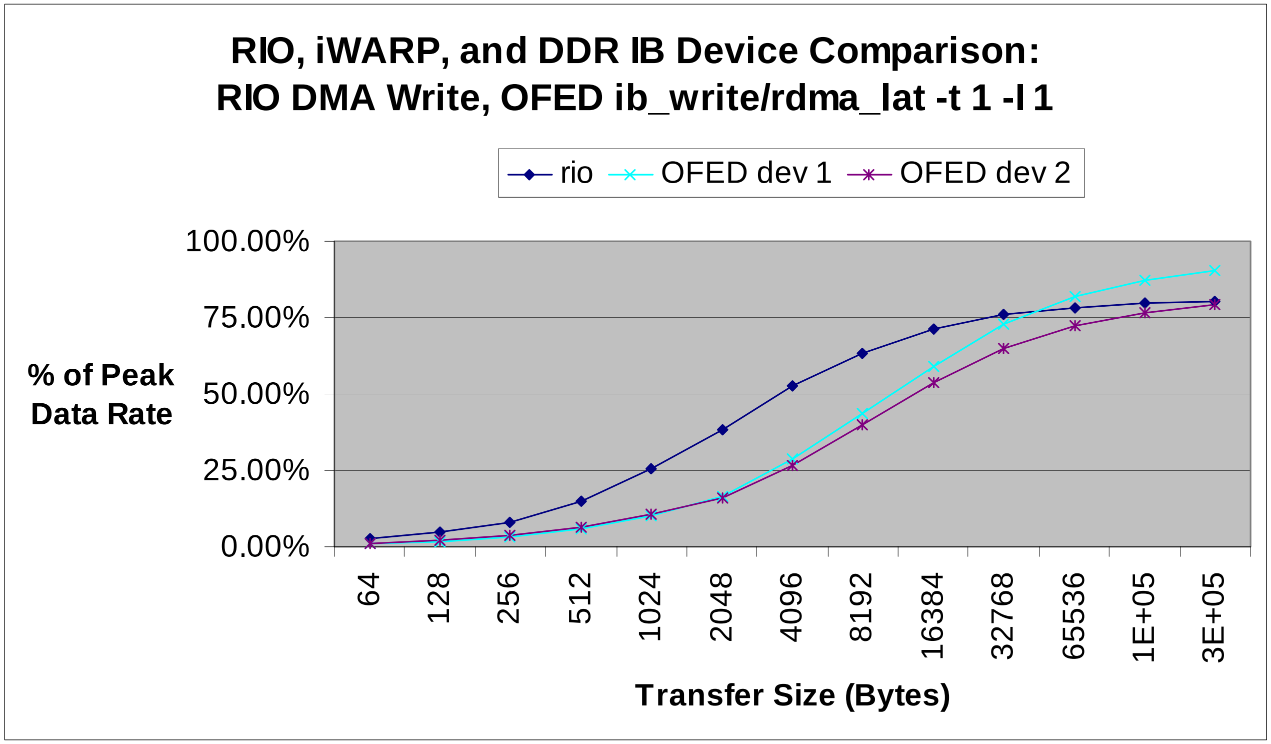
Switched-Fabric Characteristics

- **256-byte max packet size, small headers (12-20 bytes)**
- **Hop-to-hop reliable delivery**
- **End-to-end transaction reliable delivery**

Device Characteristics

- **Examples: Freescale (8548,864x), Mercury PCI-X / PCIe**
- **Ops: DMA write, read, atomic, mailbox/message**
- **Shared Send Queue**
  - Target multiple endpoints in 1 request/chain - efficient, scalable.
- **Hardware atomic queuing (avoid software arbitration)**
- **Uniform striding (local/remote, for submatrix move)**

Small Transfer Performance Comparison

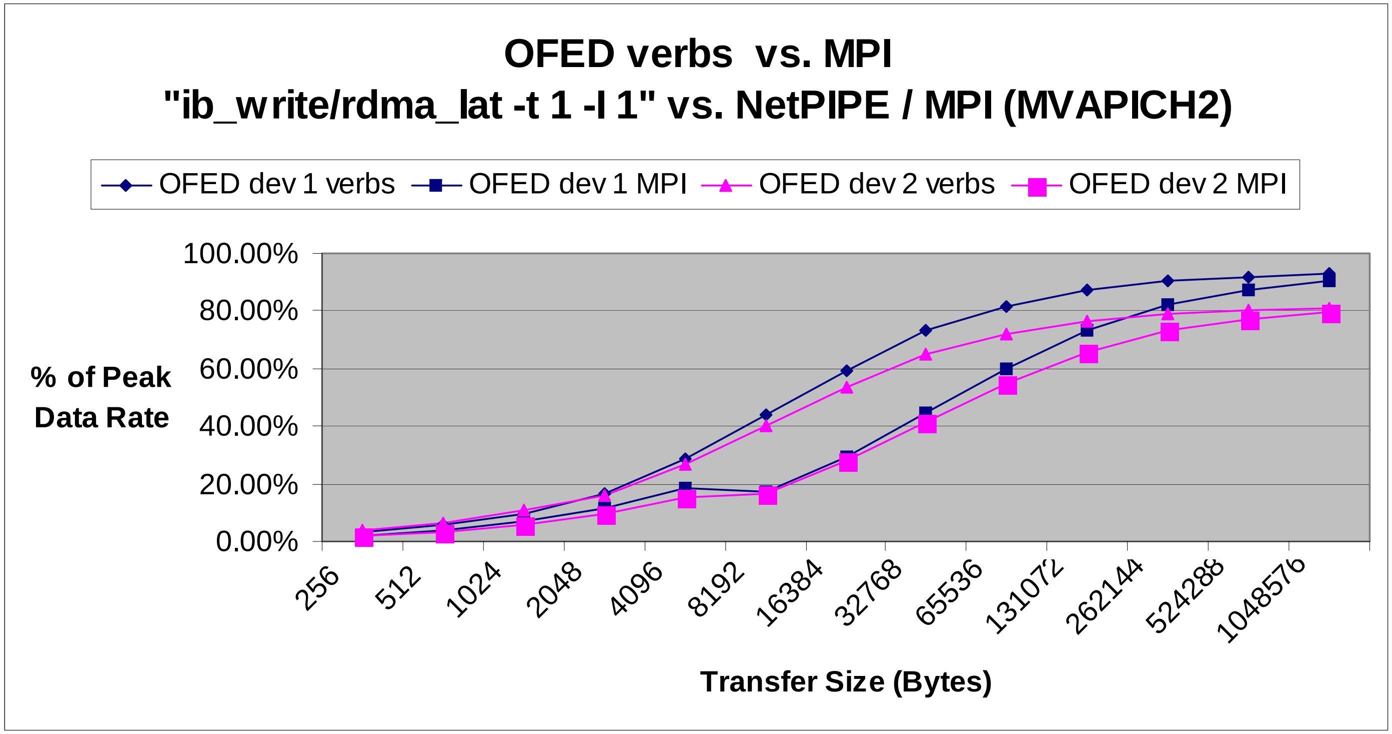


- **These sizes are typical in embedded applications**
- **And small transfers generally important**
  - Efficient producer/consumer protocol, polling completion, etc.

 www.mc.com

Embedded Middleware, OS, DMA

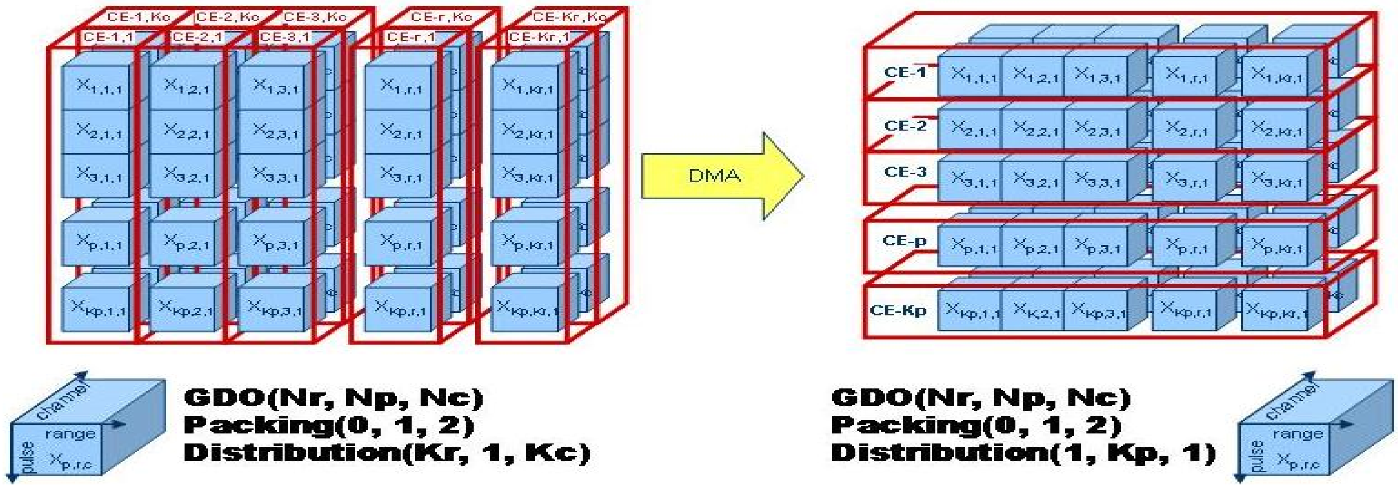
MPI Performance



- **Traditional MPI performance concerns: matching, rendez-vous**
- **Hardware assist for MPI likely will help**
- **Additional middleware constructs also can help**

 www.mc.com

Embedded Middleware Example: DRI



- **DRI = Data Reorganization Interface (www.data-re.org)**
  - Developed by DOD signal/image processing community
- **N-dimensional data decomposition, many to many redistribution**
- **Streamlined producer/consumer "channels"**
  - Outer loop setup/binding, inner loop DMA with multi-buffering
- **First developed for multi-node - similar patterns used within node**
  - data flows to/from accelerators such as FPGA, GPU, Cell B.E. SPEs

 www.mc.com

DMA, Virtual+Physical Memory

OPENFABRICS ALLIANCE

Current Practice

- **Embedded: transfer physically contiguous memory**
  - Manage a pool of contiguous memory outside OS control
  - Good and bad: performance potential, but less flexible
- **Server: transfer virtual/discontiguous memory**
  - Via page locking, physical page lists, registration techniques, etc.

Potential Tactics

- **Embedded**
  - Have middleware allocate contiguous memory (MPI_Alloc_mem)
  - Extend (implement similar techniques, adopt OpenFabrics)
- **Server / OFED**
  - Extend: optimize for the contiguous “DMA memory” case

Real-Time OS in Embedded

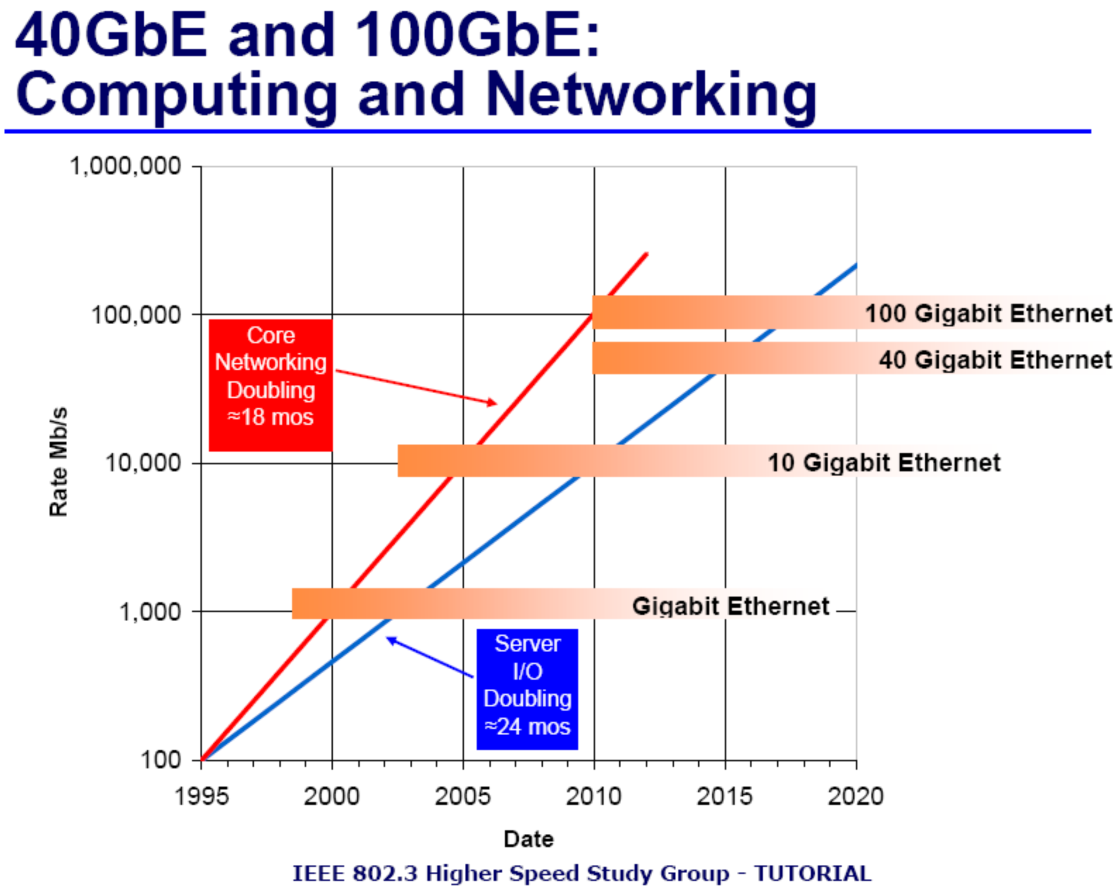
- **RTOS in past systems:**
  - DSP modules use RTOS (ex: Mercury “MC/OS” or VxWorks)
  - SBC modules several OS (Linux, Solaris, Windows, VxWorks)
- **Current view: Linux and VxWorks (mixed / same system)**
- **VxWorks motivation can include**
  - Legacy VME/PCI device drivers, application investment
  - Real-time/predictable behavior of running applications
- **RTOS + IPC Implementation Choices**
  - When low-latency IPC required: common DMA stack (OFED?)
  - When legacy is most important: rely on TCP/IP and middleware

- **The real challenge ahead is efficiency + predictability of whole applications (comm + compute) on multicore/SMP**

Interconnect Trends

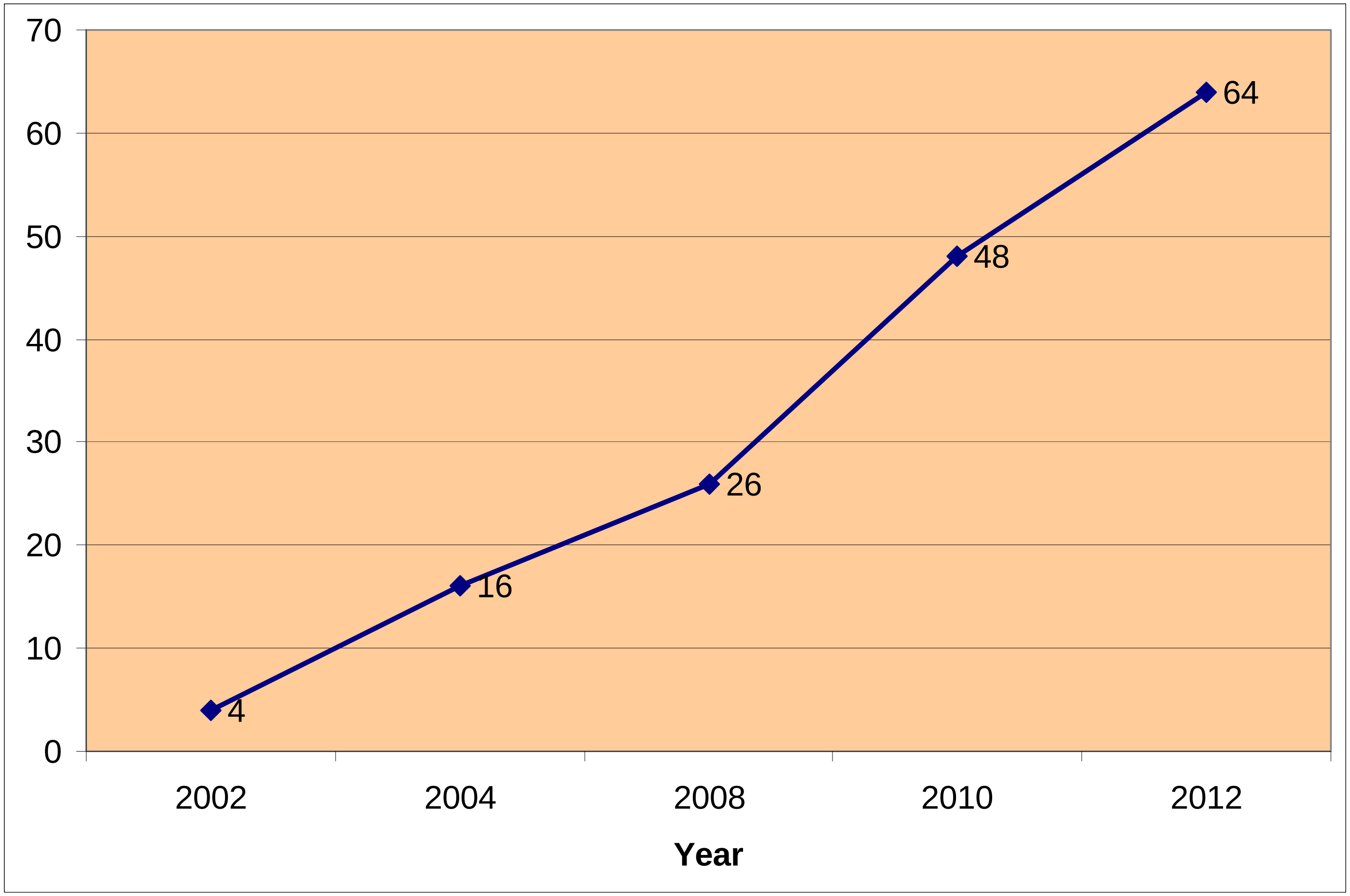
Ethernet Data Rate Trends

Catching up rapidly to embedded fabrics

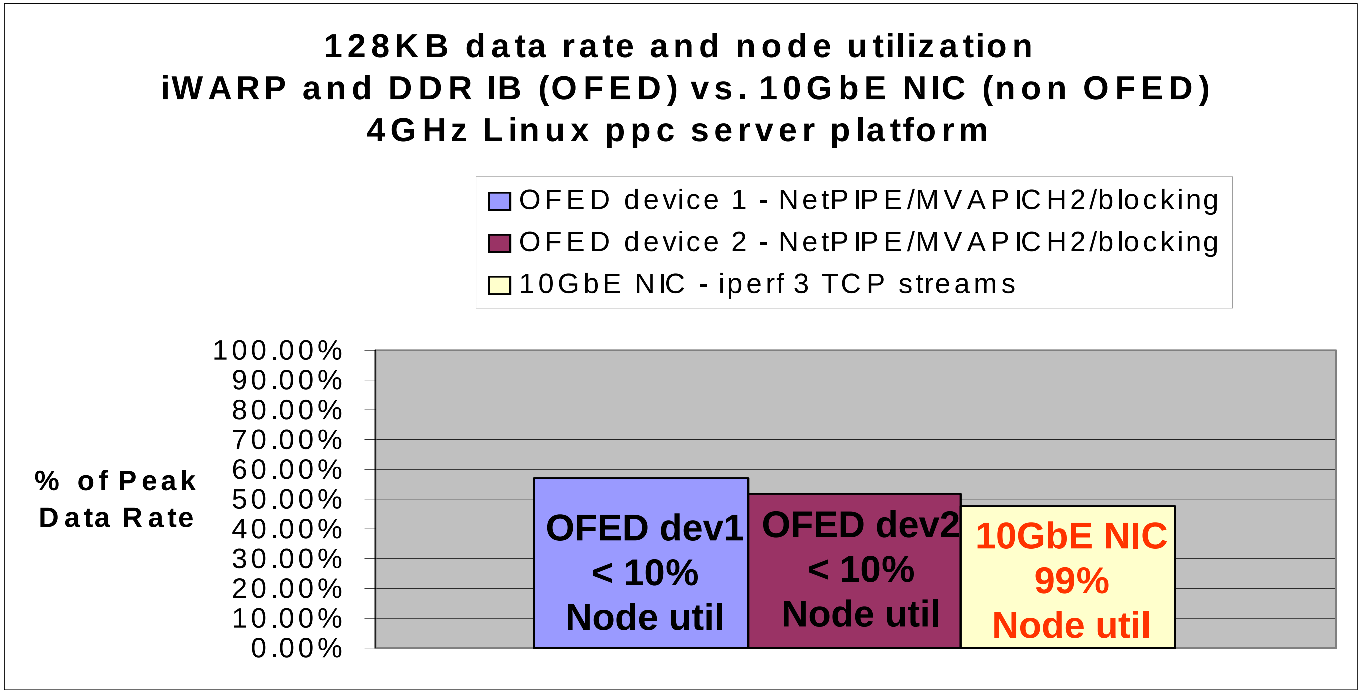


 www.mc.com

10GbE Switch ASIC Port Counts

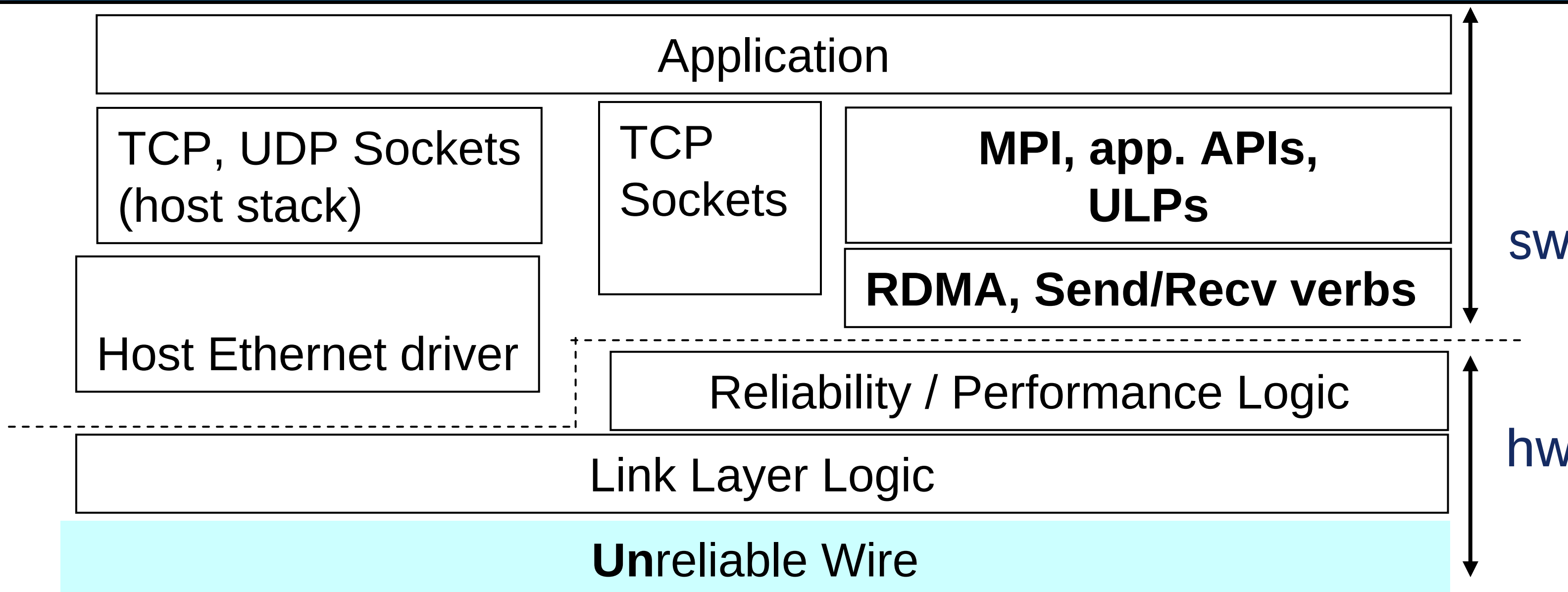


The Case for Offload



- **Anecdotal rule of thumb TCP/IP stack: 1 GHz per 1Gbit/sec**
- **Tested: 99 % utilization on 4 GHz node handling 4.76 Gbit/sec**
- **DMA/Offload still required, although core counts are growing**

Considering “Idealized” Device Layering



- **Need both reliability and extreme (low latency) performance**
  - TCP/IP+TOE better fit for reliability (minimize headers for performance)
- **Concern in dedicating too much hardware to TCP/IP**
  - control/comm plane role vs. other performance-oriented capabilities
- **“R-NIC minus TOE”**

 www.mc.com

Considering OpenFabrics Software for Embedded Systems

OFED for Embedded Systems (1/2)

- **Software only implementations would help:**
  - i.e., OFED over arbitrary network (Ethernet) controller
  - Assume will help factoring (fabric + connection management)
  - Software QP processing facilitates mapping to lightweight device
  - Helps broaden adoption of OpenFabrics software
- **Performance – OFED vs. native platform methods**
  - Very careful (and realistic) benchmarking needed
  - Good internal instrumentation
- **Fencing/ordering semantics OFED vs. native capabilities**
  - Goal: use minimum # transactions to achieve payload/sync order

OFED for Embedded Systems (2/2)

- **Potential extensions – expose unique device capabilities**
  - Local/remote memory stride parameters
  - Multiple destination work request lists
- **Remote I/O virt. address to physical mappings**
  - May not be performed in hardware (if no channel adapter)
  - Would require alternate sw organization to maintain performance
- **Nuts + Bolts**
  - Cross-build of OFED may be needed (especially for PPC)
  - Static libraries - space efficiency on target (RAMdisk, flash fs)

Conclusions

Future Embedded Systems Will Require

- **RapidIO-like properties (latency, offload, predictability)**
- **Higher-speed fabric interfaces (32+ Gbps)**
- **Broad / prioritized ecosystem of protocols, middleware, OS**

OpenFabrics Software is Compelling

- **Demonstrated success in common areas of concern**

Potential Areas of Collaboration with OFA and Community

- **Next-generation interconnect protocols – DMA over Ethernet**
- **Expanded application of OFA software in new domains**