Interconnects

Ken Raffenetti

Software Development Specialist

Programming Models and Runtime Systems Group

Mathematics and Computer Science Division

Argonne National Laboratory

U.S. DOE Potential System Architecture Targets

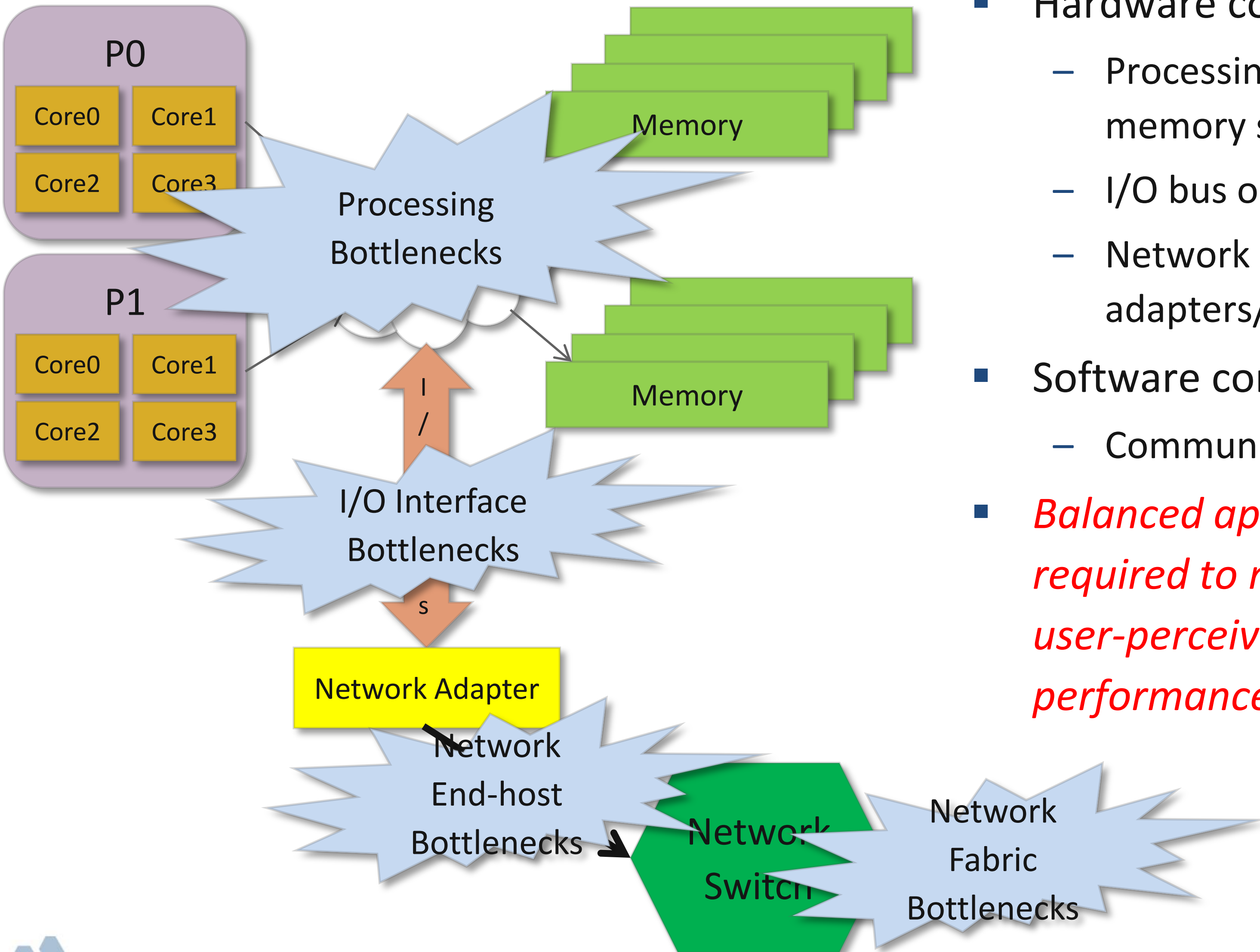
System attributes	2010	2018		2021-2022	
System peak	2 Peta	150-200 Petaflop/sec		1 Exaflop/sec	
Power	6 MW	15 MW		20 MW	
System memory	0.3 PB	5 PB		32-64 PB	
Node performance	125 GF	3 TF	30 TF	10 TF	100 TF
Node memory BW	25 GB/s	0.1TB/sec	1 TB/sec	0.4TB/sec	4 TB/sec
Node concurrency	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	18,700	50,000	5,000	100,000	10,000
Total Node Interconnect BW	1.5 GB/s	20 GB/sec		200GB/sec	
MTTI	days	O(1day)		O(1 day)	
	Current production	***Planned Upgrades (e.g., CORAL)***		***Exascale Goals***	

[Includes some modifications to the DOE Exascale report]

General Trends in System Architecture

- Number of nodes is increasing, but at a moderate pace
- Number of cores/threads on a node is increasing rapidly
- Each core is not increasing in speed (clock frequency)
- Chip logic complexity decreasing (in-order instructions, no pipelining, no branch prediction)
- What does this mean for networks?
  - More cores will drive the network
  - More sharing of the network infrastructure
  - The aggregate amount of communication from each node will increase moderately, but will be divided into many smaller messages
  - A single core will not be able to drive the network fully

A Simplified Network Architecture



- Hardware components
  - Processing cores and memory subsystem
  - I/O bus or links
  - Network adapters/switches
- Software components
  - Communication stack
- *Balanced approach required to maximize user-perceived network performance*

Agenda

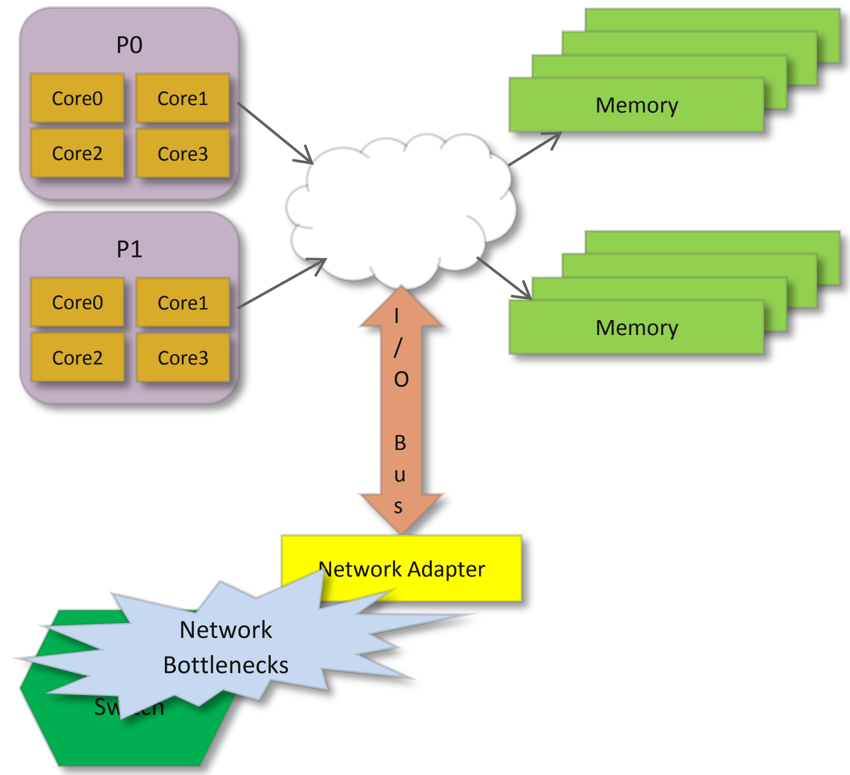
Network Adapters

Network Topologies

Network/Processor/Memory Interactions

Bottlenecks on Traditional Network Adapters

- Network speeds saturated at around 1Gbps
  - Features provided were limited
  - Commodity networks were not considered scalable enough for very large-scale systems

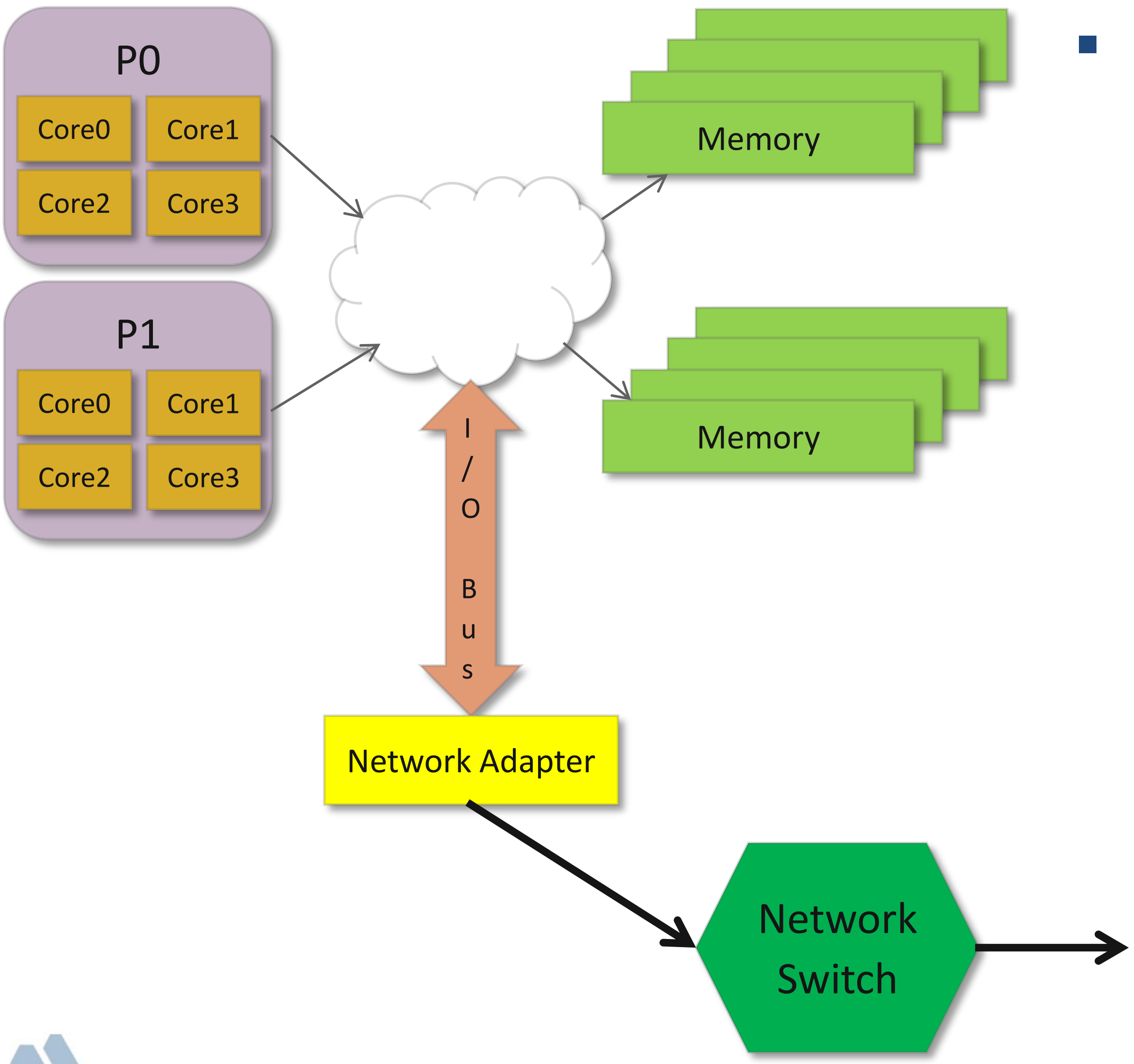


Ethernet (1979 -)	10 Mbit/sec
Fast Ethernet (1993 -)	100 Mbit/sec
Gigabit Ethernet (1995 -)	1000 Mbit /sec
ATM (1995 -)	155/622/1024 Mbit/sec
Myrinet (1993 -)	1 Gbit/sec
Fibre Channel (1994 -)	1 Gbit/sec

End-host Network Interface Speeds

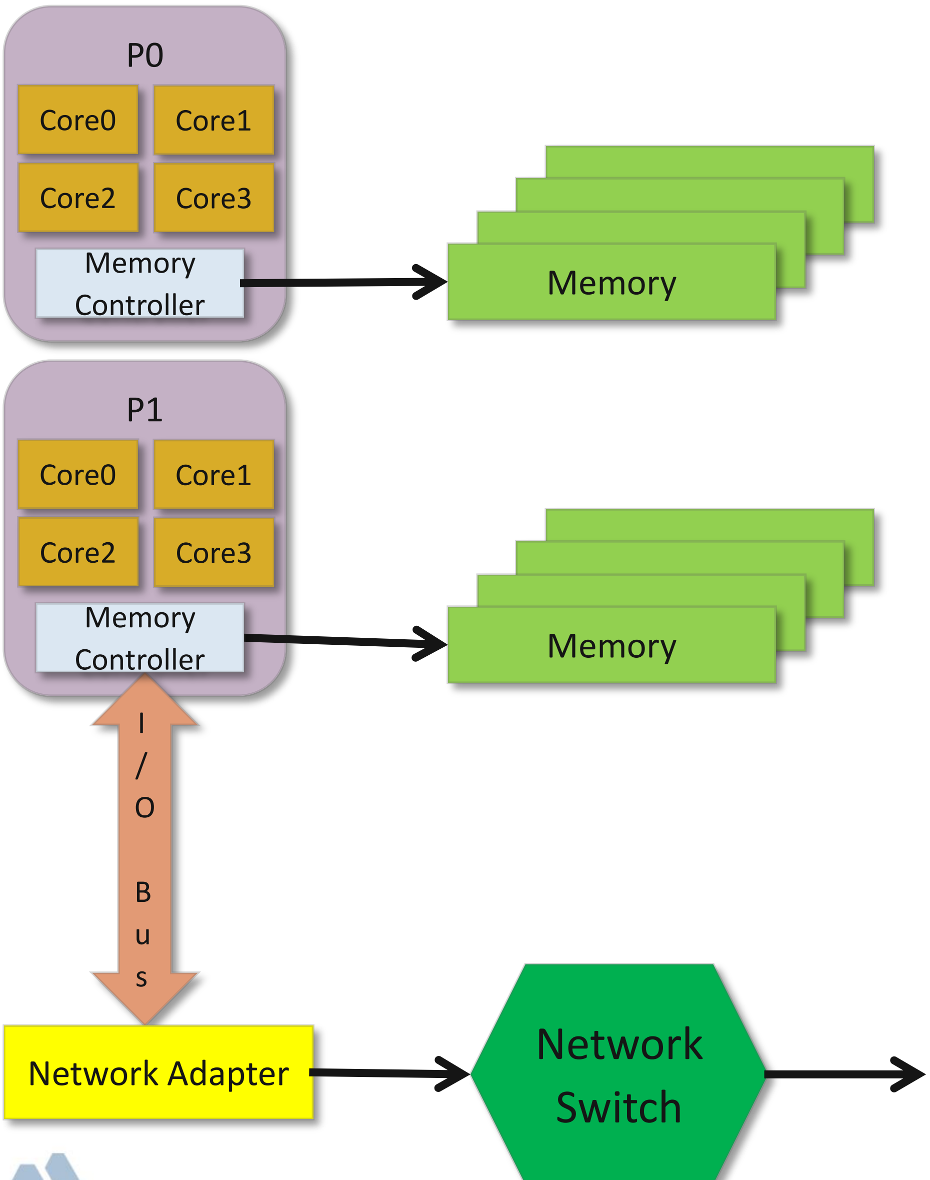
- Recent network technologies provide high bandwidth links
  - InfiniBand EDR gives 100 Gbps per network link
    - Upcoming networks expected to increase that by several fold
  - Multiple network links becoming a common place
    - ORNL Summit and LLNL Sierra machines, Japanese Post T2K machine
    - Torus style or other multi-dimensional networks
- End-host peak network bandwidth is "mostly" no longer considered a major limitation
- Network latency is still an issue
  - That's a harder problem to solve – limited by physics, not technology
    - There is some room to improve it in current technology (trimming the fat)
    - Significant effort in making systems denser so as to reduce network latency
- Other important metrics: message rate, congestion, …

Simple Network Architecture (past systems)



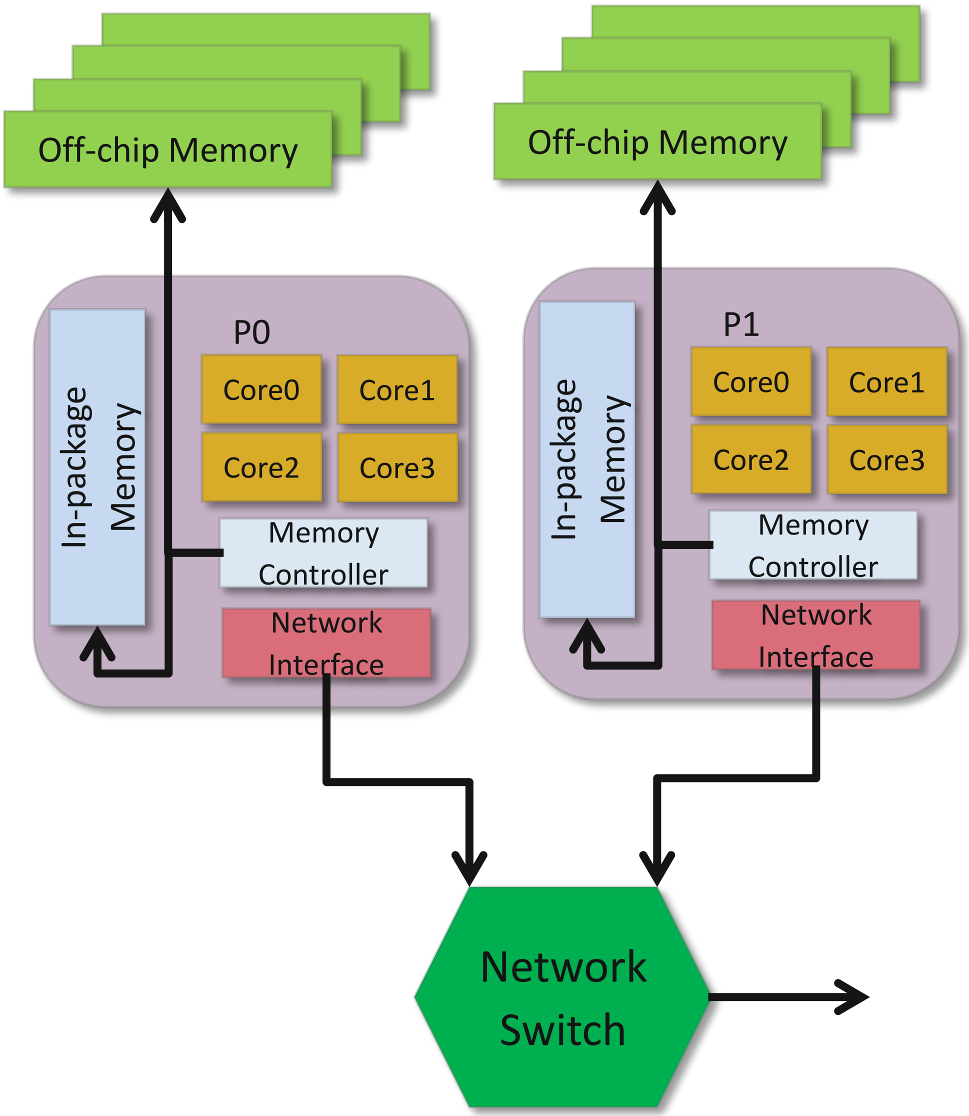
- Processor, memory, network are all decoupled

Integrated Memory Controllers (current systems)



- In the past 10 years or so, memory controllers have been integrated on to the processor
- Primary purpose was scalable memory bandwidth (NUMA)
- Also helps network communication
  - Data transfer to/from network requires coordination with caches
- Several network I/O technologies exist
  - PCIe, HTX, NVLink
  - Expected to provide higher bandwidth than what network links will have

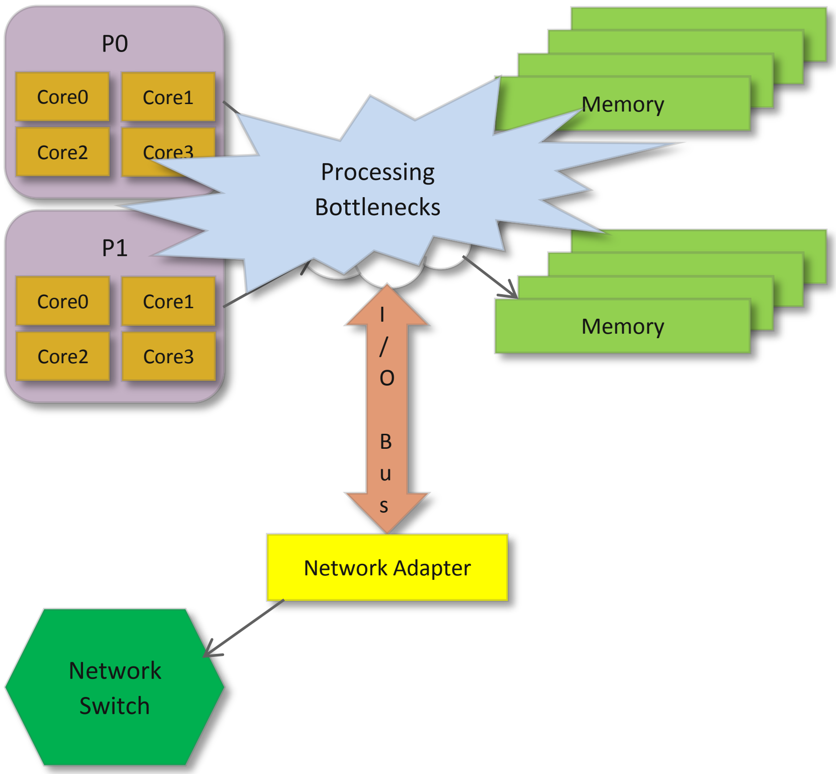
Integrated Networks (current/future systems)



- Several vendors are considering processor-integrated network adapters
- May improve network bandwidth
  - Unclear if the I/O bus would be a bottleneck
- Improves network latencies
  - Control messages between the processor, network, and memory are now on-chip
- Improves network functionality
  - Communication is a first-class citizen and better integrated with processor features
  - E.g., network atomic operations can be atomic with respect to processor atomics

Processing Bottlenecks in Traditional Protocols

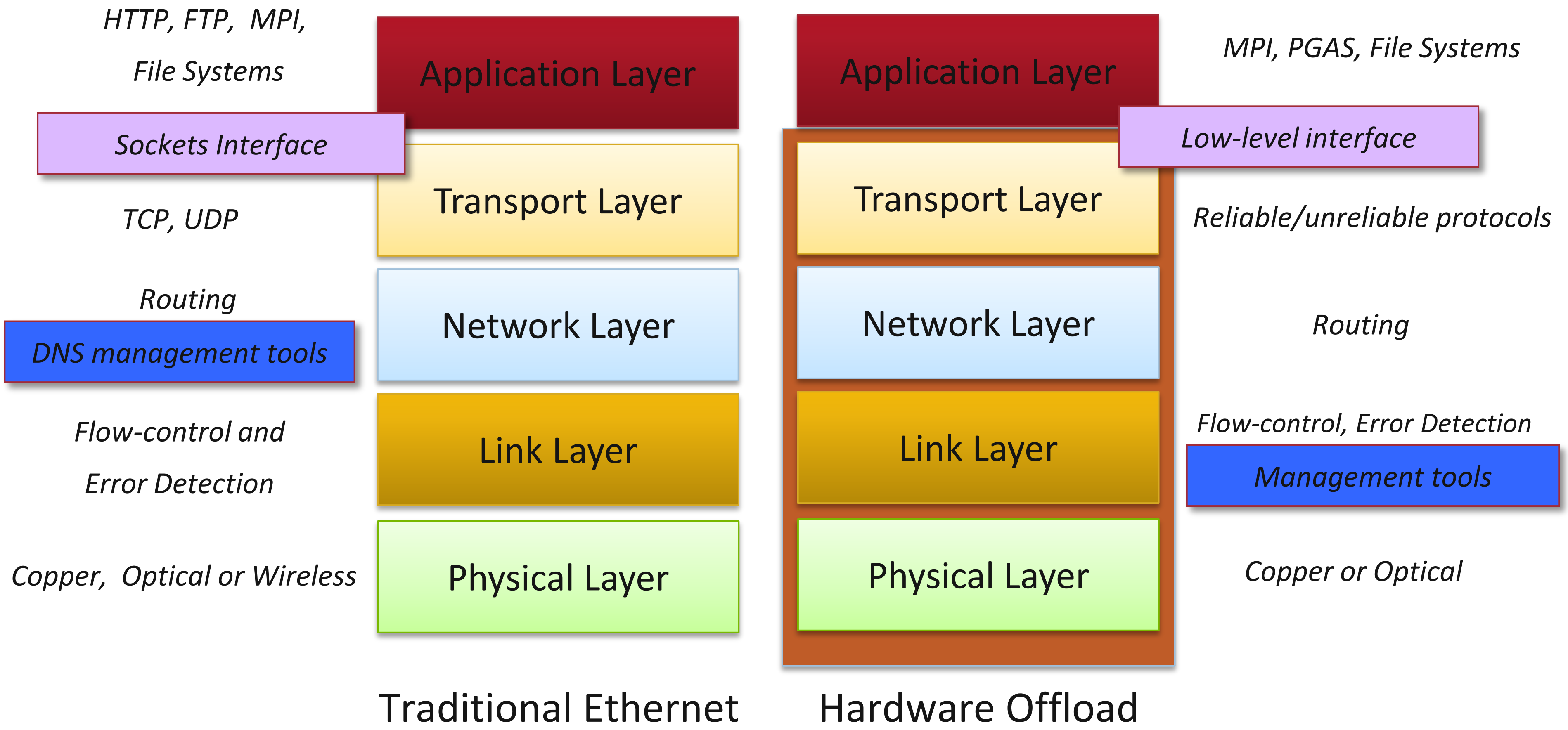
- Ex: TCP/IP, UDP/IP
- Generic architecture for all networks
- Host processor handles almost all aspects of communication
  - Data buffering (copies on sender and receiver)
  - Data integrity (checksum)
  - Routing aspects (IP routing)
- Signaling between different layers
  - Hardware interrupt on packet arrival or transmission
  - Software signals between different layers to handle protocol processing in different priority levels



Network Protocol Stacks: The Offload Era

- Modern networks are spending more and more network real-estate on offloading various communication features on hardware
- Network and transport layers are hardware offloaded for most modern networks
  - Reliability (retransmissions, CRC checks), packetization
  - OS-based memory registration, and user-level data transmission

Comparing Offloaded Network Stacks with Traditional Network Stacks

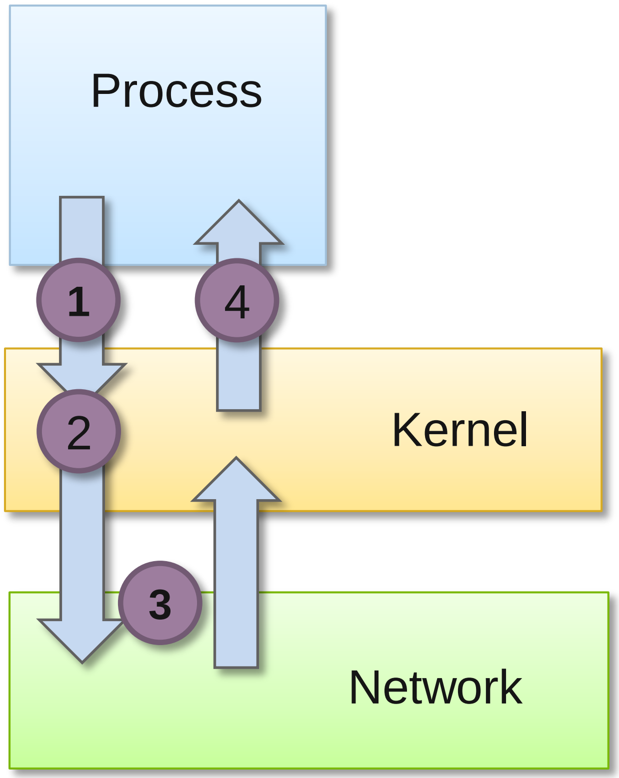


Current State for Network APIs

- A large number of network vendor specific APIs
  - InfiniBand verbs, Intel PSM2, IBM PAMI, Cray Gemini/DMAPP, …
- Recent efforts to standardize these low-level communication APIs
  - Open Fabrics Interface (OFI)
    - Effort from Intel, CISCO, etc., to provide a unified low-level communication layer that exposes features provided by each network
  - Unified Communication X (UCX)
    - Effort from Mellanox, IBM, ORNL, etc., to provide a unified low-level communication layer that allows for efficient MPI and PGAS communication
  - Portals 4
    - Effort from Sandia National Laboratory to provide a network hardware capability centric API

User-level Communication: Memory Registration

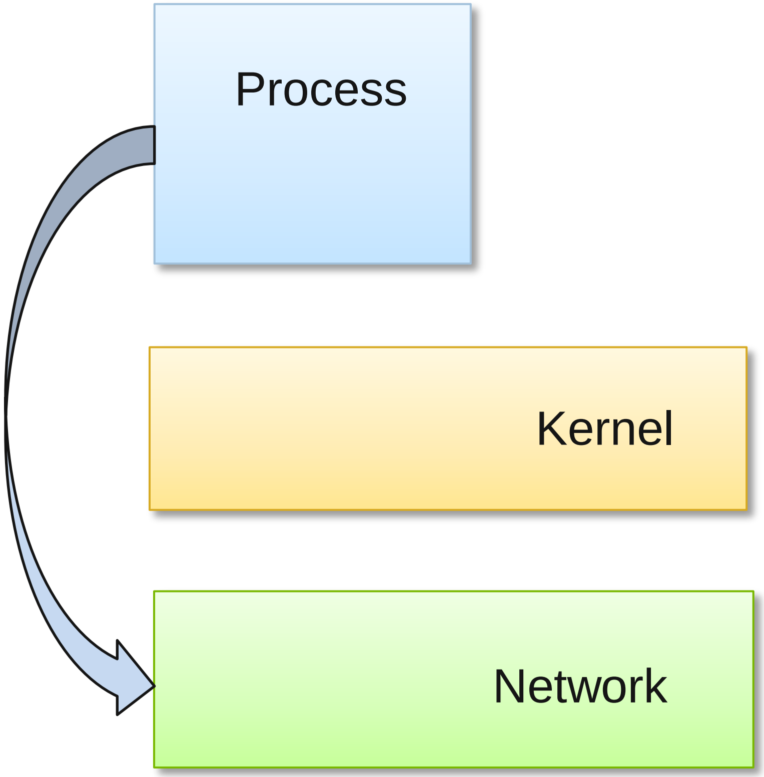
Before we do any communication:
All memory used for communication must be registered



1. Registration Request
   - Send virtual address and length
2. Kernel handles virtual->physical mapping and pins region into physical memory
   - Process cannot map memory that it does not own (security !)
3. Network adapter caches the virtual to physical mapping and issues a handle
4. Handle is returned to application

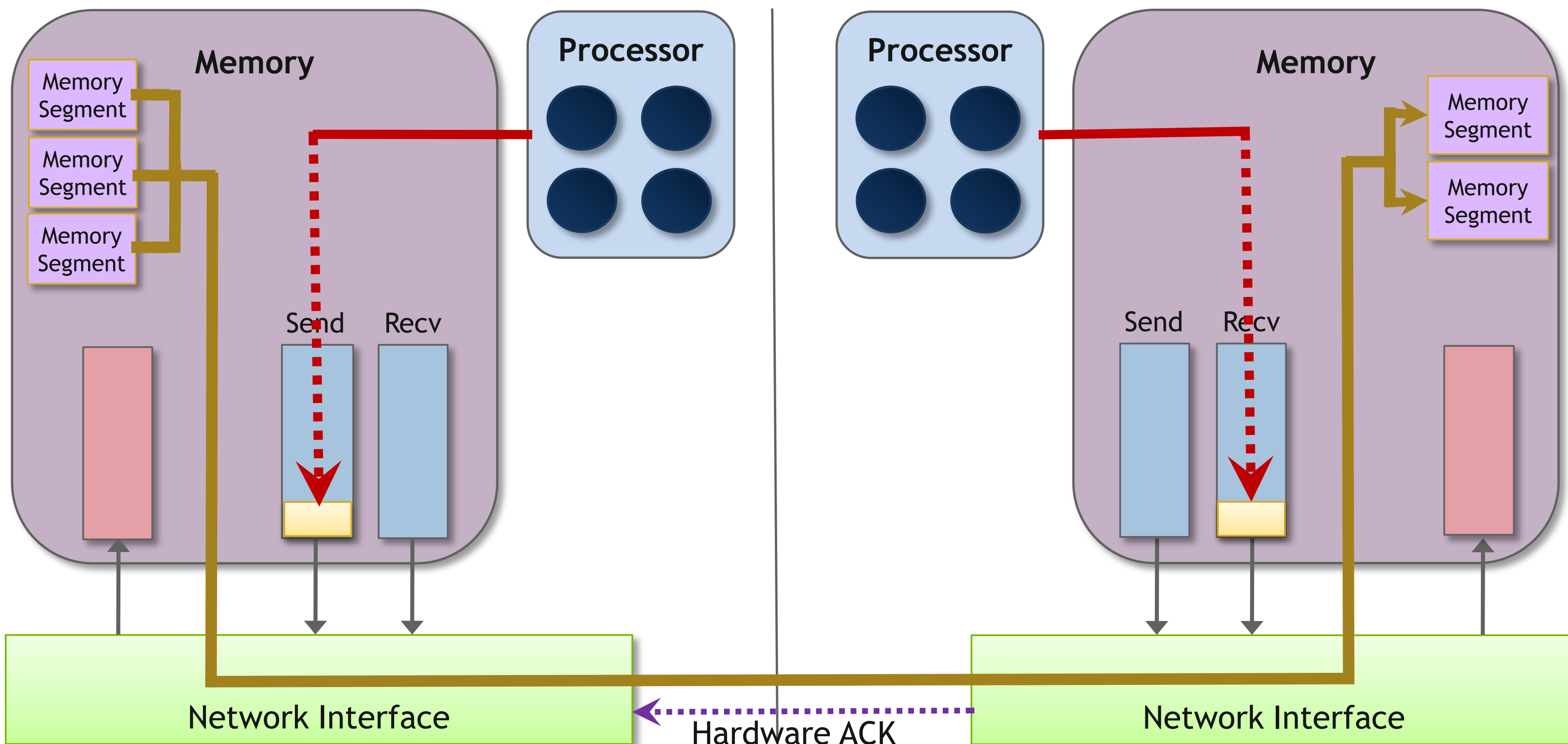
User-level Communication: OS Bypass

User-level APIs allow direct interaction with network adapters



- Contrast with traditional network APIs that trap down to the kernel
- Eliminates heavyweight context switch
- Memory registration caches allow for fast buffer re-use, further reducing dependence on the kernel

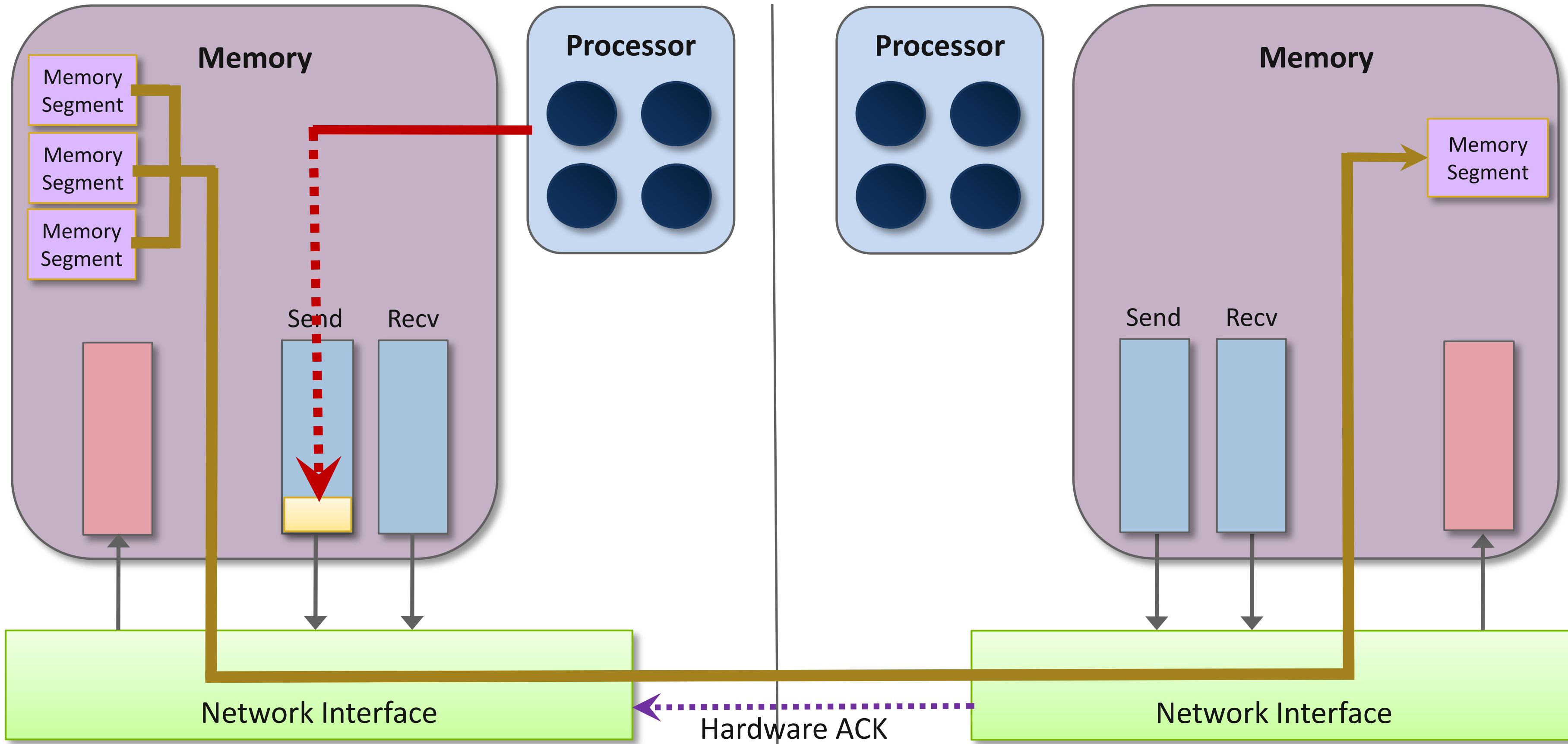
Send/Receive Communication



Send entry contains information about the send buffer (multiple non-contiguous segments)

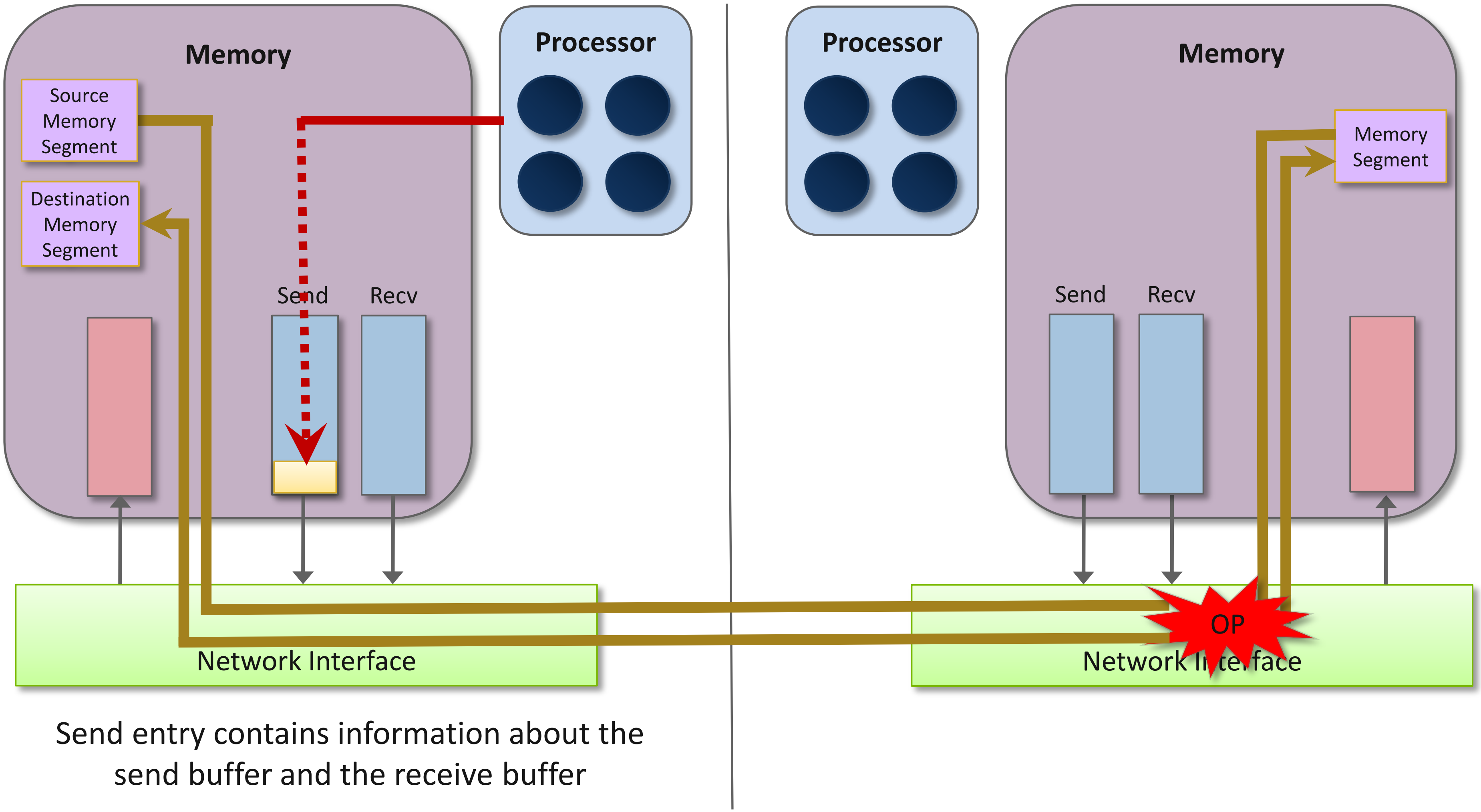
Receive entry contains information on the receive buffer (multiple non-contiguous segments); Incoming messages have to be matched to a receive entry to know where to place

PUT/GET Communication



Send entry contains information about the send buffer (multiple segments) and the receive buffer (single segment)

Atomic Operations



Send entry contains information about the send buffer and the receive buffer

Network Protocol Stacks: Specialization

- Increasing network specialization is the focus today
  - The next generation of networks plan to have further support for noncontiguous data movement, and multiple contexts for multithreaded architectures
- Some networks, such as the Blue Gene network, Cray network and InfiniBand, are also offloading some MPI and PGAS features on to hardware
  - E.g., PUT/GET communication has hardware support
  - Increasing number of atomic operations being offloaded to hardware
    - Compare-and-swap, fetch-and-add, swap
  - Collective operations (NIC and switch support)
  - Hardware tag matching for MPI send/recv
    - Cray Seastar, Bull BXI, Mellanox Infiniband (ConnectX-5 and later)

Agenda

Network Adapters

Network Topologies

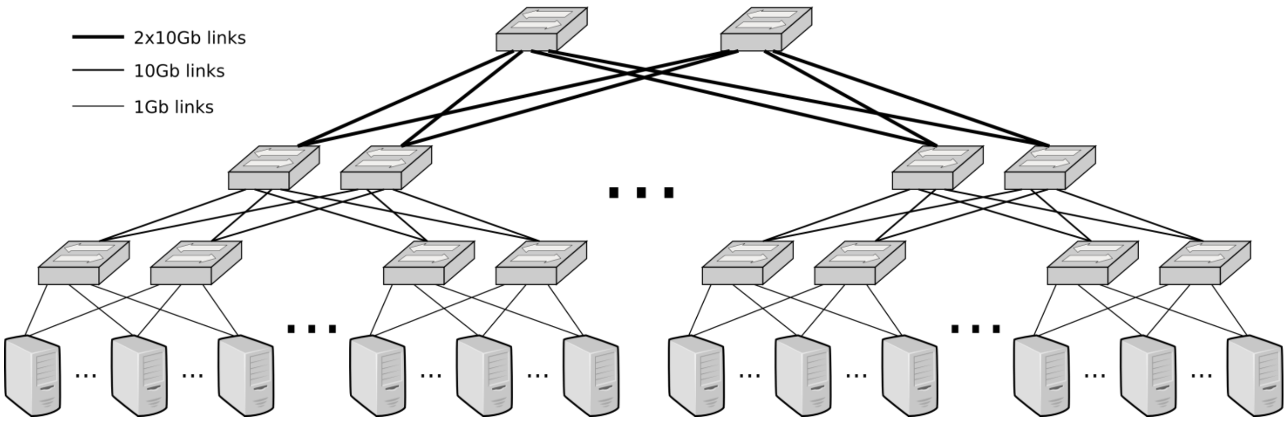
Network/Processor/Memory Interactions

Traditional Network Topologies: Crossbar

- A network topology describes how different network adapters and switches are interconnected with each other
- The ideal network topology (for performance) is a crossbar
  - Alltoall connection between network adapters
  - Typically done on a single network ASIC
  - Current network crossbar ASICs go up to 64 ports; too expensive to scale to higher port counts
  - All communication is nonblocking

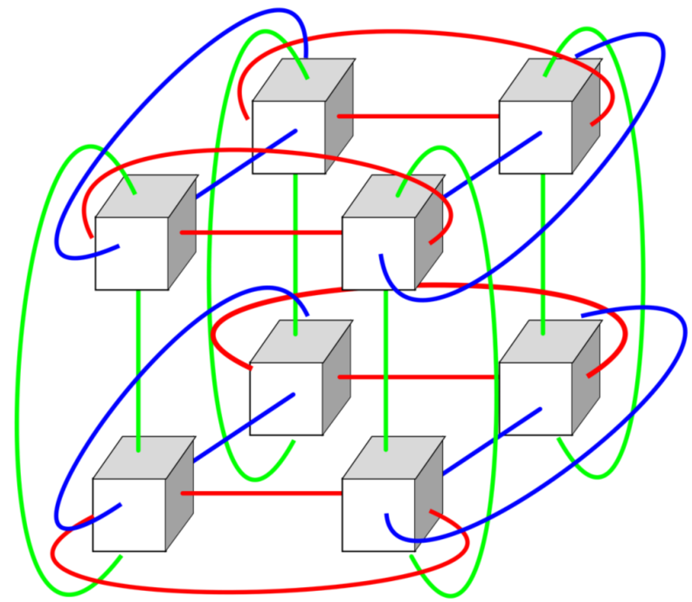
Traditional Network Topologies: Fat-tree

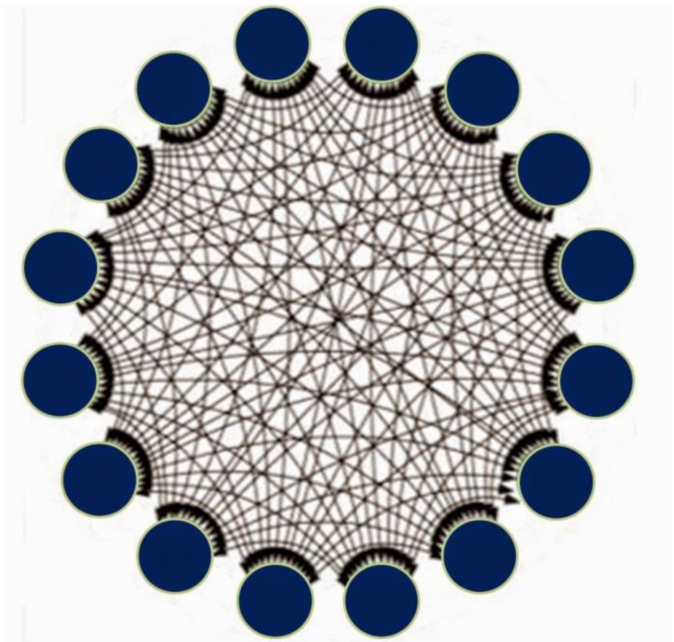
- The most common topology for small and medium scale systems is a fat-tree
  - Nonblocking fat-tree switches available in abundance
    - Allows for pseudo nonblocking communication
    - Between all pairs of processes, there exists a completely nonblocking path, but not all paths are nonblocking
  - More scalable than crossbars, but the number of network links still increases super-linearly with node count
    - Can get very expensive with scale



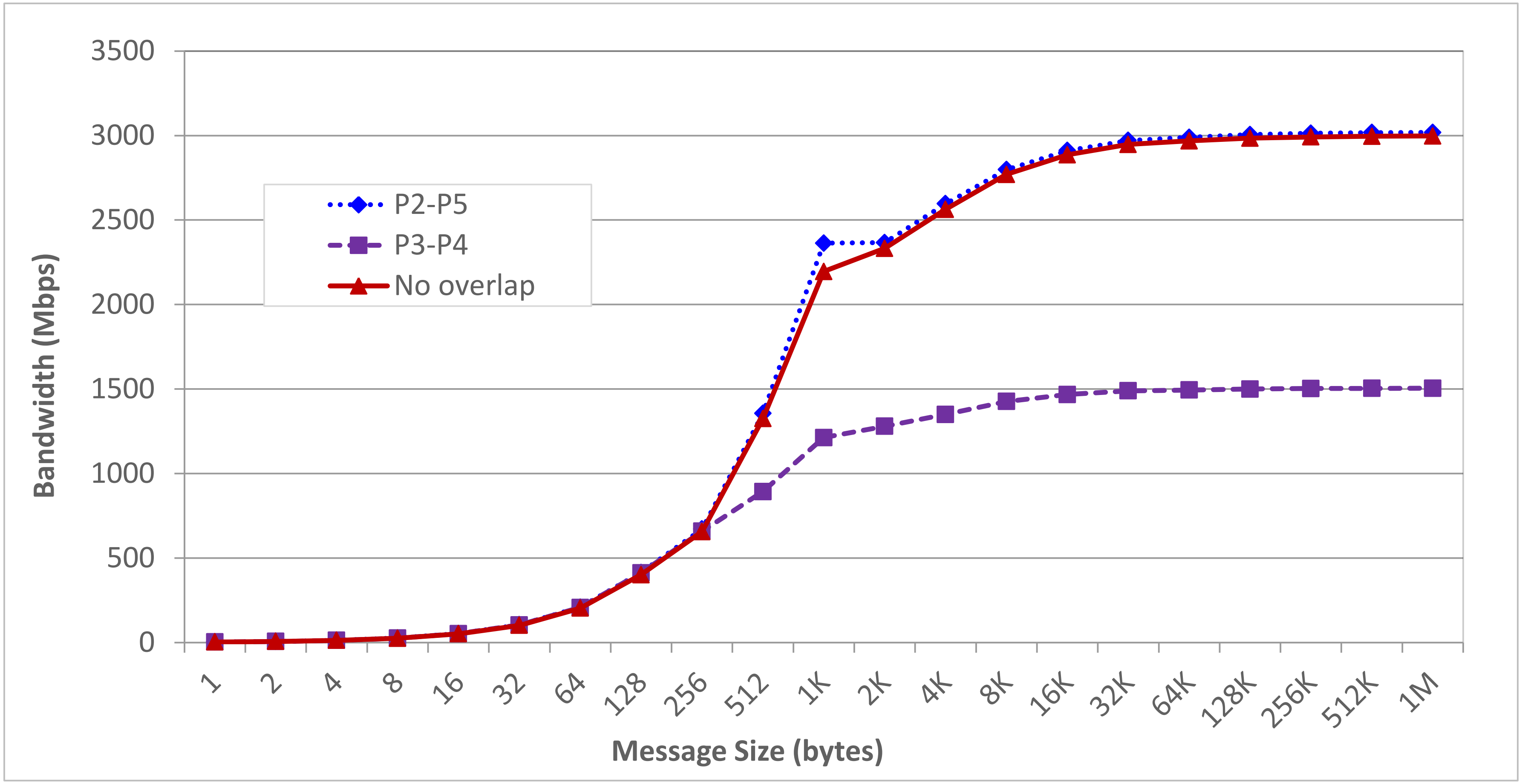
Network Topology Trends

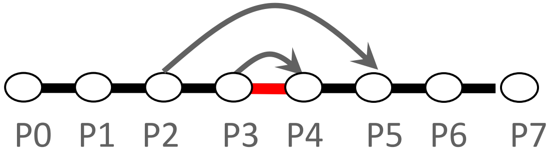
- Modern topologies are moving towards more "scalability" (with respect to cost, not performance)
- Blue Gene, Cray XE/XK, and K supercomputers use a torus-network; Cray XC uses dragonfly
  - Linear increase in the number of links/routers with system size
  - Any communication that is more than one hop away has a possibility of interference – congestion is not just possible, but common
  - Even when there is no congestion, such topologies increase the network diameter causing performance loss
- Take-away: topological locality is important and its not going to get better



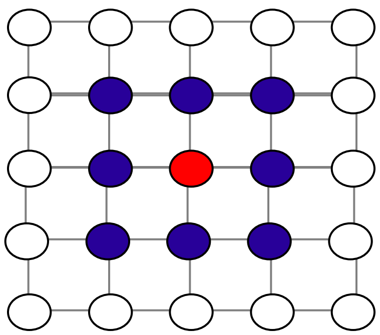


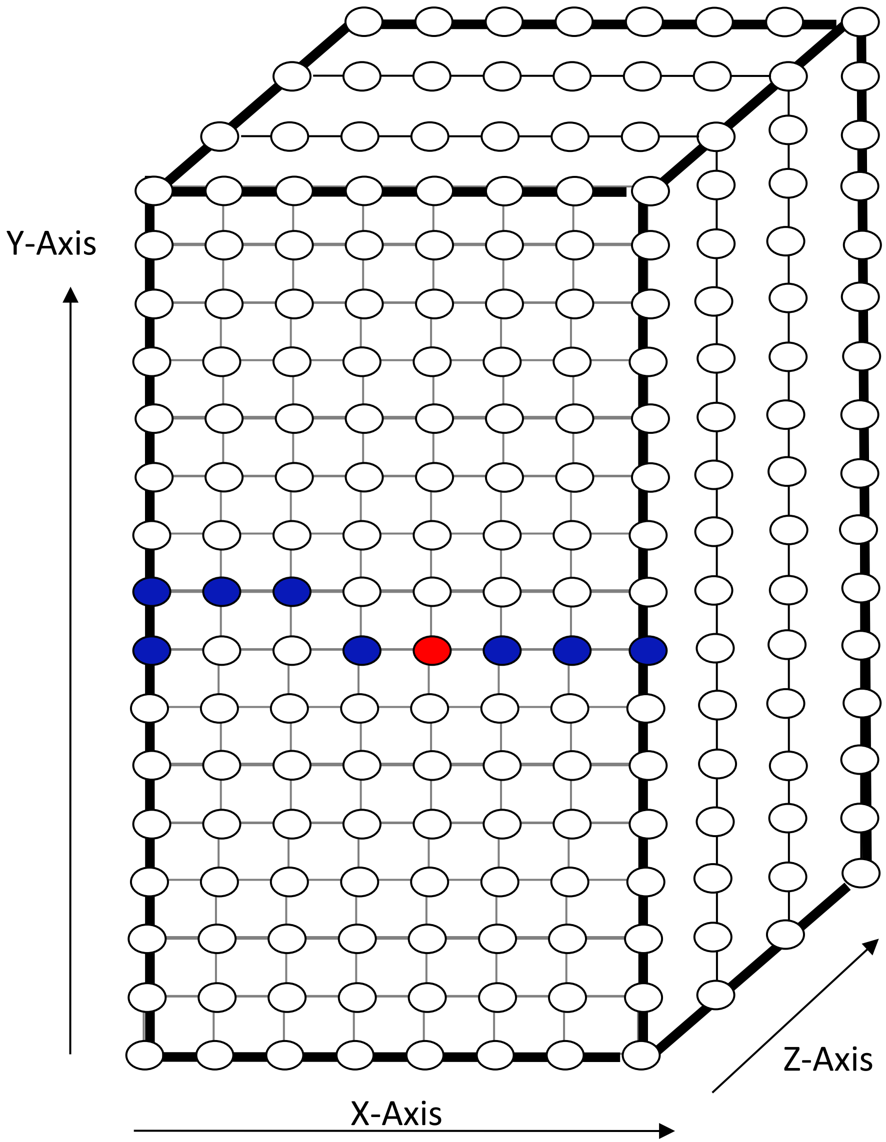
Network Congestion Behavior: IBM BG/P



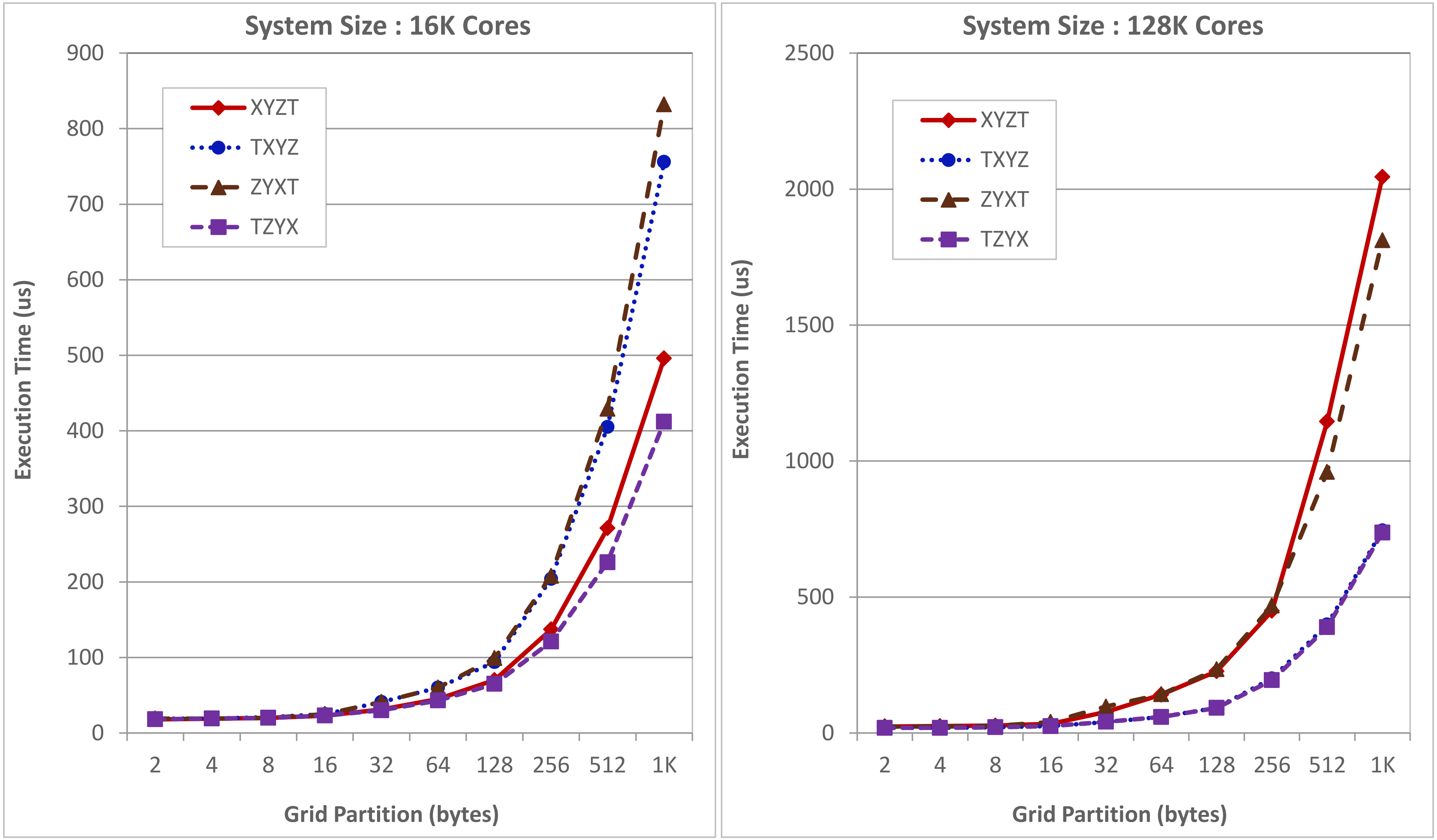


2D Nearest Neighbor: Process Mapping (XYZ)





Nearest Neighbor Performance: IBM BG/P



2D Halo Exchange

Agenda

Network Adapters

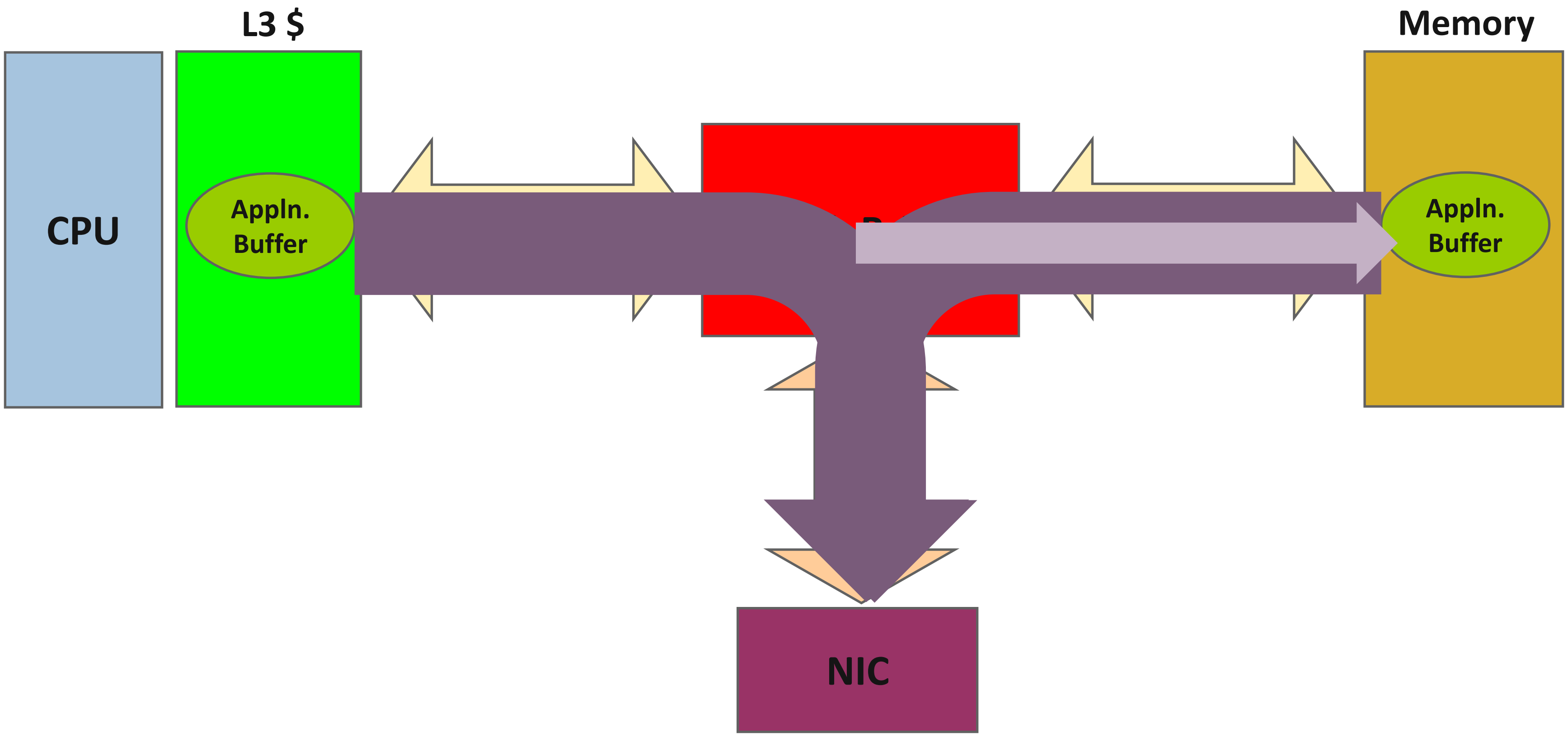
Network Topologies

Network/Processor/Memory Interactions

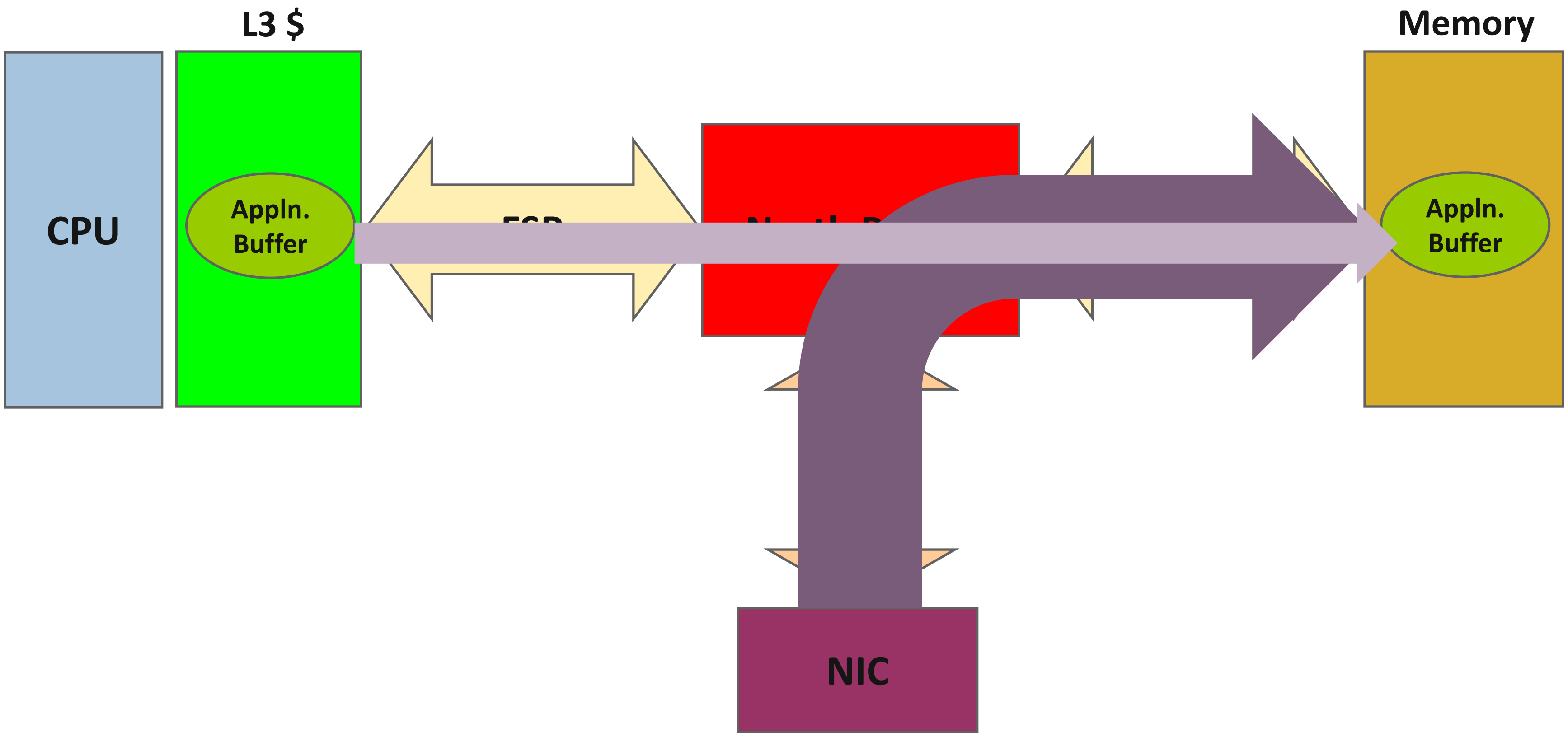
Network Interactions with Memory/Cache

- Most network interfaces understand and work with the cache coherence protocols available on modern systems
  - Users do not have to ensure that data is flushed from cache before communication
  - Network and memory controller hardware understand what state the data is in and communicate appropriately

Send-side Network Communication



Receive-side Network Communication



Network/Processor Interoperation Trends

- Direct cache injection
  - Most current networks inject data into memory
    - If data is in cache, they flush cache and then inject to memory
  - Some networks are investigating direct cache injection
    - Data can be injected directly into the last-level cache
    - Can be tricky since it can cause cache pollution if the incoming data is not used immediately
- Atomic operations
  - Current network atomic operations are only atomic with respect to other network operations and not with respect to processor atomics
    - E.g., network fetch-and-add and processor fetch-and-add might corrupt each other's data
  - With network/processor integration, this is expected to be fixed

Network Interactions with Accelerators

- PCI Express peer-to-peer capabilities enables network adapters to directly access third-party devices
  - Coordination between network adapter and accelerator (GPUs, FPGAs, …)
  - Data does not need to be copied into to/from buffers when going over the network
  - GPUDirect RDMA one example, but not limited to NVIDIA GPUs

Summary

- These are interesting times for all components in the overall system architecture: processor, memory, interconnect
  - And interesting times for computational science on these systems
- Interconnect technology is rapidly advancing
  - More hardware integration is the key to removing bottlenecks and improve functionality
    - Processor/memory/network integration is already in progress and will continue for the foreseeable future
  - Offload technologies continue to evolve as we move more functionality to the network hardware
  - Network topologies are becoming more “shared” (cost saving)

Thank You!

Email: raffenet@mcs.anl.gov