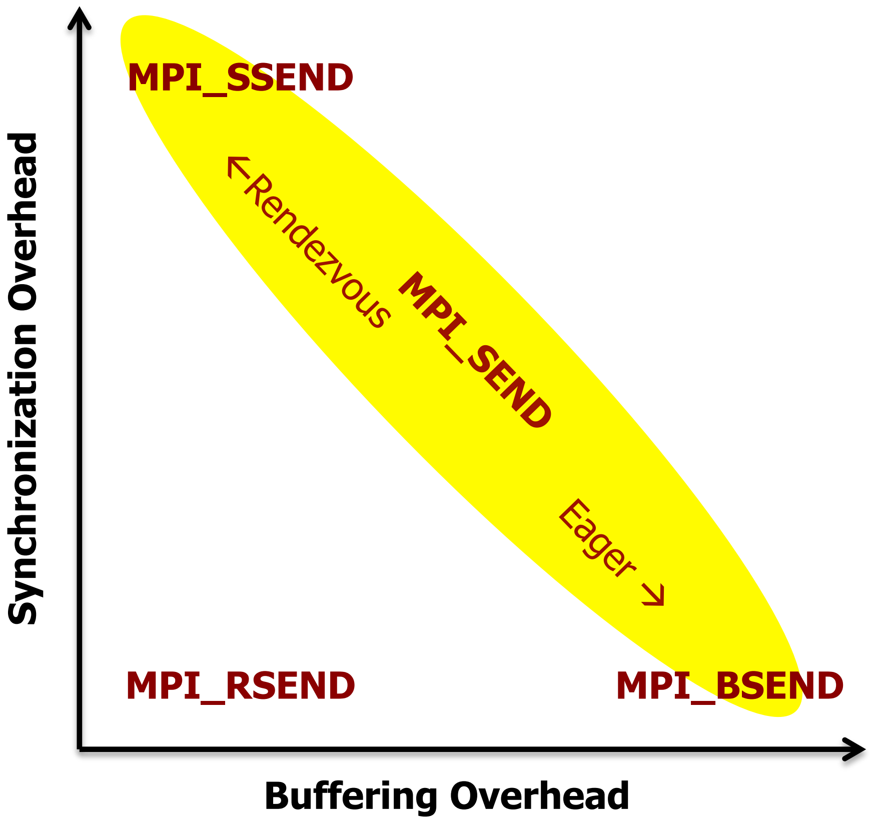
RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Introduction to High Performance Computing

Lecture 07 – Charateristics

Holger Fröning
Institut für Technische Informatik
Universität Heidelberg

Introduction



- Many ways to send a message
  - Goal: As few overhead as possible
  - Or: as much overlap as possible
- According to Amdahl
  - For scalability, communication must not contribute to the serial fraction
  - E.g., 10% of the execution time for communication, max. 10x SU
- Thus: **large overlap, low overhead** required for scalable parallel computing

Introduction

- So, what can we expect does happen if we perform non-blocking send/recv operations?
  - It all depends on the **network** respectively **network interface**…
  - Hard- and software components contribute to the overhead associated with message passing
  - Usually, SW is responsible to handle HW shortcomings
    - Examples: reliability, ordering, copy operations, …
- Goal of this lecture
  - The most important performance characteristics of message passing (resp. parallel computing)
  - Their importance for different applications

Latency and Bandwidth Characteristics

Analogy to Shared-Memory Computers

- Standard memory access is characterized using access latency and transfer bandwidth
  - About which of both do you worry more?
- Remember caching effects
  - Caches are used to reduce average access latencies!
  - Leveraging spatial and temporal locality
- Caching for message passing systems?
  - Spatial and temporal locality?

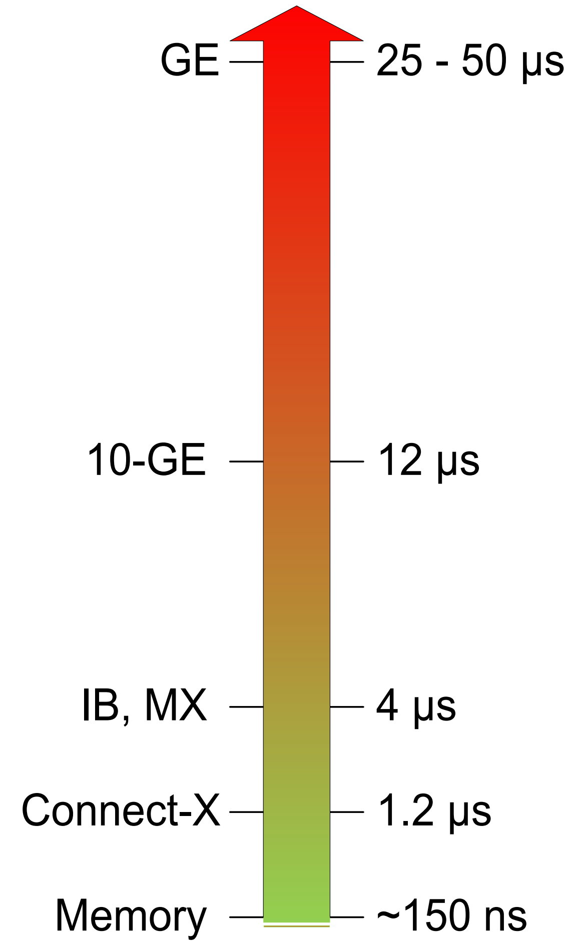
- What is most important for message passing systems?

Latency

- Latency is the time between starting a send and completing a receive
  - Typically in micro-seconds (usec)
  - Tends to vary widely across architectures
  - Software latency vs. hardware latency
    - Usually people care about the first one
  - Hardware latency about 10-100 times lower
- Diameter of a network
  - For a pair of two nodes with the highest distance between them, the diameter is the number of hops of the shortest path
- Latency is important for programs with many small messages

Latency

- Patterson stated: "*Latency lags Bandwidth" (CACM 2004)*
  - Bandwidth improves much more quickly than latency: memory, storage, networking
- Every component in between sender and receiver contributes to latency
  - Buffering, queues, pipeline stages, …
- Also, remember that the speed of light is limited
  - Vacuum: c_0=300m/usec or 3ns/m
  - PCB (FR4): typ. c = c_0/2.0 or about 6ns/m
  - Optical fiber: typ. c = c_0/1.5 or about 4.5ns/m

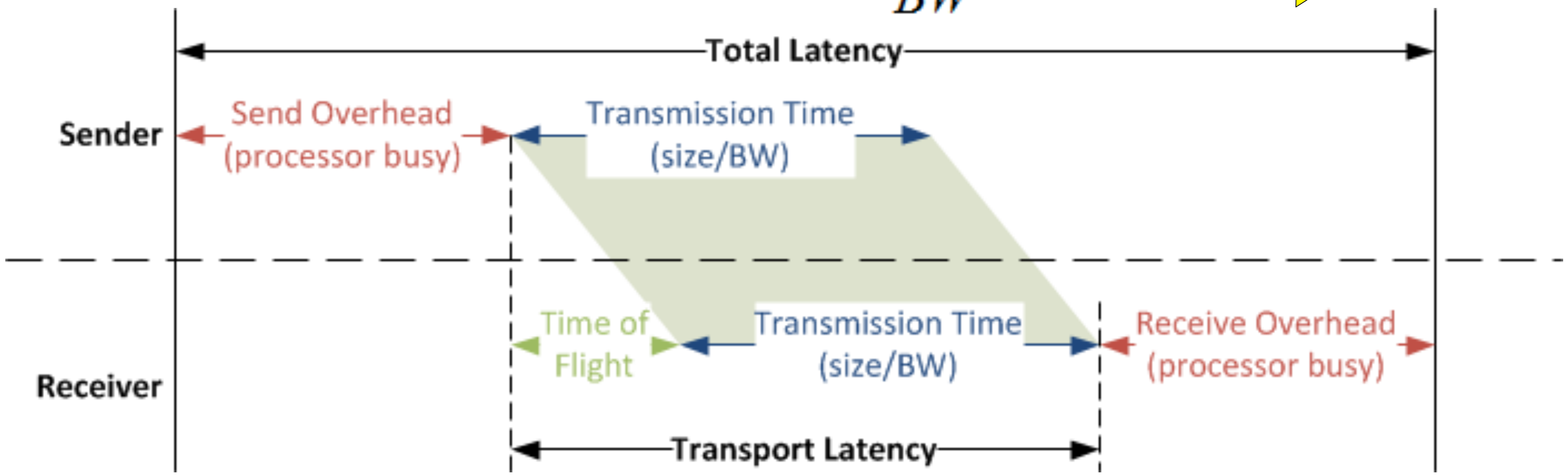


Latency

- Several components contribute to total latency
  - Simplified model here
- Start-up latency: latency of a minimum sized message
  - Zero-sized message
  - Good indicator for overhead

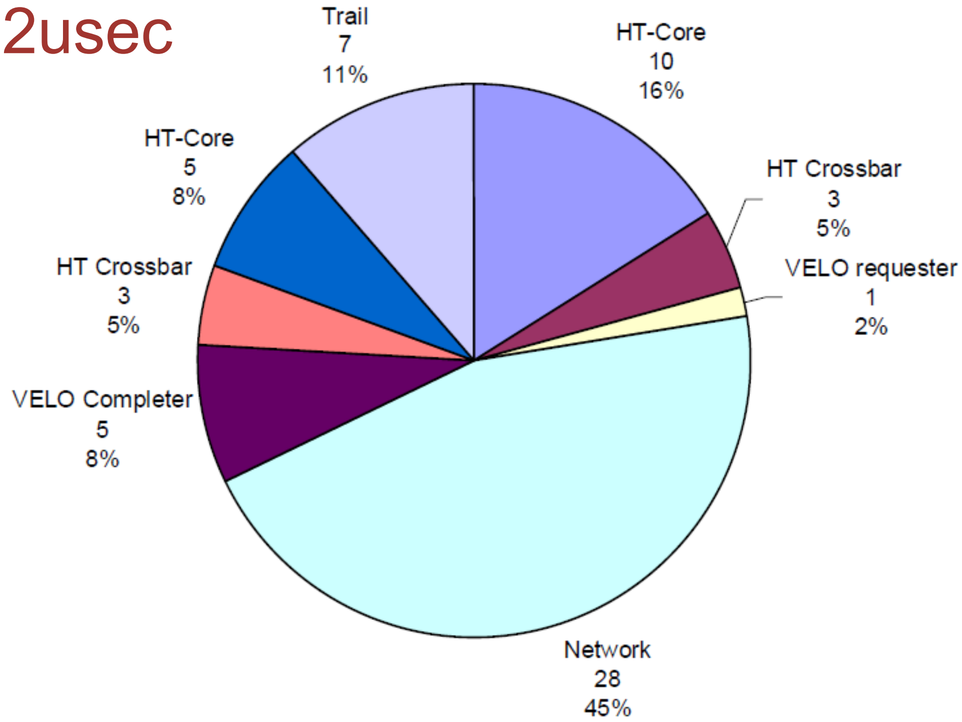
When is the processor free for other tasks?

$$t_{lat} = t_{send} + t_{flight} + \frac{size}{BW} + t_{recv}$$



Latency

- VELO Example
- 62 cycles at 100MHz: 0.62usec
- API latency: 0.97usec
  - Diff = CPU/MC/SW overhead
  - ~0.35usec
- MPI adds 0.2-0.5 usec
  - MTL resp. BTL



Heiner Litz, Holger Fröning, Mondrian Nüssle, Ulrich Brüning, VELO: A Novel Communication Engine for Ultra-low Latency Message Transfers, *37th International Conference on Parallel Processing (ICPP-08)*, Sept. 08 - 12, 2008, Portland, Oregon, USA.

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

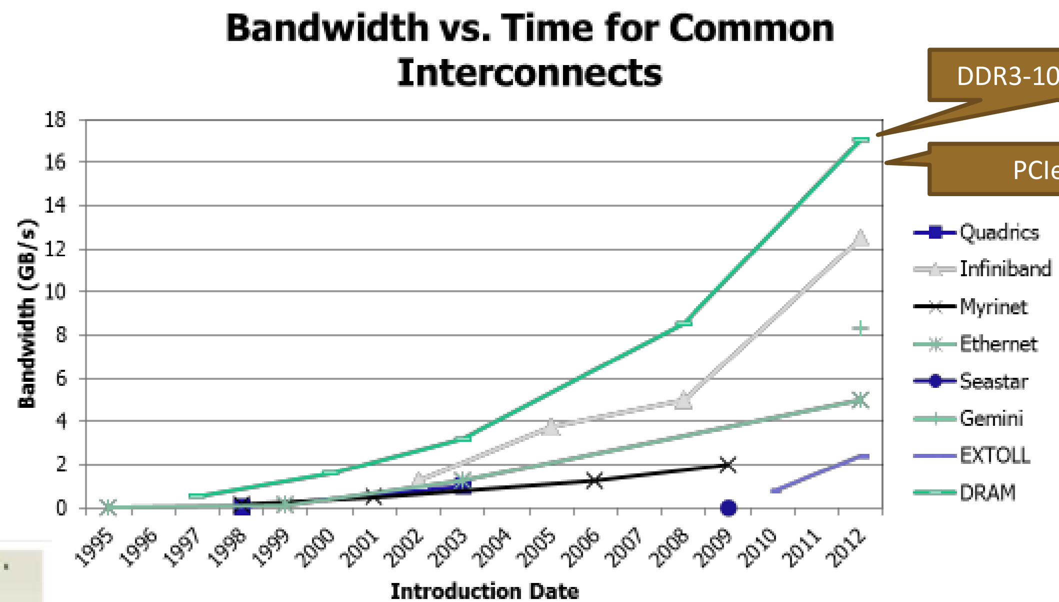
Bandwidth

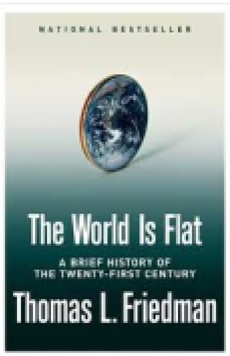
- Physical or peak bandwidth BW = data width / cycle time
  - Cycle time = 1 / frequency
  - Applies for a network link, internal data path, PCIe subsystem, …
- Unidirectional vs. bidirectional bandwidth
  - Some old topologies (buses) did not allow bidirectional transfers
  - Not of importance today (except marketing)
- Effective or sustained bandwidth typically lower
  - Protocol overhead
- Bandwidth is maybe the most important characteristic today (Big Data era)
  - Processor/Memory gap ➔ Processor/Network gap

Bandwidth – Current Trends

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Future – A Flat World?





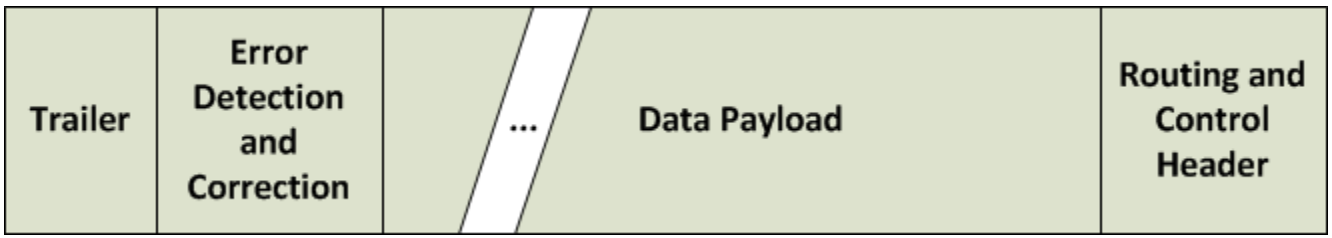
Similar applies for latency!

„Everyone is getting closer and we need better sharing […]“
Sudha Yalamanchili, UCAA Workshop 2012

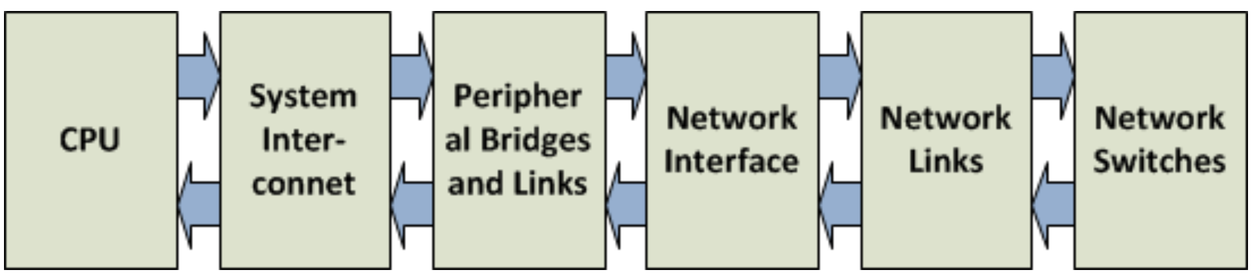
H. Fröning, Data Movement Options for Accelerated Clusters, OMHI Workshop 2013 5

Bandwidth

- Protocol overhead
  - Payload size / Packet size ratio



- Overall bandwidth limited by individual subsystem bandwidth



Latency Bandwidth Analysis

- Intel MPI Benchmarks (IMB)
  - Easy to use, free benchmark suite
- Benchmarks a large set of MPI functions
  - Point-to-point message passing
    - Single transfer & parallel transfer
  - Global data movement and computation routines
    - Collective
  - One-sided communications & File I/O
- **PingPong & PingPing**
  - Start-up latency & peak bandwidth
  - No difference for good MPI implementations

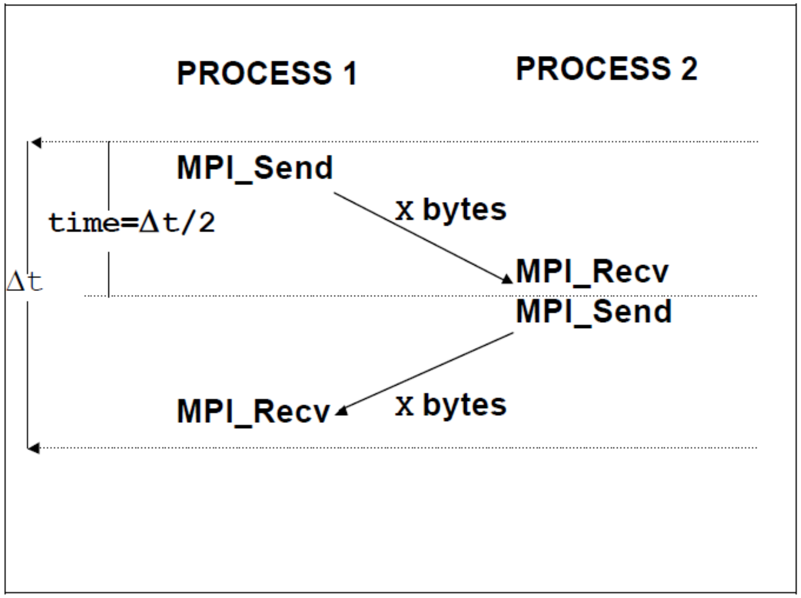


Figure 1: `PingPong` pattern

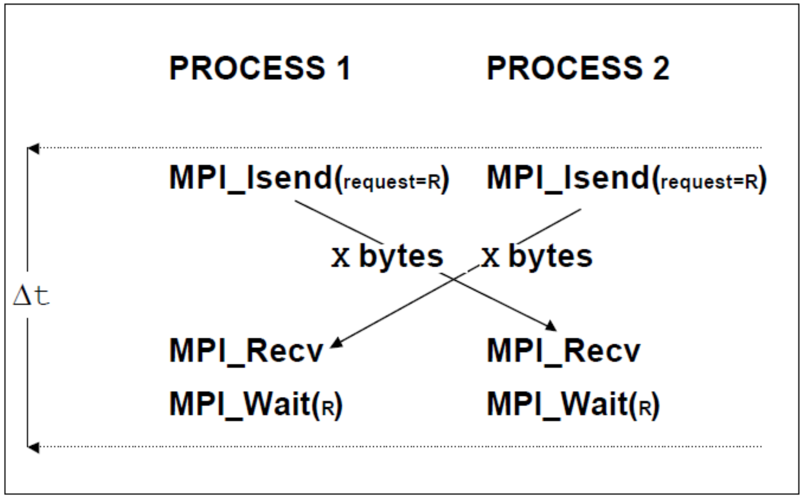
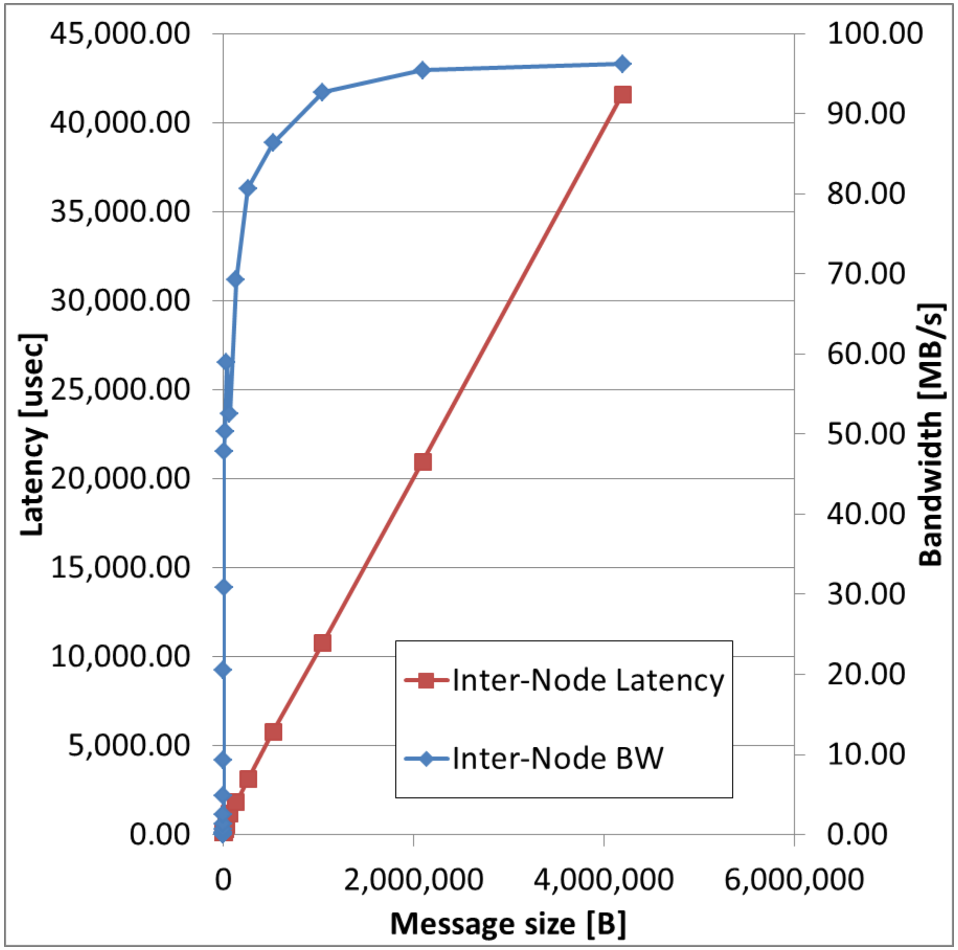


Figure 2: `PingPing` pattern

Source: http://software.intel.com/en-us/articles/intel-mpi-benchmarks

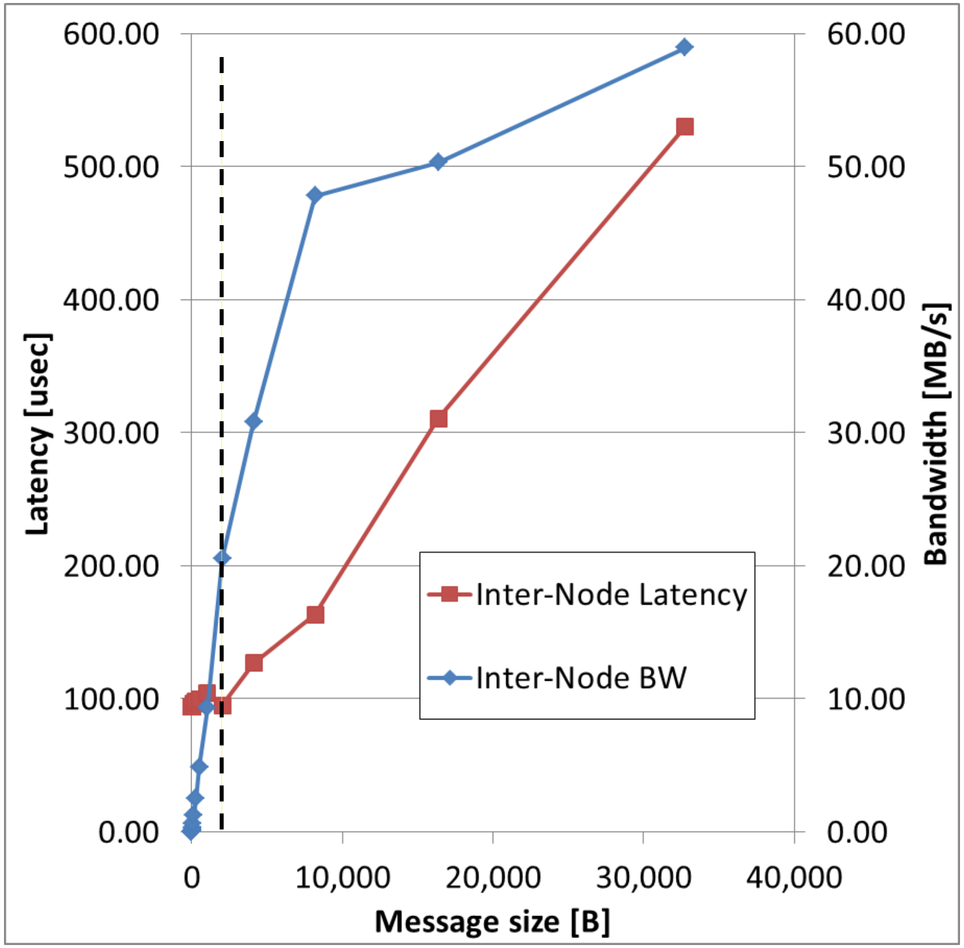
Example: IMB & Gigabit Ethernet

- Pingpong test
- Quiescent system
  - Contention and congestion will dramatically affect performance
- Mapping
  - Inter- vs. intra-node effects
- Observations - BW
  - Low compared to memory BW
  - Reaches saturation asymptotically
- Observations - Latency
  - Linear scaling for large messages



Example: IMB & Gigabit Ethernet

- Pingpong test - zoomed view
- Observations - BW
  - Really low BW for small messages
  - Saturation reached slowly
  - Overheads!
    - Protocol, send, receive
- Observations - Latency
  - For small messages (<2kB) constant
  - Otherwise linear scaling



Latency Bandwidth Analysis

- Intel MPI Benchmarks (IMB) – SendRecv
  - Periodic communication chain, send and receive can overlap
  - Reports 2x peak BW (1 in, 1 out)
  - Double throughput for perfectly bidirectional systems
- Exchange
  - Reports 4x peak BW (2 in, 2 out)

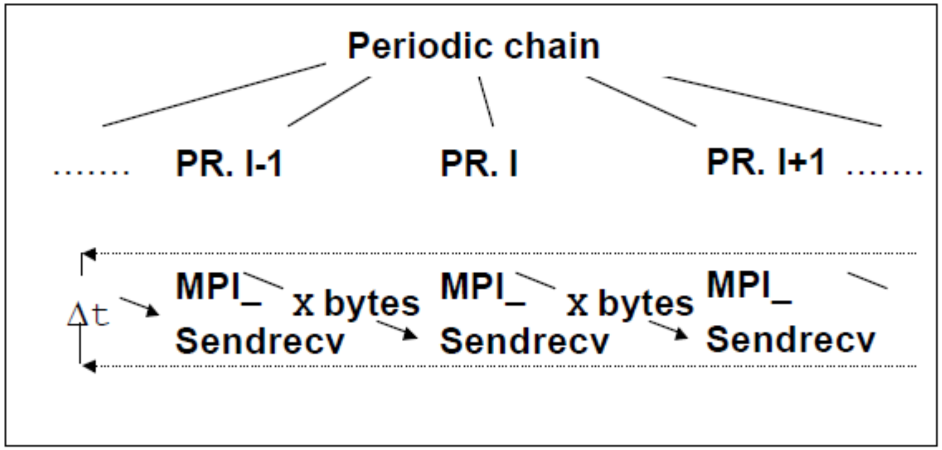


Figure 3: Sendrecv pattern

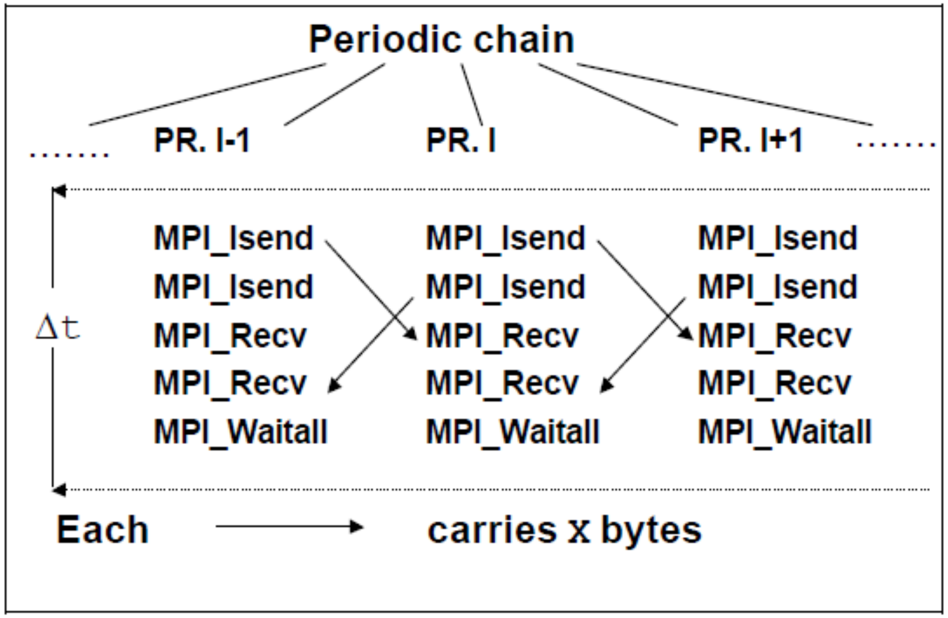


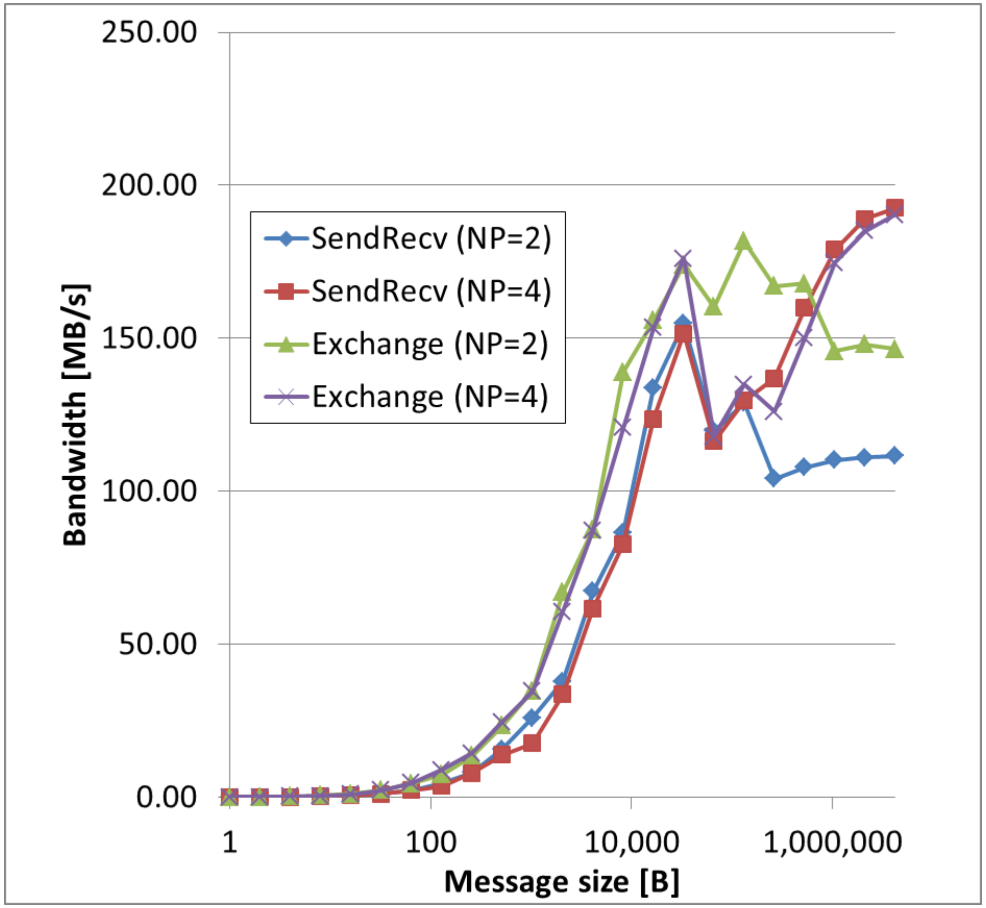
Figure 4: Exchange pattern

Source: http://software.intel.com/en-us/articles/intel-mpi-benchmarks

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Example: IMB & Gigabit Ethernet

- Parallel transfer tests
  - One process per node
- Insights
  - Small messages (<1kB): exchange better
  - Large messages (>64kB): contention for exchange
  - Only NP=2 experiments are able to get close to saturation
  - 4P@64kB: performance downgrades
- Reported BW much lower than expected



RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Overhead and Availability Characteristics

Overhead and Overlap

- **Overhead** is defined as the length of time that a processor is engaged in the transmission or reception of each message; during this time, the processor cannot perform other operations.
  - [Culler et. al, LogP: Towards a Realistic Model of Parallel Computation, *PPoPP*, 1993]
- **Application availability** is defined to be the fraction of total transfer time that the application is free to perform non-MPI related work
  - [Lawry et. al, COMB: A Portable Benchmark Suite for Assessing MPI Overlap, *CLUSTER*, 2002]

$$Availability[\%] = 1 - (overhead[usec] / transfer\_time[usec])$$

Overhead and Overlap

- Measuring overhead: basic idea is a post-work-wait loop
  - For each message size, repeat following steps with increasing work_t:

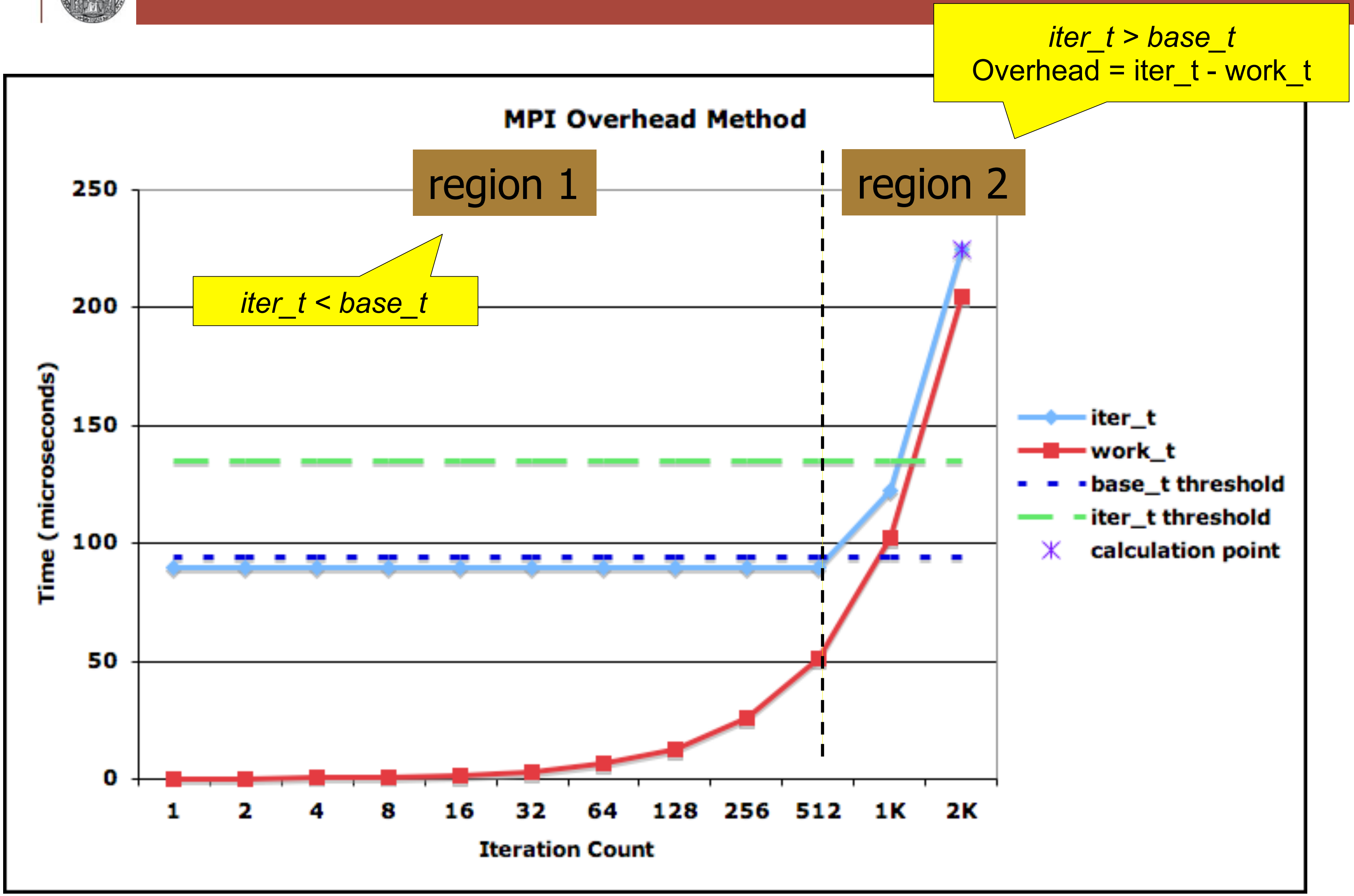
1. MPI_Isend
2. Work loop — *work_t*
3. MPI_Wait

(steps 1–3: *iter_t*)

- Three steps:

1. Work completes before message transfer is complete (region 1)
   - Derive *base_t* based on first *iter_t* measurement
     - E.g., *base_t = iter_t*1.05*
   - *iter_t < base_t*
   - Message transfer time equals loop time *iter_t*
2. Work time exceeds message transfer time (region 2)
   - *iter_t > base_t*
   - Overhead = iter_t – work_t
3. Stop if iter_t > threshold

Overhead and Overlap



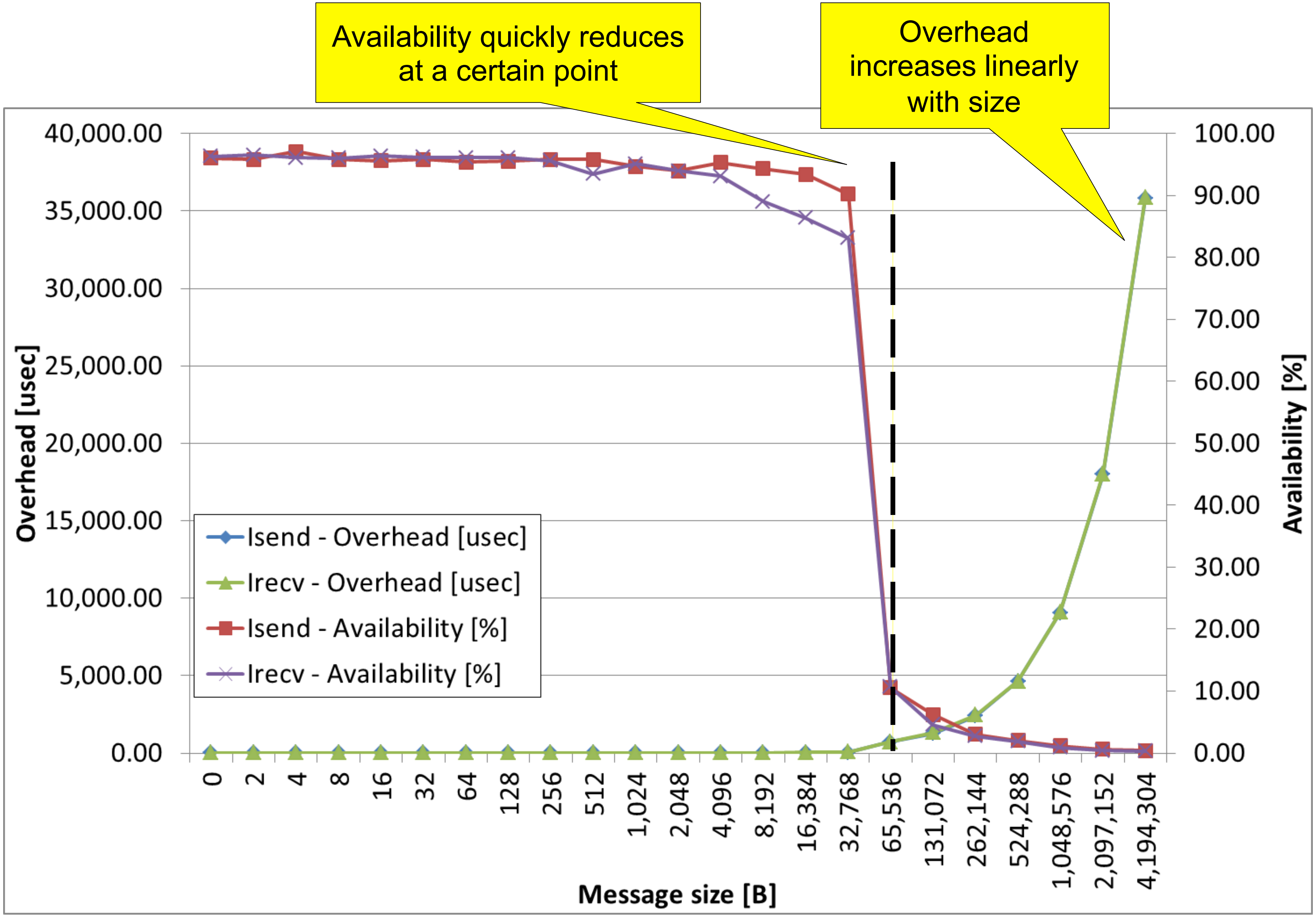
Source: http://www.cs.sandia.gov/smb/overhead.html

Overhead and Overlap Analysis

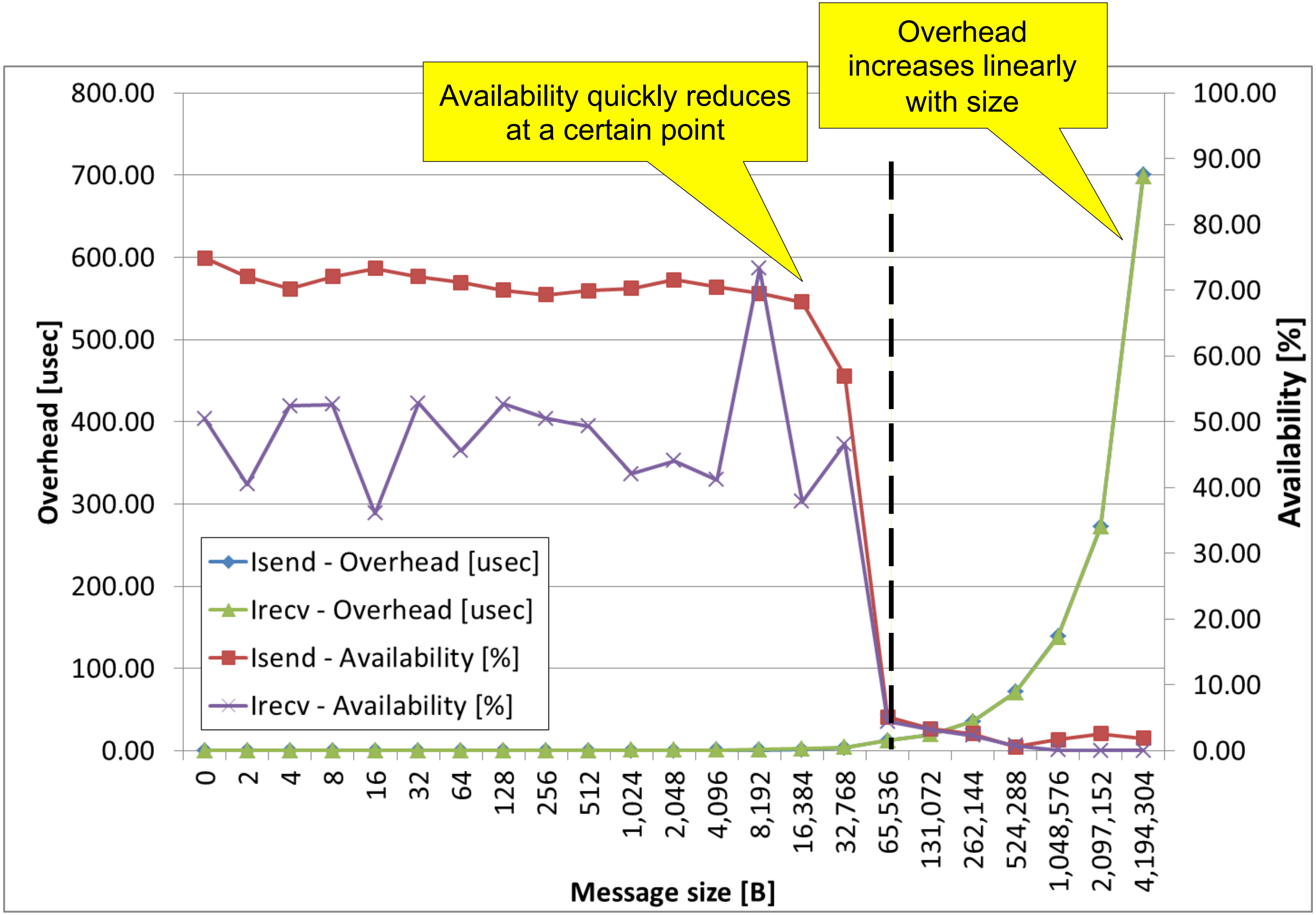
- Sandia MPI Micro-Benchmark Suite (SMB)
  - Free benchmark suite

1. Host Processor Overhead
   - Measures host processor overhead and availability during non-blocking MPI operations
2. Real World Message Rate Benchmark
   - Measures sustained message throughput at scale with multiple peers, as would be expected in real Sandia application scenarios

Example: SMB & Gigabit Ethernet

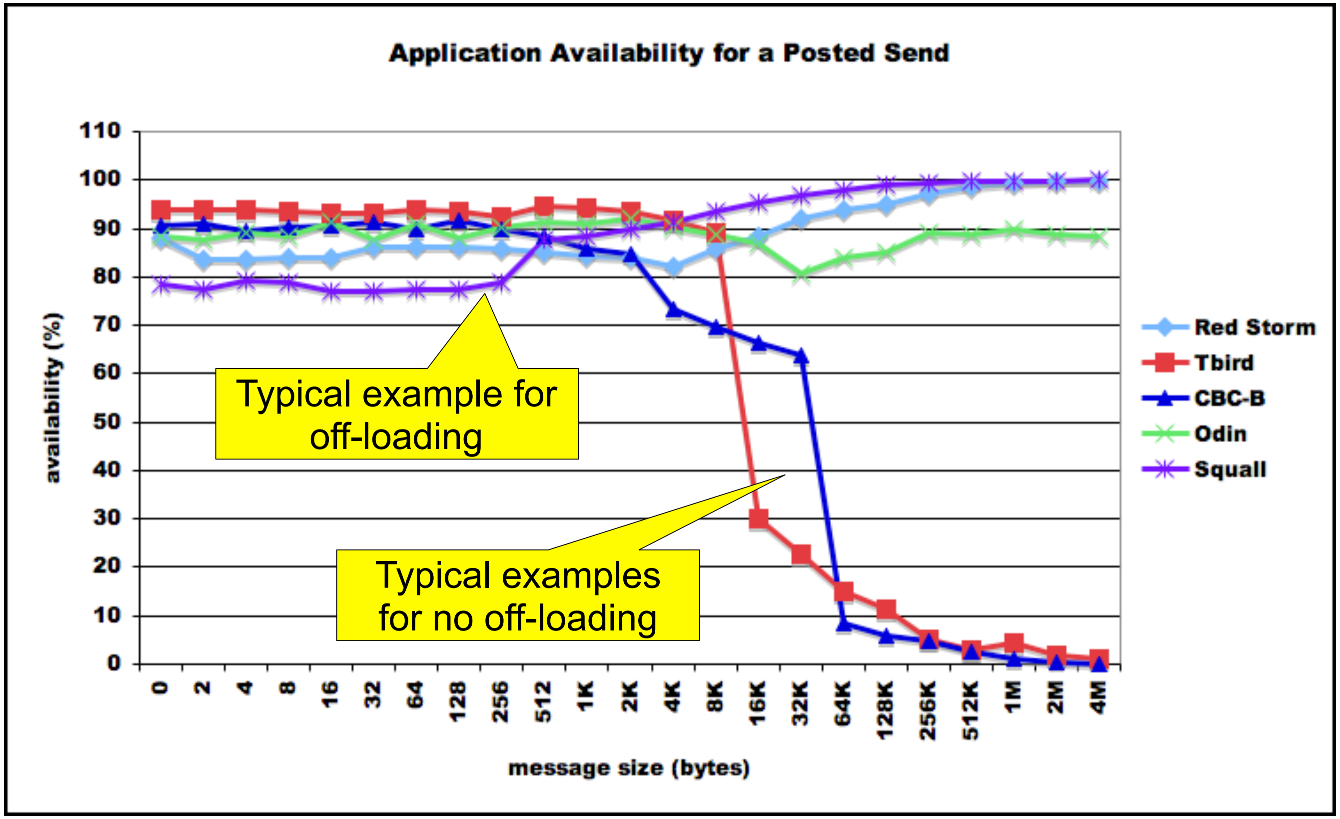


Example: SMB & Infiniband



Example: SMB - various

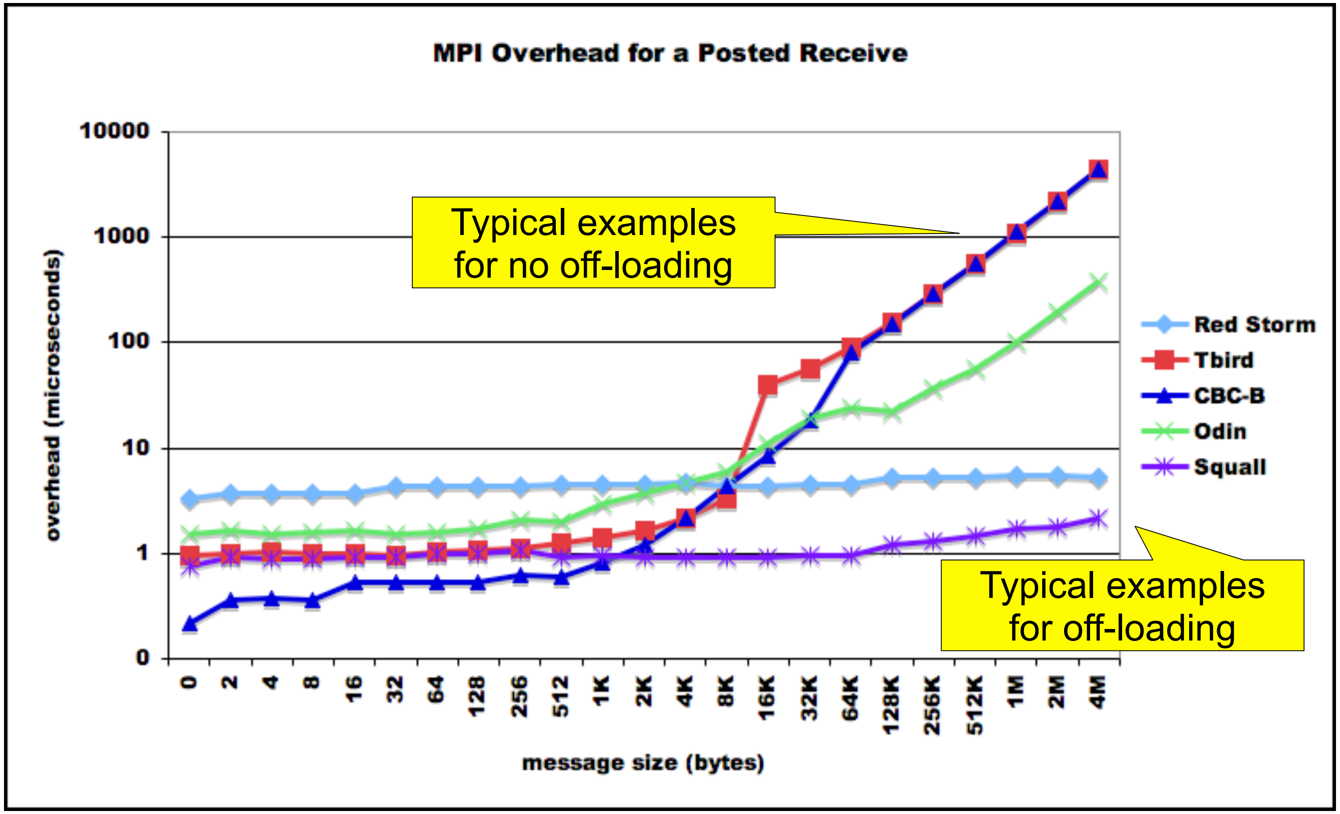
	Red Storm	Thunderbird	CBC-B	Odin	Red Squall
Interconnect	Seastar 1.2	InfiniBand	InfiniBand	Myrinet 10G	QsNetII
Adapter	Custom	PCI-Express HCA	InfiniPath	Myri-10G	Elan4
Host Interface	HT 1.0	PCI-Express	HT 1.0	PCI-Express	PCI-X



Source: http://www.cs.sandia.gov/smb/overhead.html

Example: SMB - various

	Red Storm	Thunderbird	CBC-B	Odin	Red Squall
Interconnect	Seastar 1.2	InfiniBand	InfiniBand	Myrinet 10G	QsNetII
Adapter	Custom	PCI-Express HCA	InfiniPath	Myri-10G	Elan4
Host Interface	HT 1.0	PCI-Express	HT 1.0	PCI-Express	PCI-X

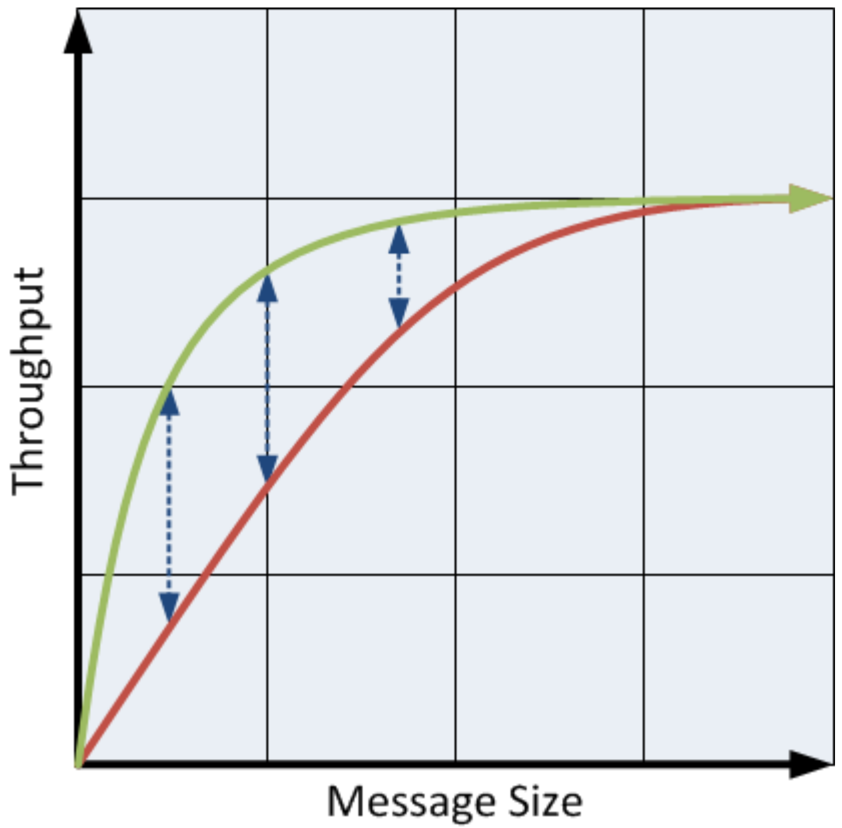


Source: http://www.cs.sandia.gov/smb/overhead.html

Message Rate Characteristics

Message Rate

- Up to now: latency, bandwidth, overhead
  - Basically sufficient for characterization
- However, small messages:
  - Latency cannot characterize overlap in this case
  - Bandwidth typically reported for large messages
- More important:
  - How many messages per second one can send out?
  - Push-model instead of round-trip communication



➔ Message Rate (MR), in messages per second

Message Rate

- Theoretical upper bound: BW/size
  - Sustained bandwidth for given message size
  - I.e., peak bandwidth without protocol overhead for given message size
- Practical upper bound: gaps
  1. Network protocol overhead including framing, headers, CRC, etc
  2. Message passing protocol overhead including tags, source identification, etc
  3. Packet-to-packet gaps caused by network interface
  4. Packet-to-packet gaps caused by switching units
  5. Software overhead for sending and receiving

Message Rate

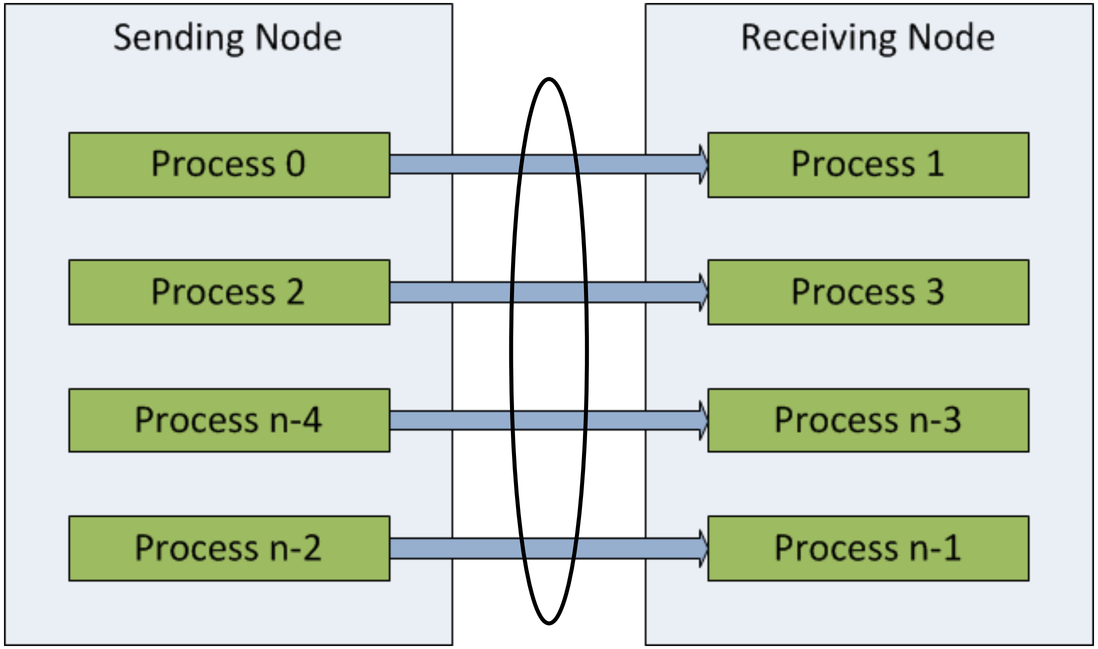
- Various overhead sources
- Multi-pair tests help overcoming software overhead limitations
  - Multiple end points per node
- Thus:
  - Latency & bandwidth should not (best case) be affected by multi-pair tests
  - MPI message rate typically benefits a lot from multiple end points per node

Network	10GE[1]	IB-QDR[2]	EXTOLL[3]
Net Speed	10 Gbps	32 Gbps	5 Gpbs
Theoretical peak message rate (8B payload)	156.3	500.0	78.0
Network protocol overhead	82 B	38 B	32 B
MPI protocol overhead	24 B	10 B	16 B
Packet-to-Packet gap of switching units	NA	NA	8 B
Packet-to-Packet gap of network interface	NA	NA	0 B
Overhead total (as appropriate)	114 B (w/o gaps)	56 B (w/o gaps)	64 B (total)
Sustained Message Rate	0.66 (0.42%)	6.67 (1.33%)	9.73 (12.4%)
Calculated overhead derived from sustained MR	416.67 B	599.70 B	64.14 B

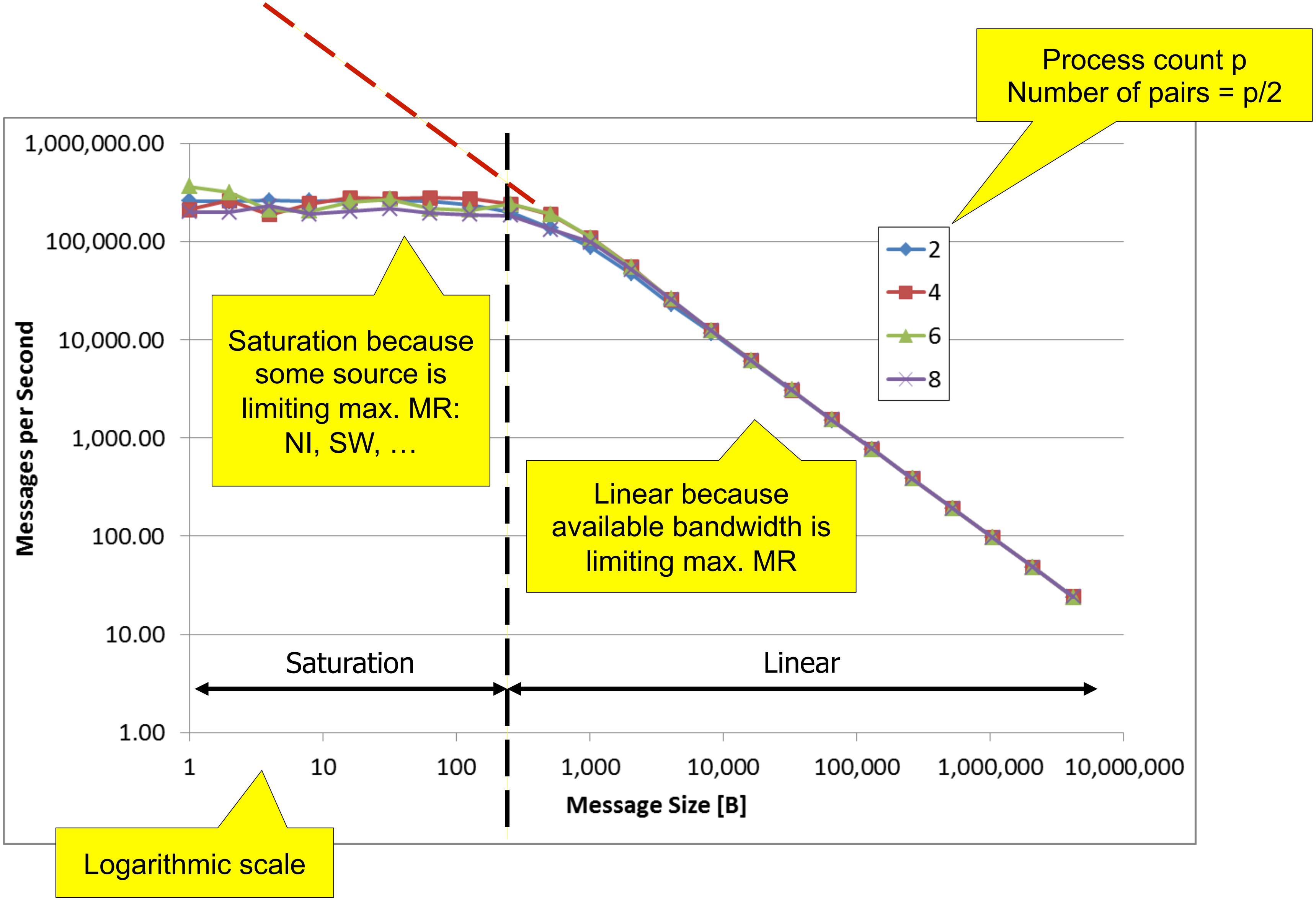
Holger Fröning, Mondrian Nüssle, Heiner Litz, Christian Leber and Ulrich Brüning, On Achieving High Message Rates, 13th IEEE/ ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), May 13-16, 2013, Delft, The Netherlands.

Message Rate Analysis

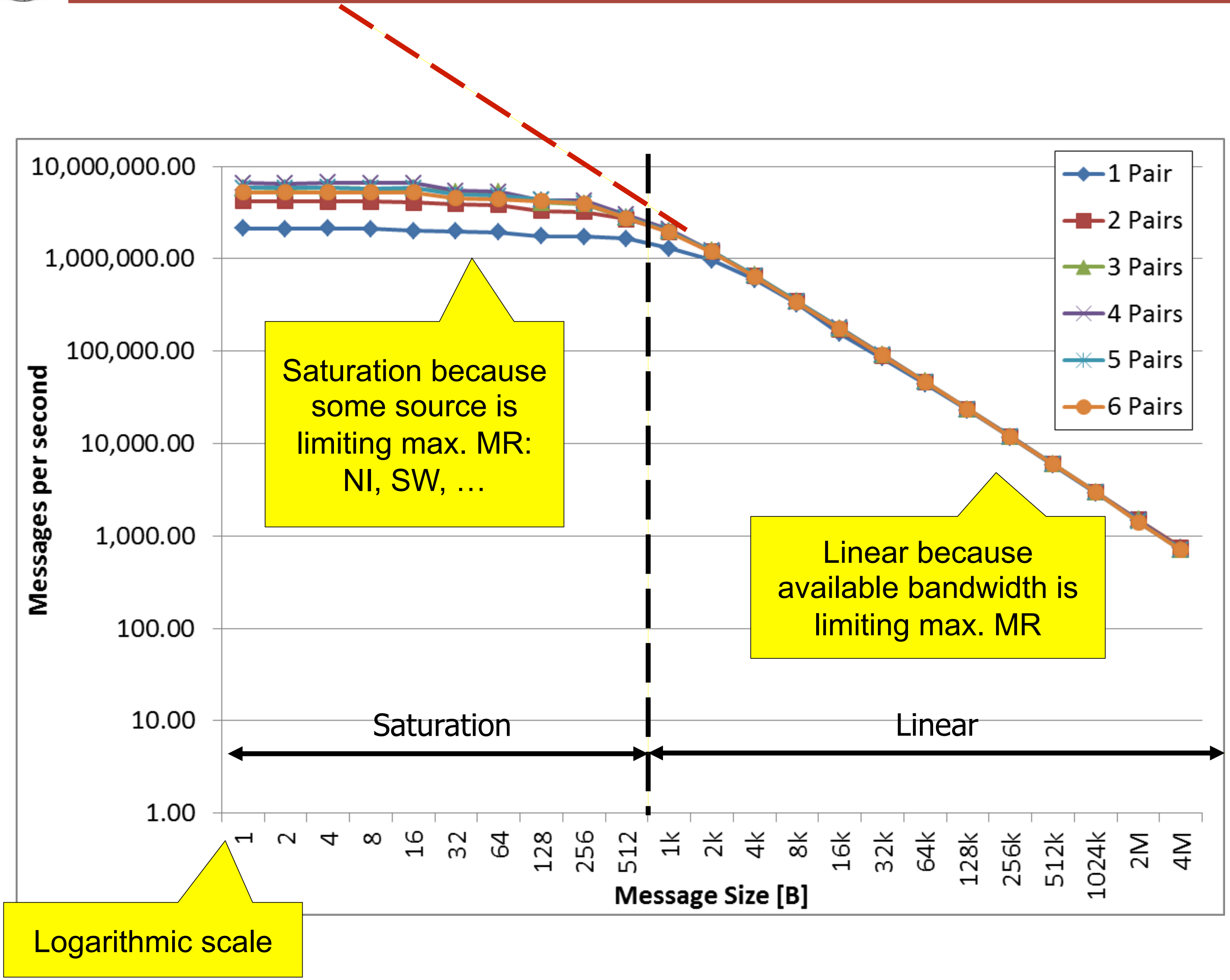
- Sandia MPI Micro-Benchmark Suite (SMB)
- Ohio State University (OSU) Micro-Benchmarks
  - Both report cumulative messages per second
  - Ensure correct mapping!



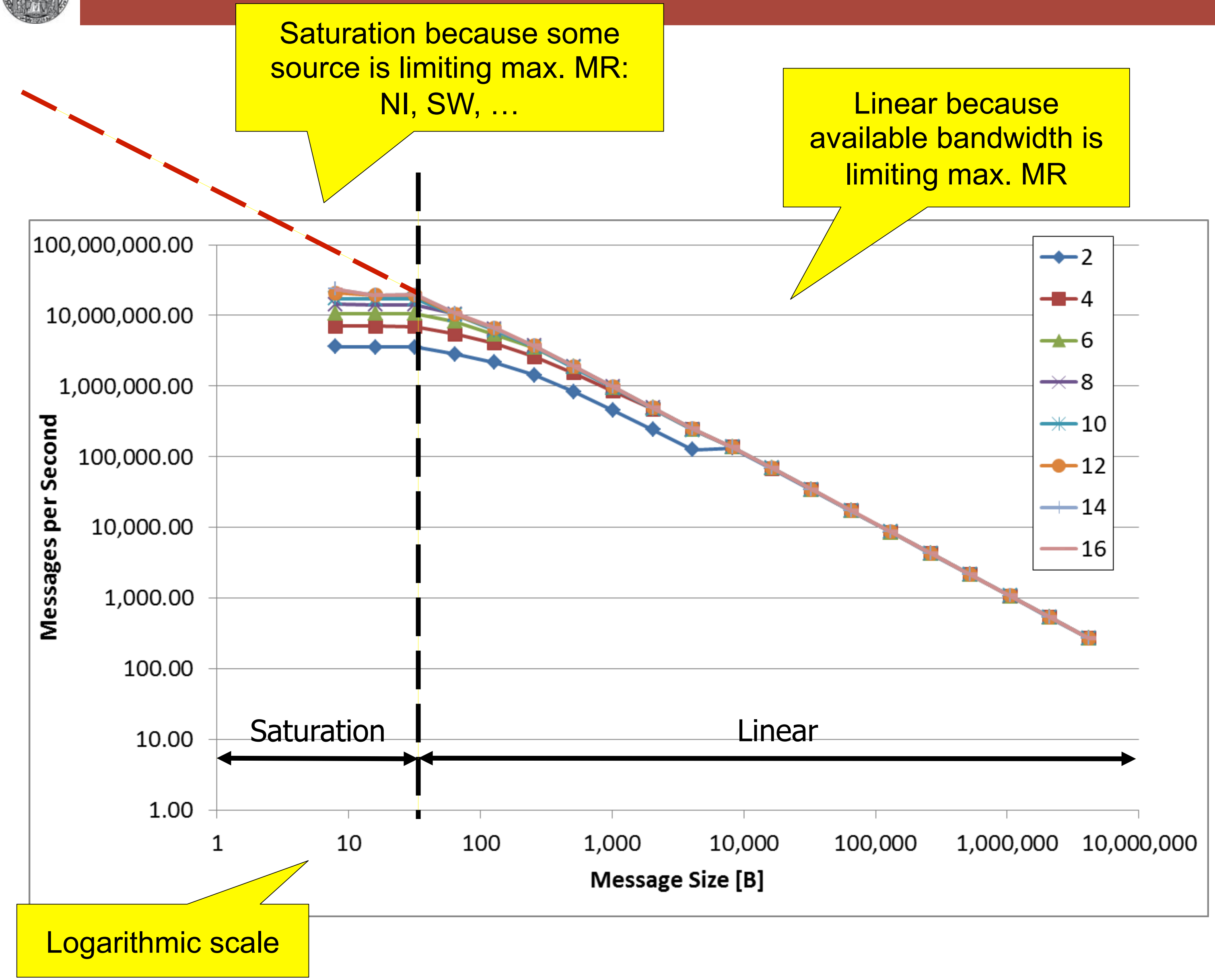
Example: OSU_MBW_MR & Gigabit Ethernet



Example: OSU_MBW_MR & Infiniband



Example: OSU_MBW_MR & EXTOLL R2 (FPGA)



Summary

„Right“ characteristic has to be chosen based on a certain workload!

- Several characteristics
  - Latency
  - Bandwidth
  - Message rate
  - Overhead and Overlap

Optimization strategy depends on networking features

- Complete communication stack contributes
  - Software for sending and receiving: API, libraries, drivers
  - Network interface architecture
  - Network switching resources & links
- Not covered here: contention and congestion
  - Can have huge impact on performance
  - Characterization of those highly depends on applied workload