# Evaluation of High-Performance Networks as Compilation Targets for Global Address Space Languages

**Mike Welcome**

*In conjunction with the joint UCB and NERSC/LBL UPC compiler development project*

http://upc.nersc.gov

# GAS Languages

- **Access to remote memory is performed by de-referencing a variable**
  - **Cost of small (single word) messages is important**

- **Desirable Qualities of Target Architectures**
  - **Ability to perform one-sided communication**
  - **Low latency performance for remote accesses**
  - **Ability to hide network latency by overlapping communication with computation or other communication**
  - **Support for collective communication and synchronization operations**

# Purpose of this Study

- **Measure the performance characteristics of various UPC/GAS target architectures.**
  - **We use micro-benchmarks to measure network parameters, including those defined in the LogP model.**
- **Given the characteristics of the communication subsystem, should we…**
  - **Overlap communication with computation?**
  - **Group communication operations together?**
  - **Aggregate (pack/unpack) small messages?**

# Target Architectures

- Cray T3E
  - 3D Torus Interconnect
  - Directly read/write E-registers
- IBM SP
- Quadrics/Alpha Quadrics/Intel
- Myrinet/Intel
- Dolphin/Intel
  - Torus Interconnect
  - NIC on PCI bus
- Giganet/Intel (old, but could foreshadow InfiniBand)
  - Virtual Interface Architecture
  - NIC on PCI bus

# IBM SP

- Hardware: NERSC SP – Seaborg
  - 208 - 16 processor Power 3+ SMP nodes running AIX
- Switch Adapters
  - 2 Colony (switch2) adapters per node connected to a 2GB/sec 6XX memory bus (not PCI).
  - No RDMA, reliable delivery or hardware assist in protocol processing
- Software
  - “user space” protocol for kernel bypass
  - 2 MPI libraries – single threaded & thread-safe
  - LAPI
    - Non-blocking one-sided remote memory copy ops
    - Active messages
    - Synchronization via counters and fence (barrier) ops
    - Polling or Interrupt mode

# Quadrics

- Hardware: Oak Ridge —”Falcon” cluster
  - 64 4-way Alpha 667 MHz SMP nodes running Tru64
- Low latency network
  - Onboard 100 MHz processor with 32 MB memory
  - NIC processor can duplicate up to 4 GB of page tables
    - Uses virtual addresses, can handle page faults
  - RDMA allows async, one-sided communication w/o interrupting remote processor.
  - Runs over 66 MHz, 64 bit PCI bus
  - Single switch can handle 128 nodes: federated switches can go up to 1024 nodes
- Software:
  - Supports MPI, T3E’s shmem, and ‘elan’ messaging APIs
  - Kernel bypass provided by elan layer

# Myrinet 2000

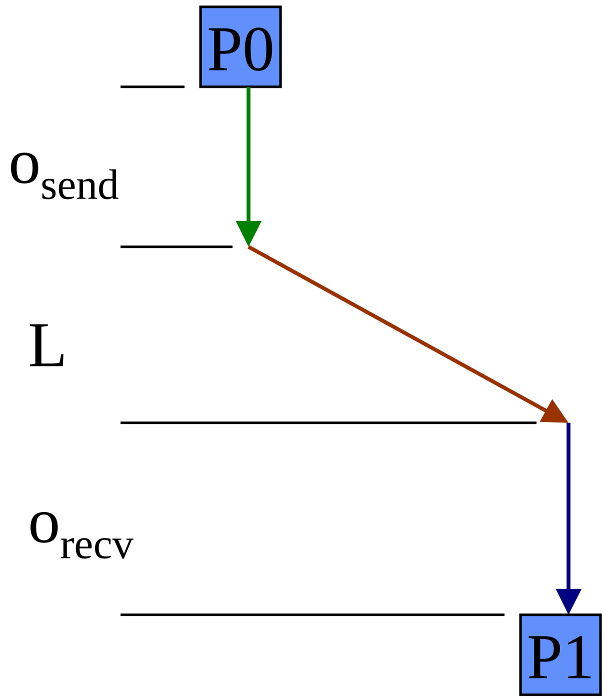
- **Hardware: UCB Millennium cluster**
  - **4-way Intel SMP, 550 MHz with 4GB/node**
    - **33 MHz 32 bit PCI bus**
  - **Myricom NIC: PCI64B**
    - **2MB onboard ram**
    - **133 MHz LANai 9.0 onboard processor**
- **Software: MPI & GM**
  - **GM provides:**
    - **Low-level API to control NIC sends/recvs/polls**
    - **User space API with kernel bypass**
    - **Support for zero-copy DMA directly to/from user address space**
      - Uses physical addresses, requires memory pinning

# The Network Parameters

- **EEL – End to end latency or time spent sending a short message between two processes.**
- **BW – Large message network bandwidth**
- **Parameters of the LogP Model**
  - **L – “Latency”or time spent on the network**
    - **During this time, processor can be doing other work**
  - **O – “Overhead” or processor busy time on the sending or receiving side.**
    - **During this time, processor cannot be doing other work**
    - **We distinguish between “send” and “recv” overhead**
  - **G – “gap” the rate at which messages can be pushed onto the network.**
  - **P – the number of processors**

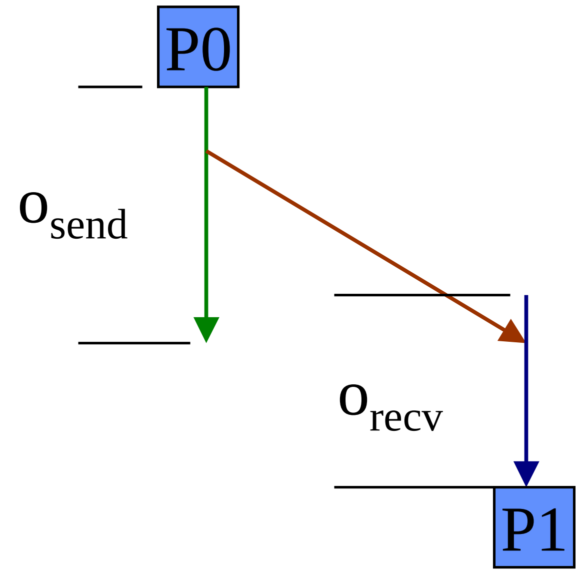
# LogP Parameters: Overhead & Latency

- **Non-overlapping overhead**




$$EEL = o_{send} + L + o_{recv}$$

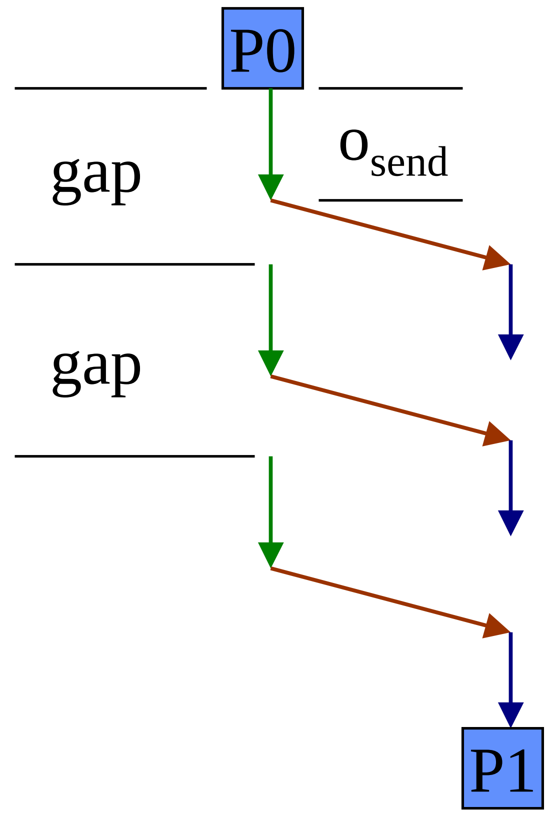
- **Send and recv overhead can overlap**




$$EEL = f(o_{send}, L, o_{recv})$$

# LogP Parameters: gap

- **The Gap is the delay between sending messages**
- **Gap could be larger than send ovhd**
  - **NIC may be busy finishing the processing of last message and cannot accept a new one.**
  - **Flow control or backpressure on the network may prevent the NIC from accepting the next message to send.**
- **The gap represents the inverse bandwidth of the network for small message sends.**



# LogP Parameters and Optimizations

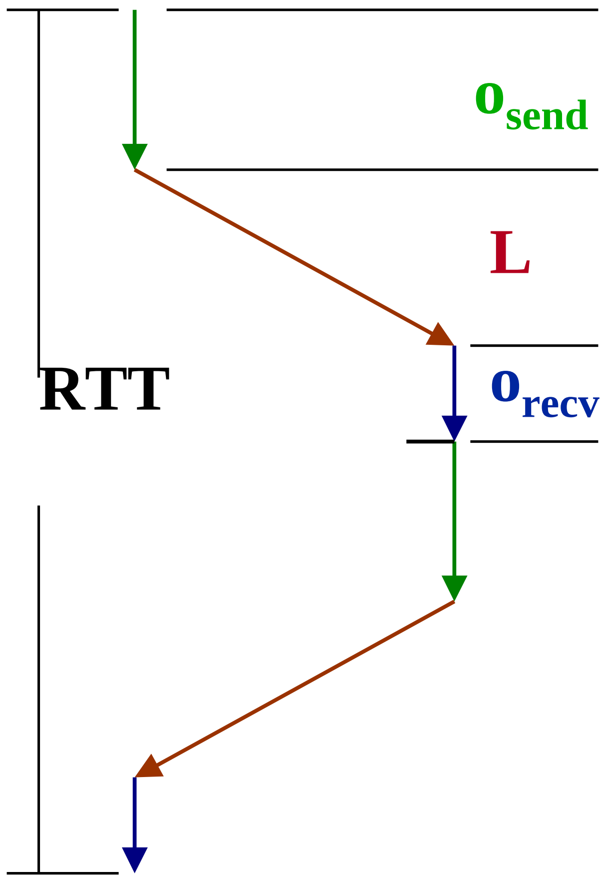
- If gap > $o_{send}$
  - Arrange code to overlap computation with communication
- The gap value can change if we queue multiple communication operations back-to-back
  - If the gap decreases with increased queue-depth
    - Arrange the code to overlap communication with communication (back-to-back).
- If EEL is invariant of message size, at least for a range of message sizes
  - Aggregate (pack/unpack) short message if possible

# Benchmarks

- **Designed to measure the network parameters for each target network.**
  - **Also provide: gap as function of queue depth**
- **Implemented once in MPI**
  - **For portability and comparison to target specific layer**
- **Implemented again in target specific communication layer:**
  - **LAPI**
  - **ELAN**
  - **GM**
  - **SHMEM**
  - **VIPL**

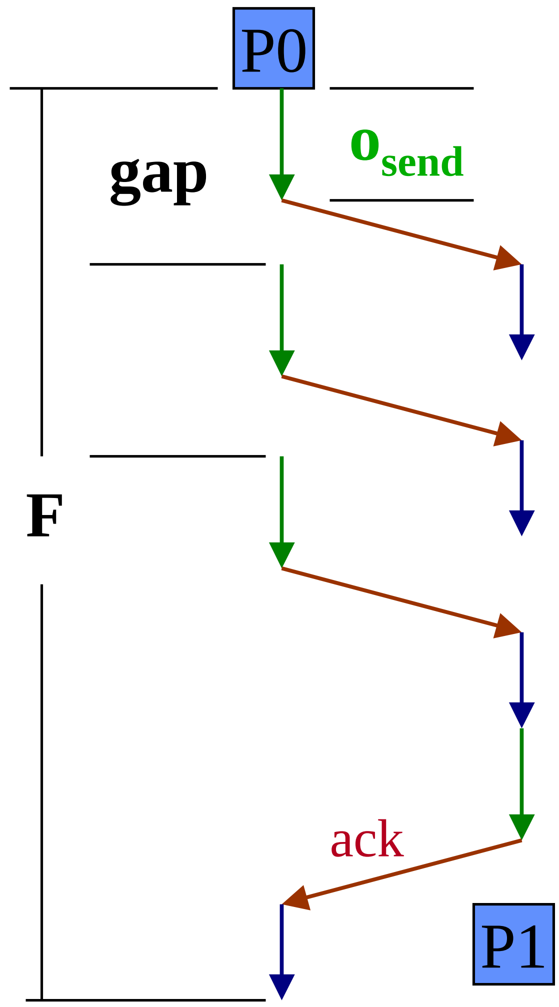
# Benchmark: Ping-Pong

- **Measure the round trip time (RTT) for messages of various size**
- **Report the average RTT of a large number (10000) of message sends.**
- **EEL = RTT/2 = $f(L, o_{send}, o_{recv})$**
- **Approximate:**
  - $f(L, o_{send}, o_{recv}) = L + o_{send} + o_{recv}$
- **Also provides large message bandwidth measurement**



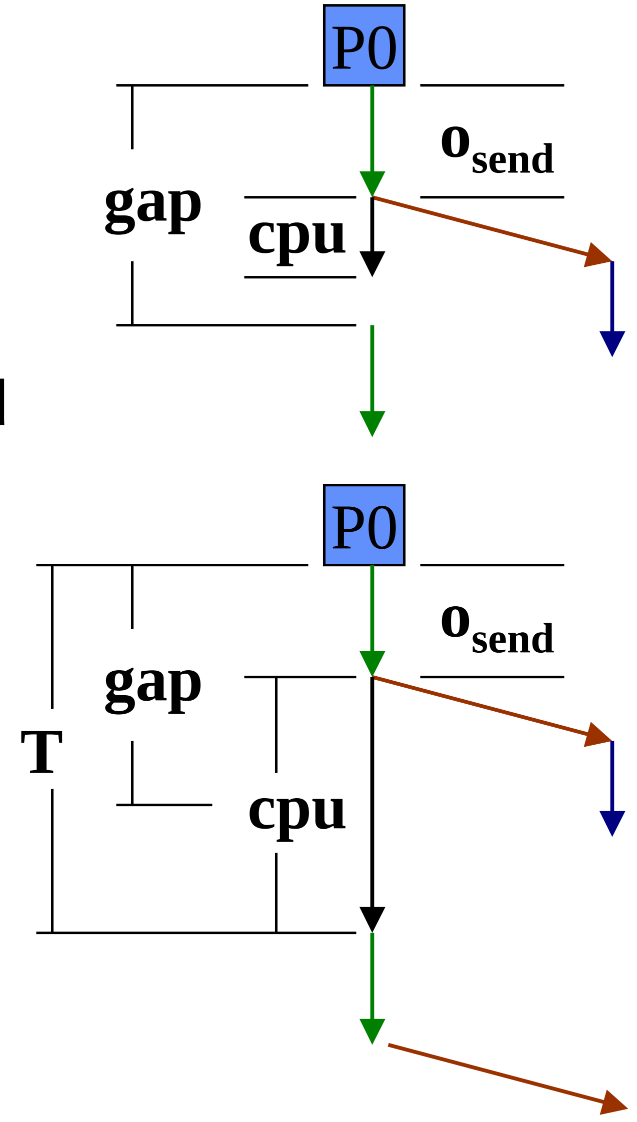
# Benchmark: Flood Test

- Calculate the rate at which messages can be injected into the network.
- Issue N=10000 non-blocking send messages and wait for final ack from receiver.
  - Next send is issued as soon as previous send is complete at sender.
- $F = 2o + L + N*\max(o_{send},g)$
- $F_{avg} = F/N \sim \max(o_{send},g)$
  - For large N
- Can run: Q_Depth >= 1



# Benchmark: Overlap Test

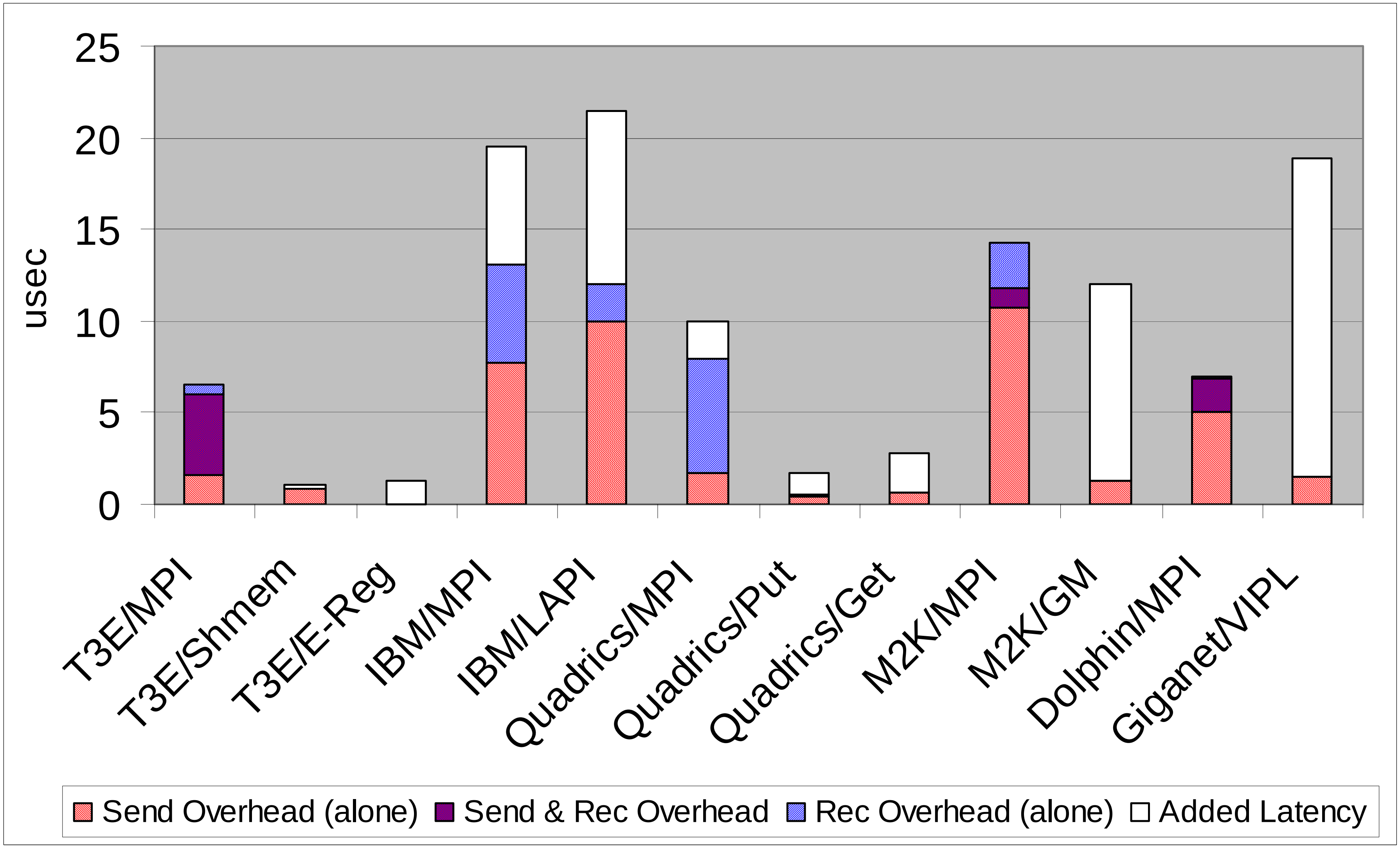
- In the overlap test, we interleave send and receive communication calls with a cpu loop of known duration
- Allows measurement of send and receive overhead.
- Similar to the Flood Test, we can measure the average value of T.
- We vary the "cpu" time until T begins to increase, at T*
  - $o_{send}$ = T* – cpu
- By moving the cpu loop to recv side we measure $o_{recv}$



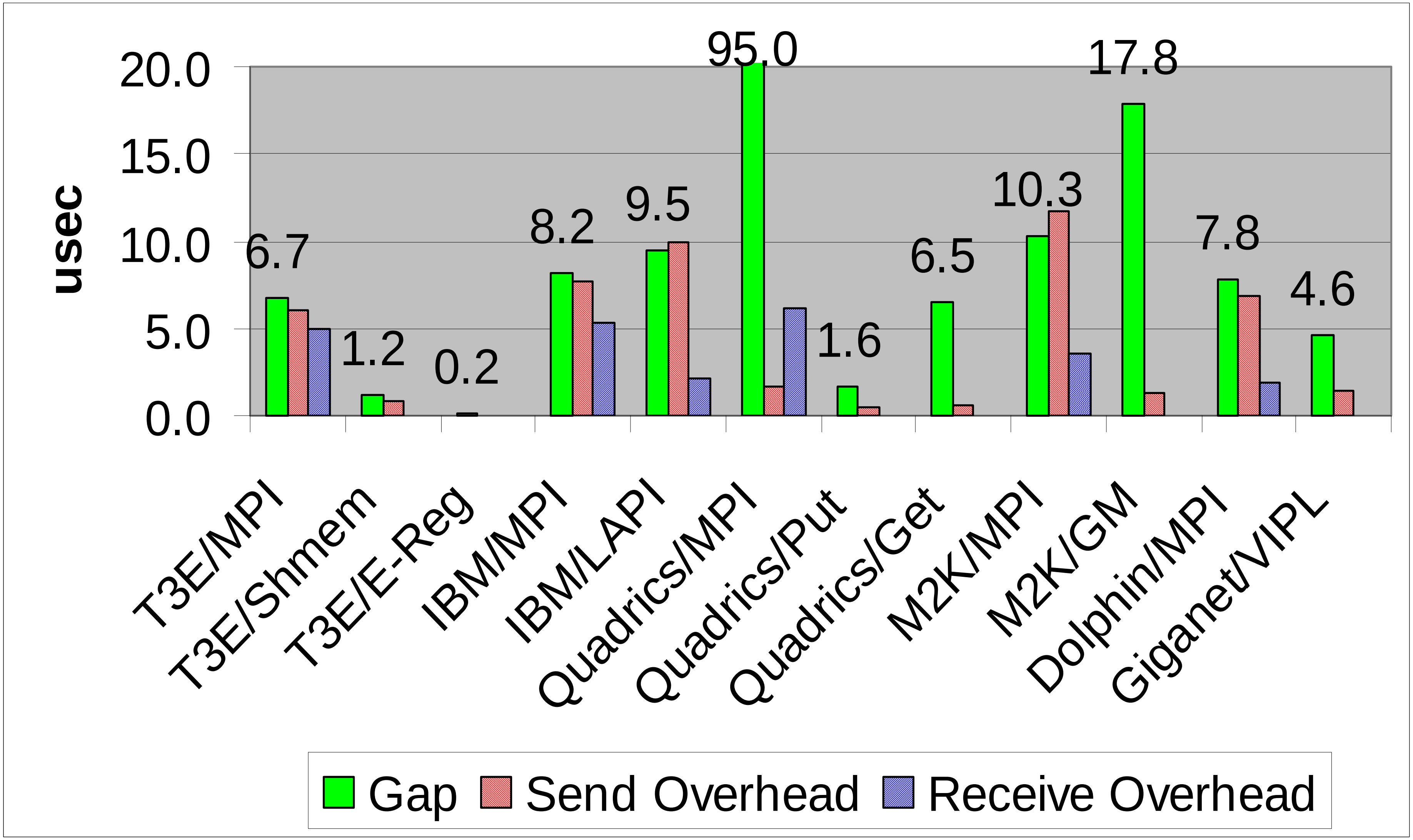
# Putting it all together…

- From Overlap Test, we get:
  - $o_{send}$
  - $o_{recv}$
- From Ping-Pong Test:
  - EEL
  - BW
  - If no overlap of send and receive processing:
    - $L = EEL - o_{send} - o_{recv}$
- From Flood Test:
  - $F_{avg} = \max(o_{send}, g)$
  - If ($F_{avg} > o_{send}$) then
    - $g = F_{avg}$
  - Otherwise
    - cannot measure gap, but its not important

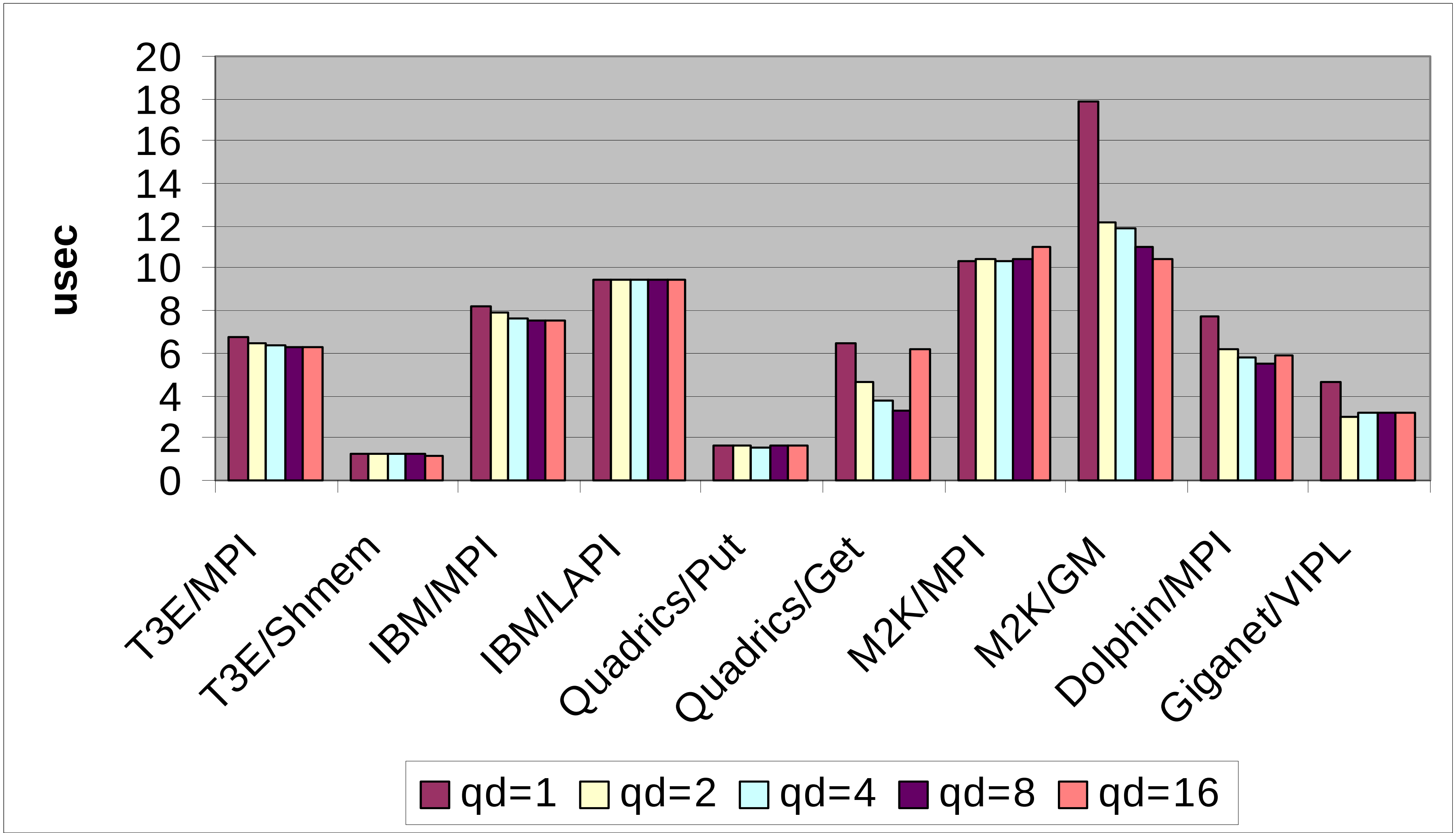
# Results: EEL and Overhead



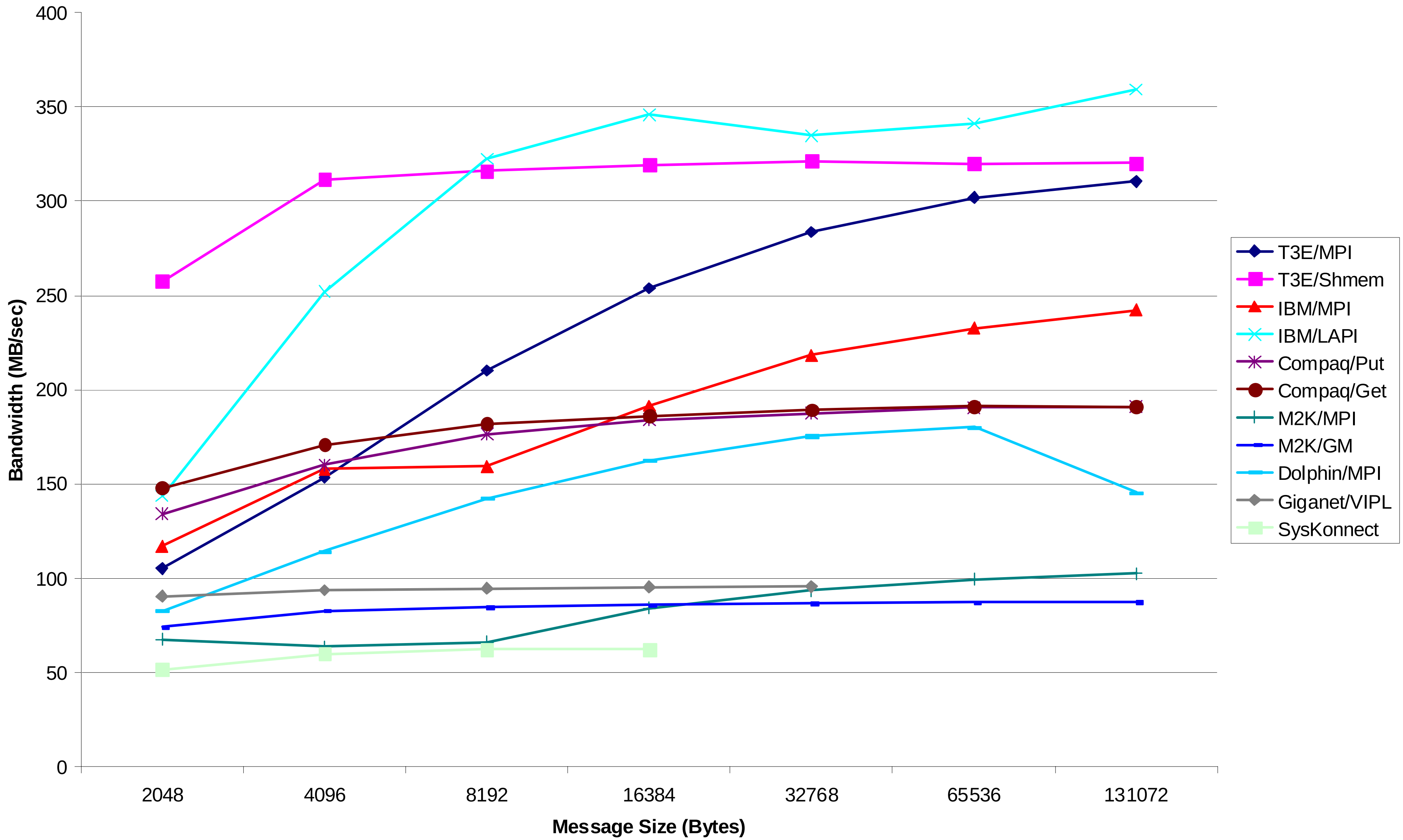
# Results: Gap and Overhead



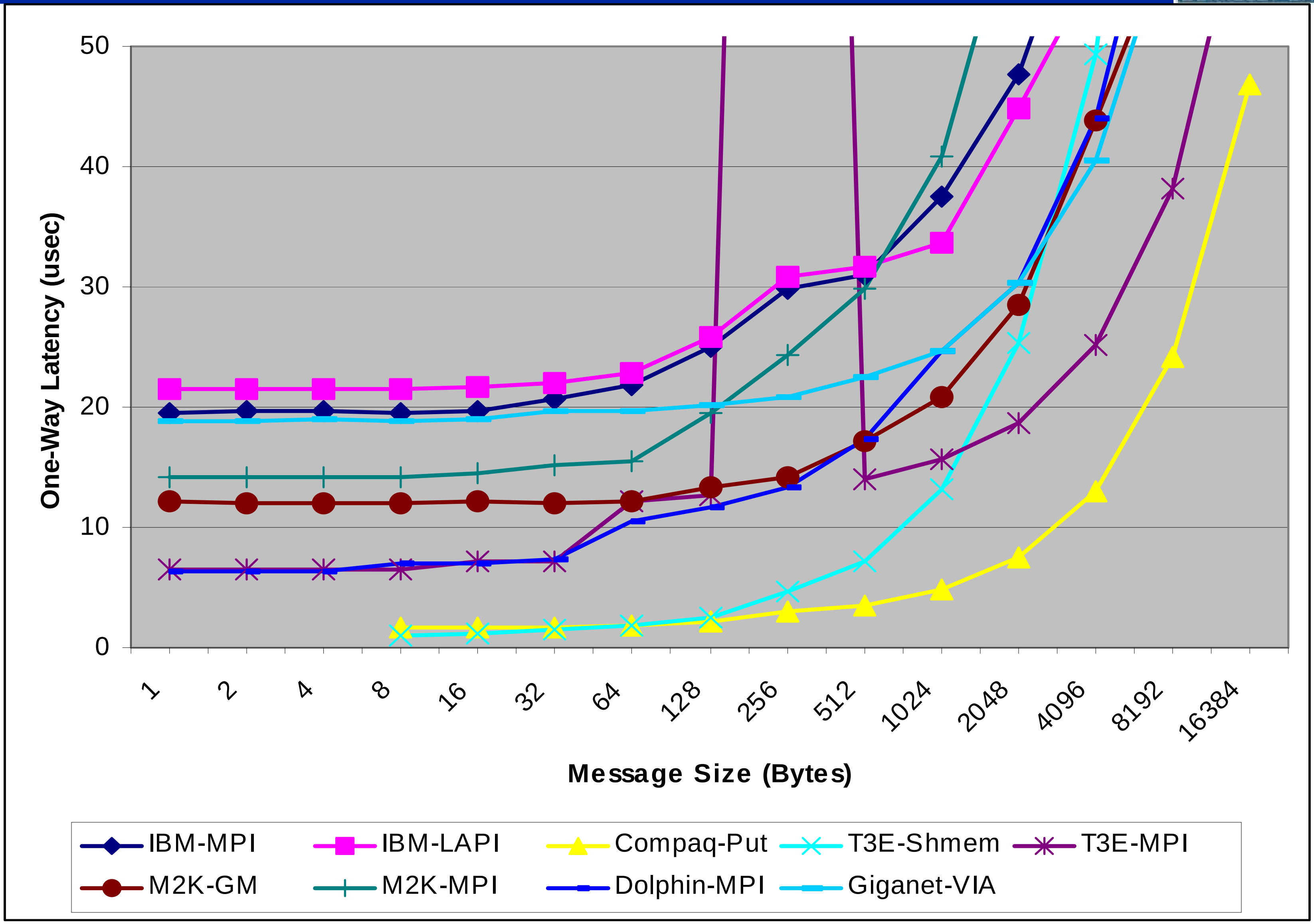
# Flood Test: Overlapping Communication



# Bandwidth Chart



# EEL vs. Message Size



# Benchmark Results: IBM

| IBM Performance | $O_{send}$ usec | Gap usec | $O_{recv}$ usec | EEL usec | L usec | BW MB/s |
|---|---|---|---|---|---|---|
| IBM Published | N/A | N/A | N/A | 17.9 | 2.5* | 500* |
| MPI | 7.8 | 7.6 | 5.4 | 19.5 | 6.3 | 242 |
| LAPI | 9.9 | 9.5 | 2.4 | 21.5 | 9.4 | 360 |

* Theoretical Peak

- **High Latency, High Software Overhead**
- **Gap ~ $O_{send}$**
  - **No overlap of computation with communication**
- **Gap does not vary with number of queued ops**
  - **No overlap of communication with communication**
- **LAPI Cost to send 1 byte ~ cost to send 1KB**
  - **Short message packing is best option**

# Benchmark Results: Myrinet 2000

| Myrinet Performance | $O_{send}$ usec | Gap usec | $O_{recv}$ usec | EEL usec | L usec | BW MB/s |
|---|---|---|---|---|---|---|
| Myricom Published | 0.3 | N/A | N/A | N/A | 9 | 100-130 |
| GM (measured) | 1.3 | 17.8 | ~0 | 12.0 | 10.7 | 88 |

- **Small $o_{send}$ and large gap: g - $o_{send}$ = 16.5 usec**
  - **Overlap of computation with communication a big win**
- **Big reduction in Gap with queue depth > 1 (5-7 usec)**
  - **Overlap of communication with communication is useful**
- **RDMA capability allows for minimal $o_{recv}$**
- **Bandwidth limited by 33MHz 32bit PCI bus. Should improve with better bus.**

# Benchmark Results: Quadrics

| Quadrics Performance | $O_{send}$ usec | Gap usec | $O_{recv}$ usec | EEL usec | L usec | BW MB/s |
|---|---|---|---|---|---|---|
| Quadrics Published | N/A | N/A | N/A | 2 | N/A | N/A |
| MPI (measured) | 1.7 | 95.0* | 6.2 | 9.9 | 2.0 | 470* |
| Quadrics Put | 0.5 | 1.6 | ~0 | 1.7 | 1.2 | 180 |

* MPI Bugs?

- **Observed one-way msg time slightly better than advertised!**
- **Using shmem/elan is big savings over MPI for latency and CPU overhead.**
- **No CPU overhead on remote processor w/shmem**
- **Some computation overlap is possible**
- **MPI implementation a bit flaky…**

# General Conclusions

- **Overlap of Computation with Communication**
  - **A win on systems with HW support for protocol processing**
    - **Myrinet, Quadrics, Giganet**
  - **MPI $o_{send}$ ~ gap on most systems: no overlap.**

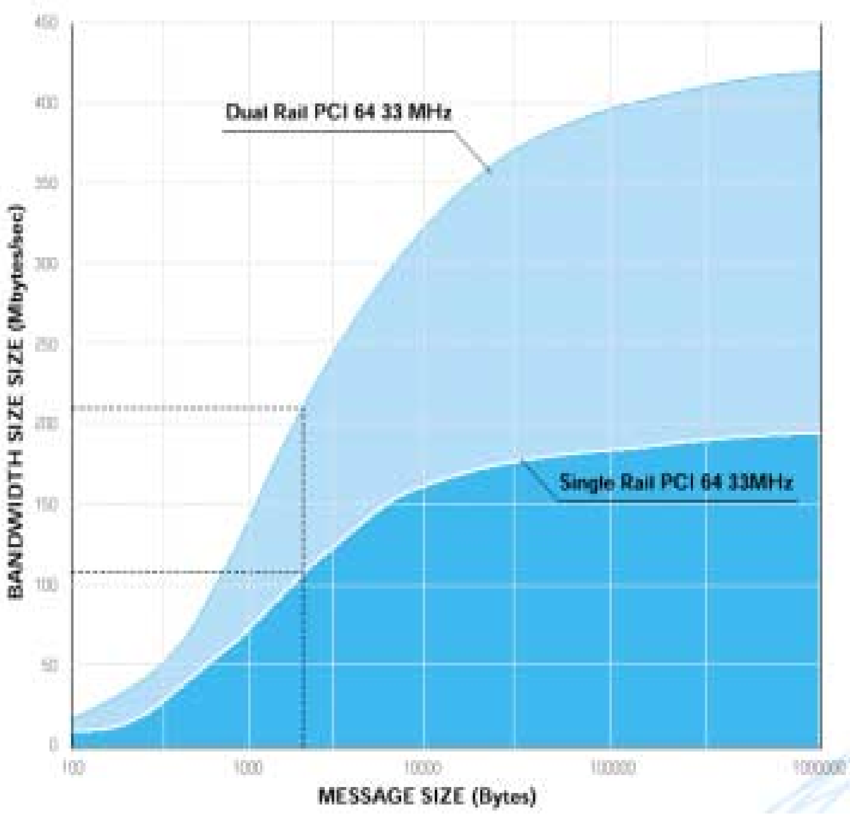
- **Overlap of Communication with Communication**
  - **Win on Myrinet, Quadrics, Giganet**
  - **Most MPI implementation exhibit this to a minor extent**

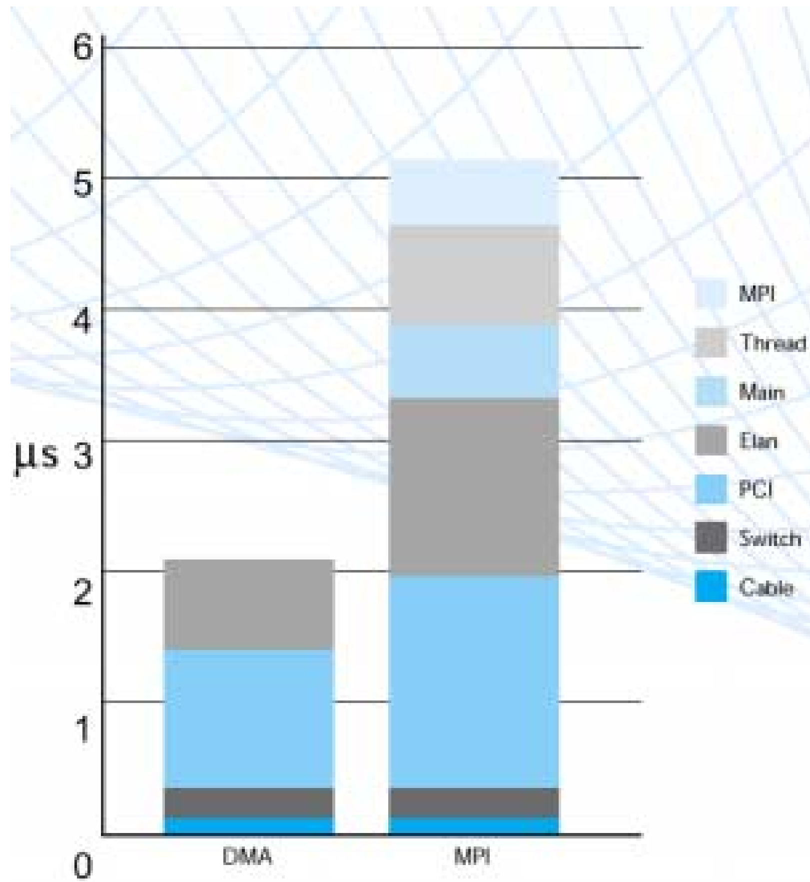
- **Aggregation of small messages (pack/unpack)**
  - **A win on all systems**

# Old/Extra Slides

# Quadrics

Advertised Bandwidth/latency, with PCI bottleneck shown



# IBM SP – Hardware Used

- NERSC SP – Seaborg
  - 208 - 16 processor Power 3+ SMP nodes
  - 16 – 64 GB memory per node
- Switch Adapters
  - 2 Colony (switch2) adapters per node connected to a 2GB/sec 6XX memory bus (not PCI).
  - Csss “bonding” driver will multiplex through both adapters
  - On-board 740 PowerPC processor
  - On-board firmware and RamBus memory for segmentation and re-assembly of user packets to and from 1KB switch packets.
  - No RDMA, reliable delivery or hardware assist in protocol processing

# IBM SP - Software

- AIX “user space” protocol for kernel bypass access to switch adapter
- 2 MPI libraries – single threaded and thread-safe
  - Thread-safe version increases RTT latency by 10-15 usec
- LAPI – Lowest level comm API exported to user
  - Non-blocking one-sided remote memory copy ops
  - Active messages
  - Synchronization via counters and fence (barrier) ops
  - Thread-safe (locking overhead)
  - Mulit-threaded implementation:
    - Notification thread (progress engine)
    - Completion handler thread for active messages
  - Polling or Interrupt mode
  - Software based flow-control and reliable delivery (overhead)

# Quadrics

- **Low latency network, w/100 MHz processor on NIC**
  - **RDMA allows async, one-sided communication w/o interrupting remote processor.**
  - **Supports MPI, T3E's shmem, and 'elan' messaging APIs.**
  - **Advertised one way latency as low as 2 us (5 us for MPI).**
  - **Single switch can handle 128 nodes: federated switches can go up to 1024 nodes (Pittsburgh running 750 nodes).**
  - **NIC processor can duplicate up to 4 GB of page tables—good for global address space languages.**
  - **Runs over PCI bus—limits both latency & bandwidth**

**4 node cluster at Oak Ridge Nat'l Lab—"Falcon"**

**4 4-way Alpha 667 MHz SMP nodes running Tru64**

**6 MHz, 64 bit PCI bus**



**uture work: look at Intel/Linux Quadrics cluster at LLNL**

# Myrinet 2000

- Hardware: UCB Millennium cluster
  - 4-way Intel SMP, 550 MHz with 4GB/node
    - 33 MHz 32 bit PCI bus
  - Myricom NIC: PCI64B
    - 2MB onboard ram
    - 133 MHz LANai 9.0 onboard processor
- Software: GM
  - Low-level API to control NIC sends/recvs/polls
  - User space API with kernel bypass
  - Support for zero-copy DMA directly to/from user address space