

HPS and latest MPI

IBM
July, 2006

Agenda

- **Evolution**
- **Technology**
- **Performance characteristics**
 - **RDMA**
 - **Collectives**

Evolution

- **High Performance Switch (HPS)**
 - Also Known As “Federation”
- **Follow on to SP Switch2**
 - Also known as “Colony”

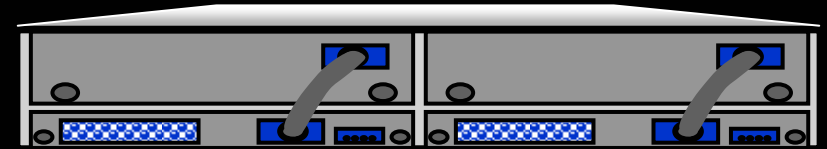
Generation	Processors
Switch	POWER2
SP Switch	POWER2 → POWER3
SP Switch 2 (Colony)	POWER3 → POWER4
HPS (Federation)	POWER4 → POWER5

Technology

- **Internal network**
 - In lieu of, e.g. Gig Ethernet
- **Multiple links per node**
 - Match number of links to number of processors

	SP Switch2	HPS
Latency	15 microsec.	< 5 microsec.
Bandwidth	500 Mbyte/s	2 Gbyte/s
Configuration	1 adaptor per logical node	1 adaptor, 2 links, per node

HPS Packaging



- **4U, 24-inch drawer**
- **16 ports for server-to-switch**
- **16 ports for switch-to-switch connections**
- **Host attachment directly to server bus via 2-link or 4-link Switch Network Interface (SNI)**
 - **Up to two links per pSeries p575**
 - **Up to eight links per pSeries p595**

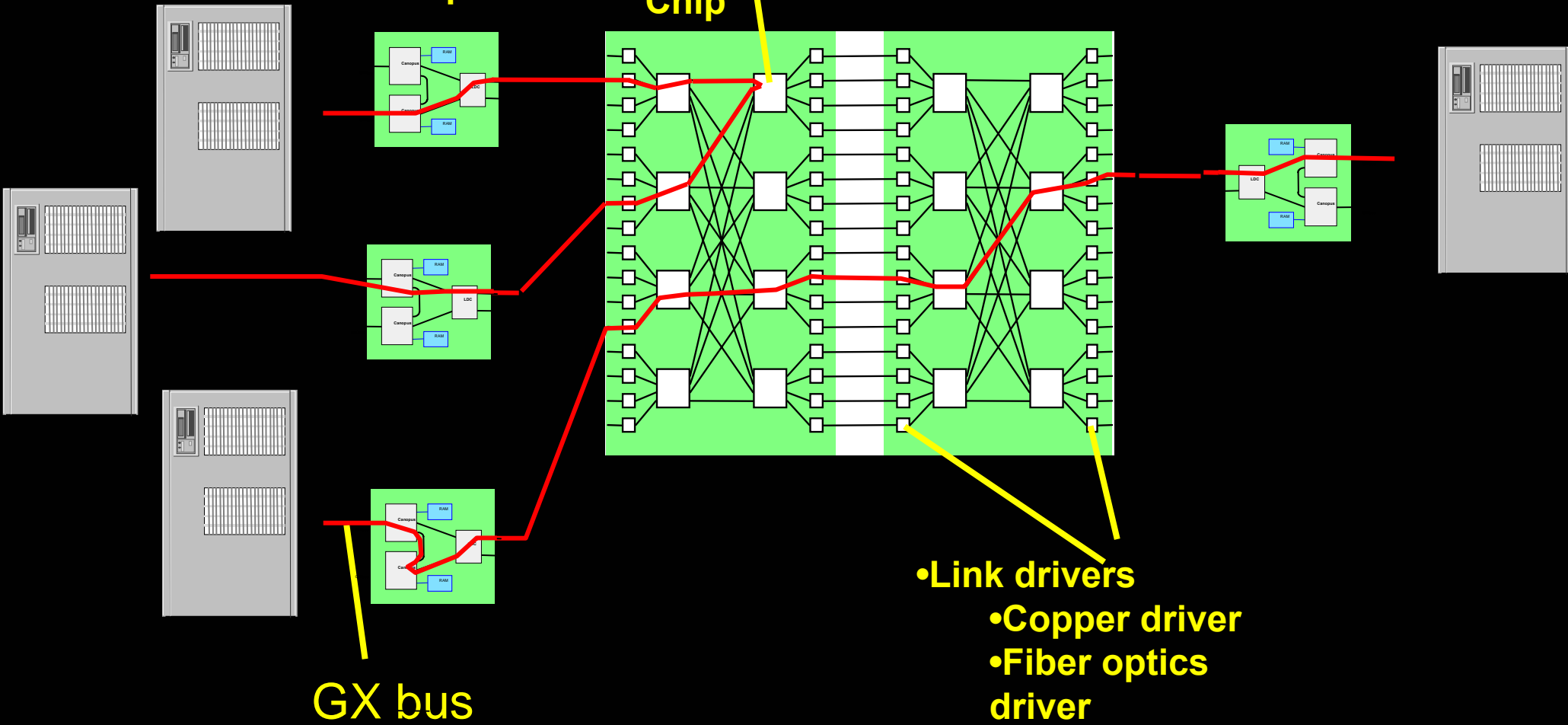
HPS Switch Configuration

Power5 Servers

HPS Adapters

HPS Switch Chip

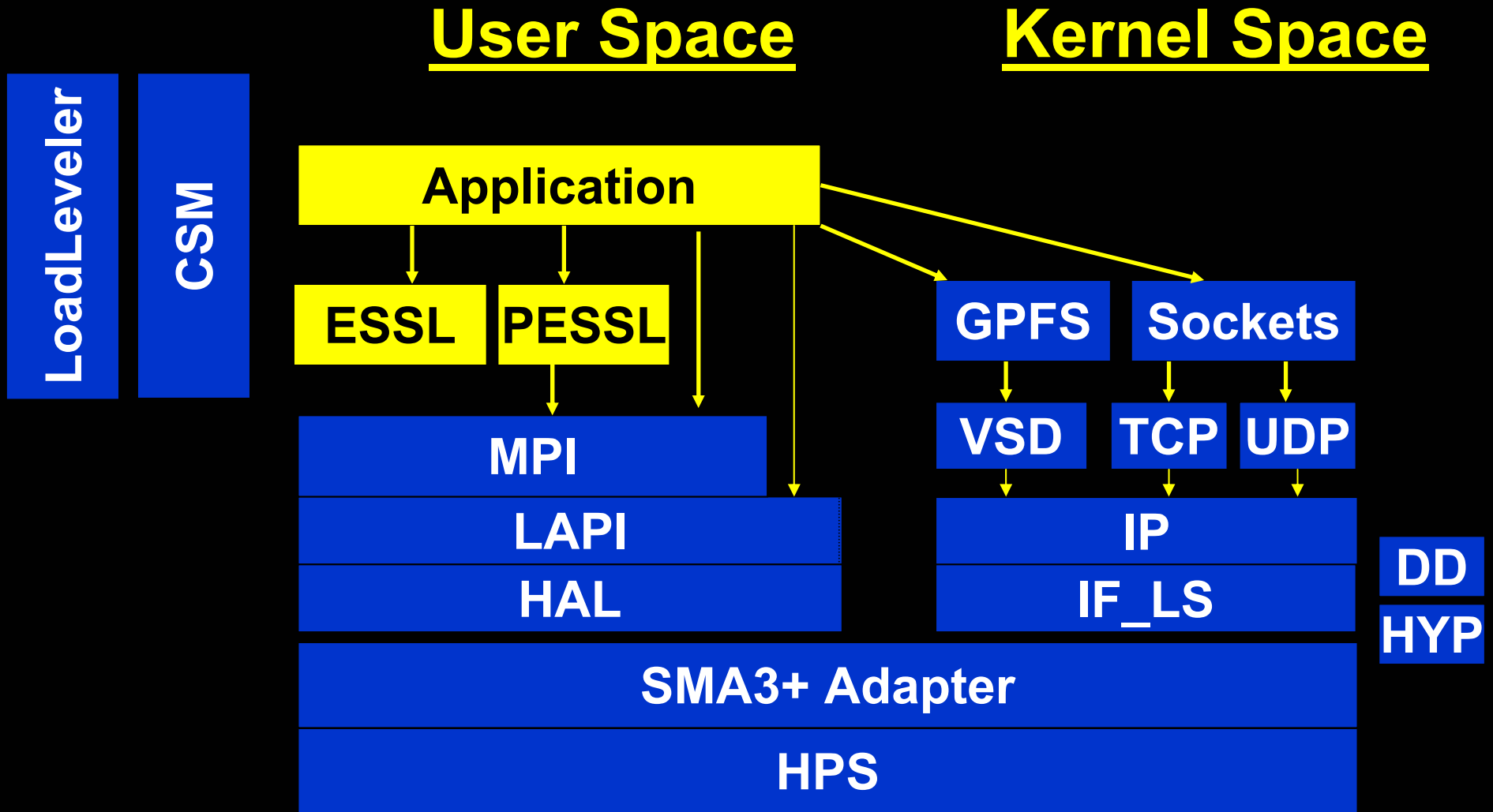
Switch board



HPS Software

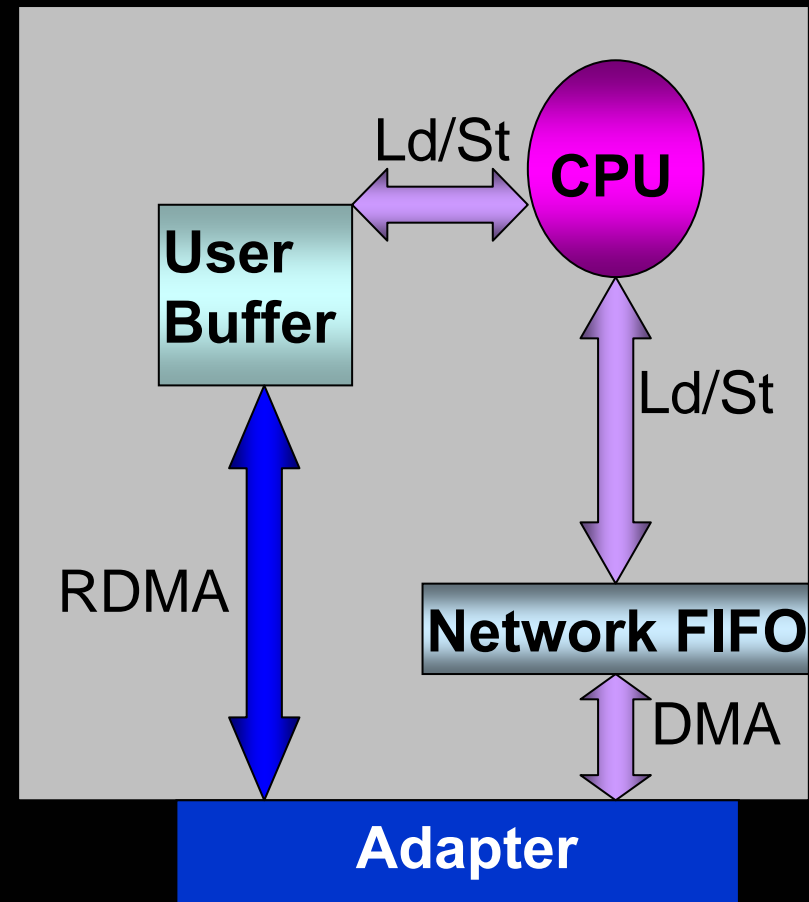
- **MPI-LAPI (PE V4.2.2)**
 - Uses LAPI as the reliable transport
 - Library uses threads, not signals for async activities
- **Existing applications binary compatible**
- **New performance characteristics**
 - Eager
 - Bulk Transfer
 - RDMA
 - Improved collective communication

HPS Software Architecture



Supported Communication Modes

- **FIFO Mode**
 - Message chopped into 2K packet chunks on the host and copied by CPU
 - Memory bus crossing depends on caching. At least 1 IO bus crossing
- **Remote Direct Memory Access (RDMA)**
 - No slave side protocol
 - CPU offload
 - Enhanced Programming model
 - 1 IO bus crossing

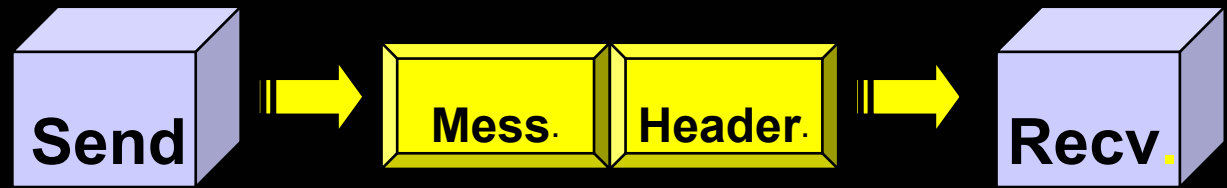


Underlying Message Procedures

- **Protocols:**
 - **Rendezvous**
 - “Large” messages
 - **Eager**
 - “Small” messages
 - **MP_EAGER_LIMIT**
 - Range: 0 - 65536
- **Mechanisms**
 - **Packet**
 - **Bulk**
 - **MP_BULK_MIN_MSG_SIZE**
 - Range: any non-negative integer

MPI Transfer Protocols

**Small Messages:
Eager**

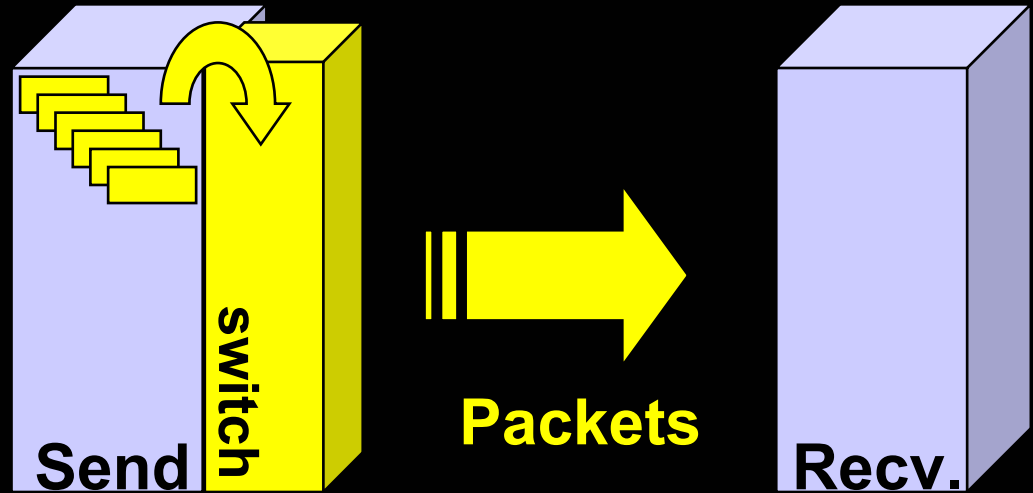


**Large Messages:
Rendezvous**

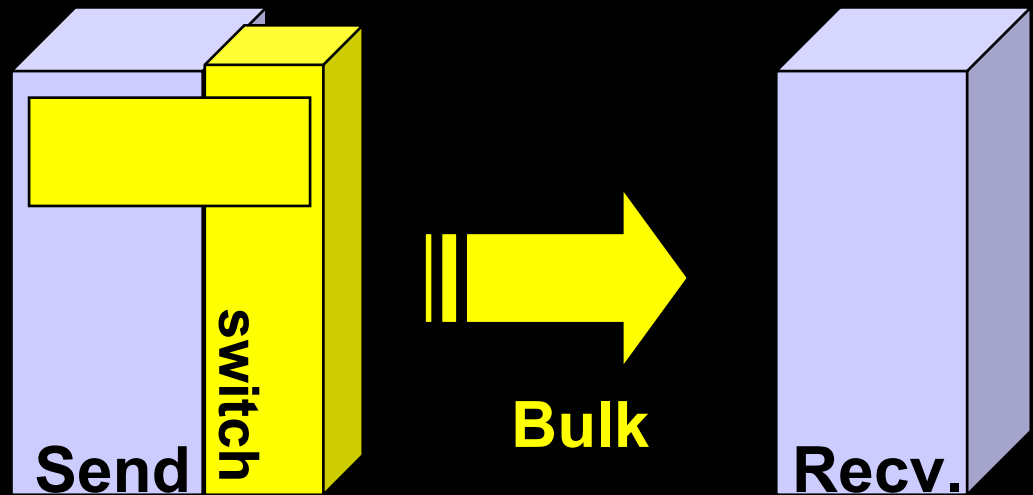


MPI Transfer Mechanisms

**Small Messages:
Packets**



**Large Messages:
Bulk**



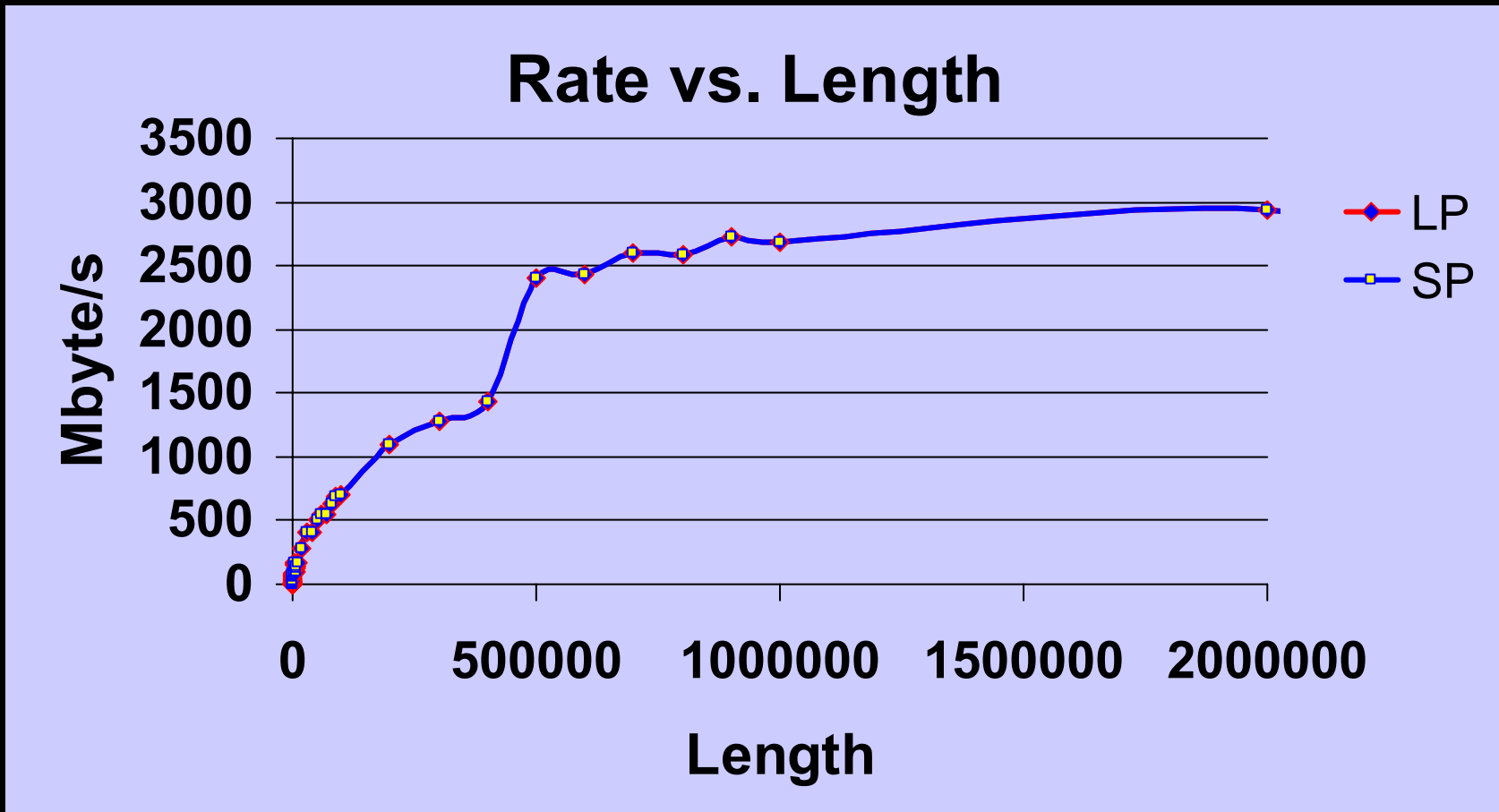
Remote Direct Memory Access (RDMA)

- **Overlap of computation and communication (possible)**
 - Fragmentation and reassembly offloaded to the adapter
 - Minimize packet arrival interrupts
 - Asynchronous messaging applications
 - All tasks sharing adapter not communicating at the same time
- **One sided programming model**
- **Zero copy transport**
 - Reduced memory subsystem load
- **Striping of very large messages**
 - Implications to interference with other tasks if copying

New MPI Performance Models

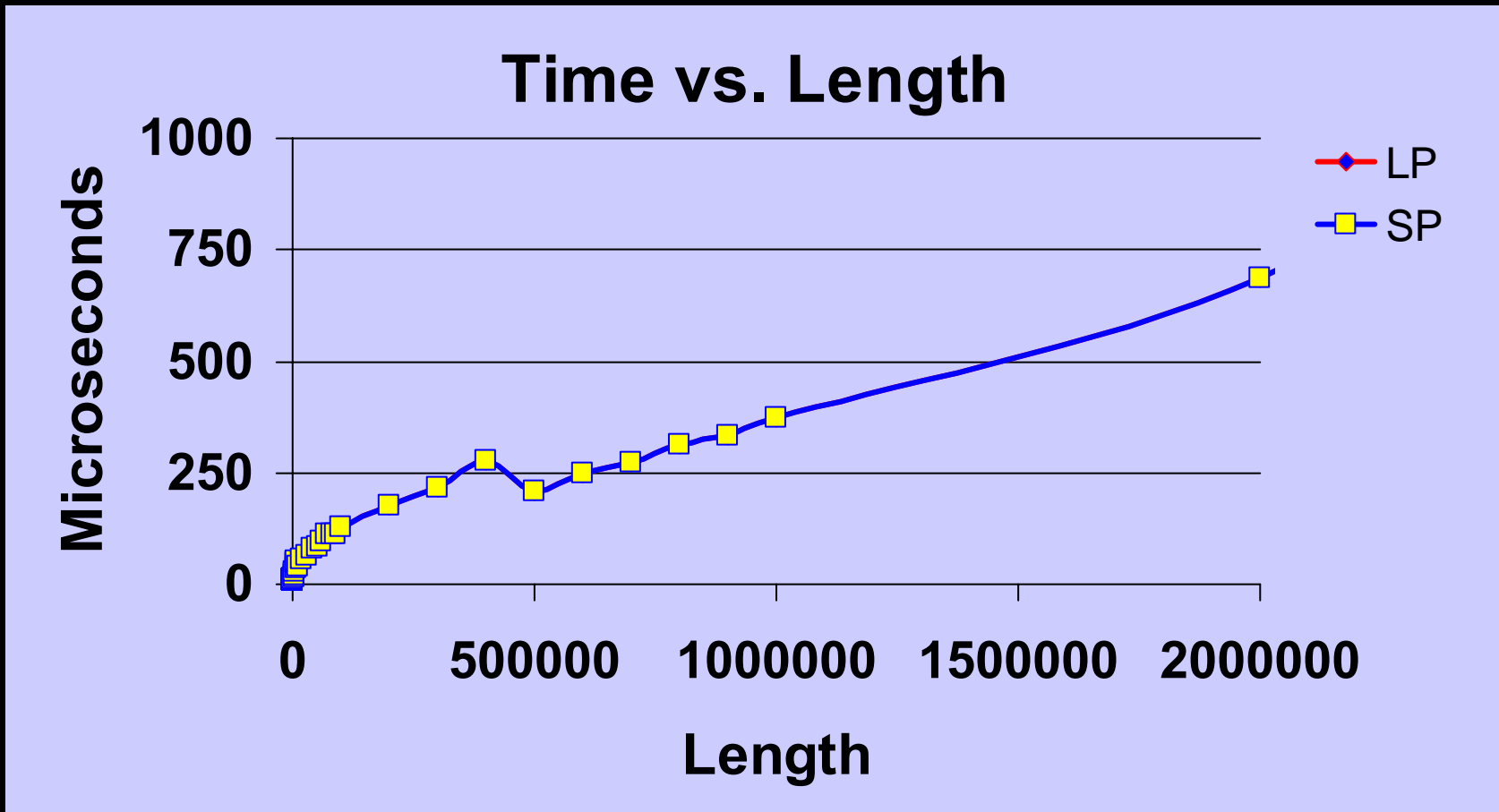
- **Possible striping of single message**
 - **Bandwidth $\sim n * 2$ Gbyte/s**
- **Small dependence on user large pages**
- **Collectives**
- **Eager limit**

MPI Single Messages: Large Page vs. Small Page



p655 1.5 GHz, HPS, RDMA

MPI Single Messages: LP vs. SP

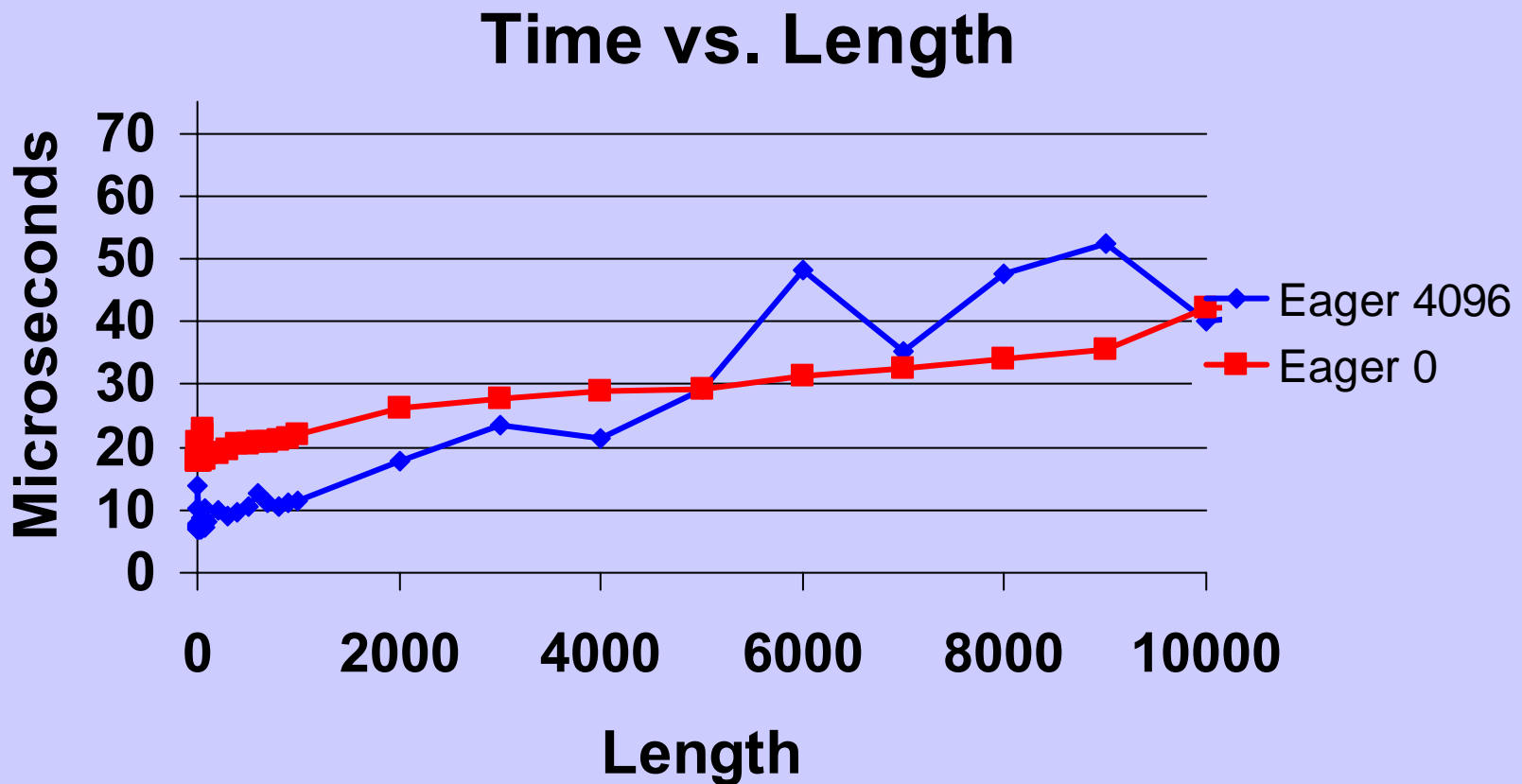


MPI Single Messages

- **Message striping at ~500000 bytes**
- **Bandwidth:**
 - 1.5 Gbyte/s “Modest” size
 - 3 Gbyte/s Large size
- **Small sensitivity to page size**

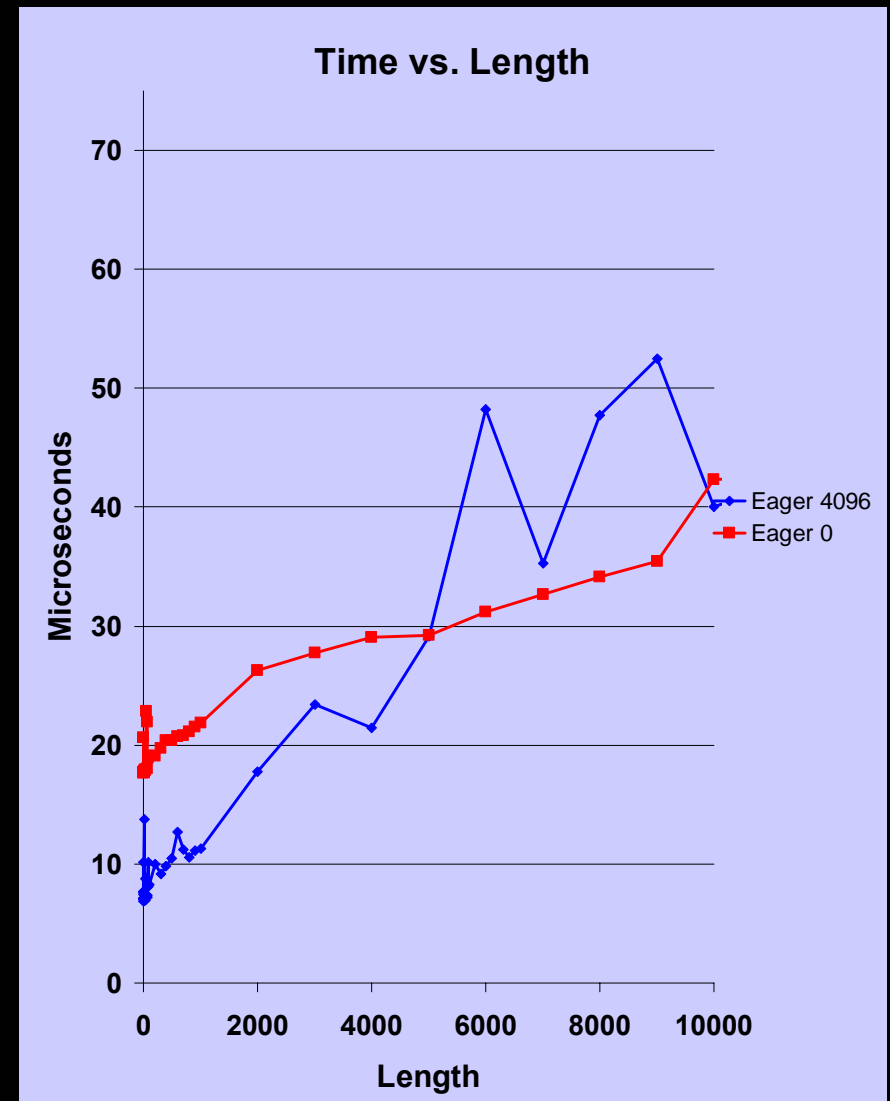


MPI Single Messages: Eager Limits

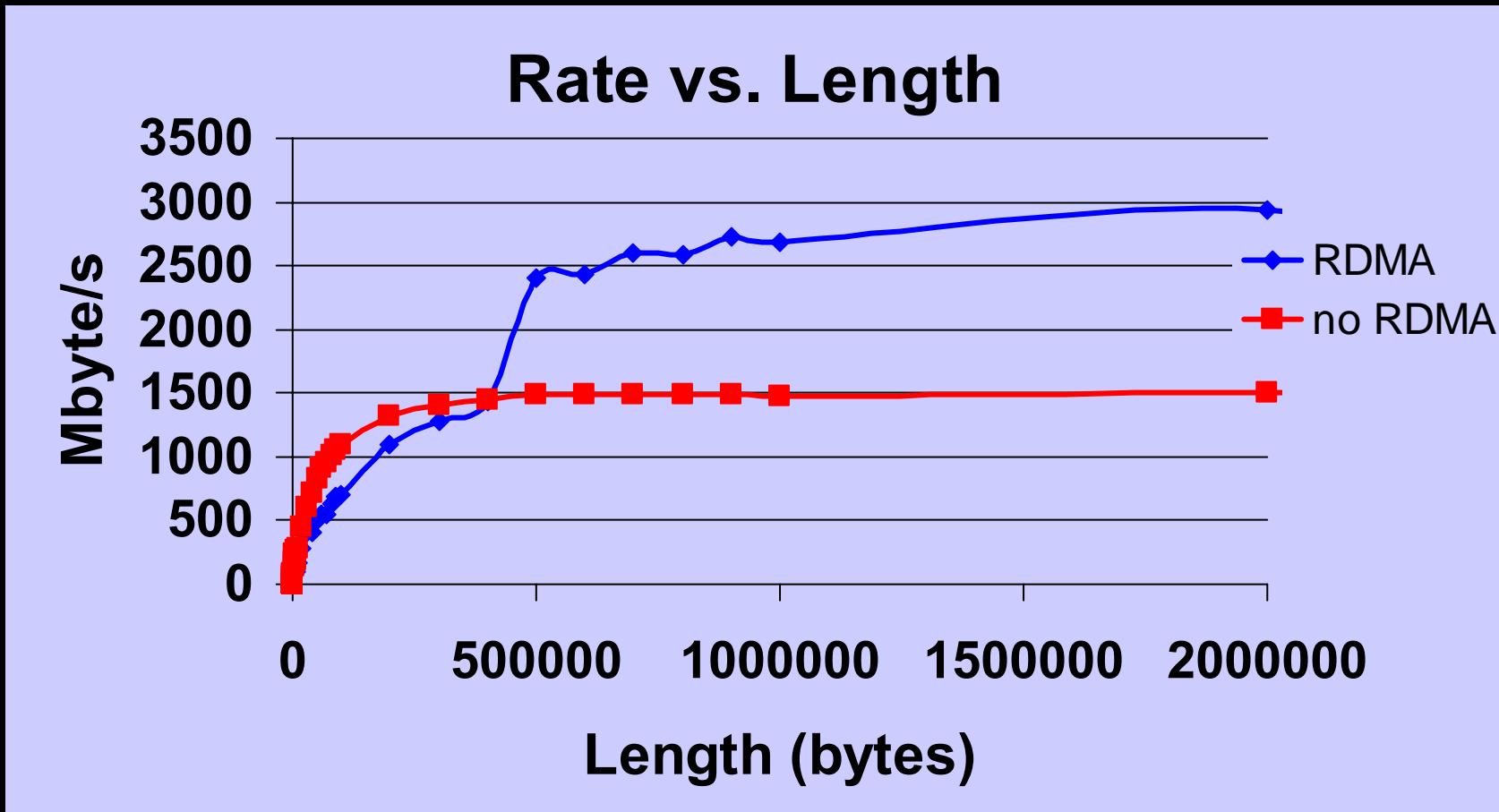


Eager Limit

- Reduce latency from 20 microseconds to 7 microseconds



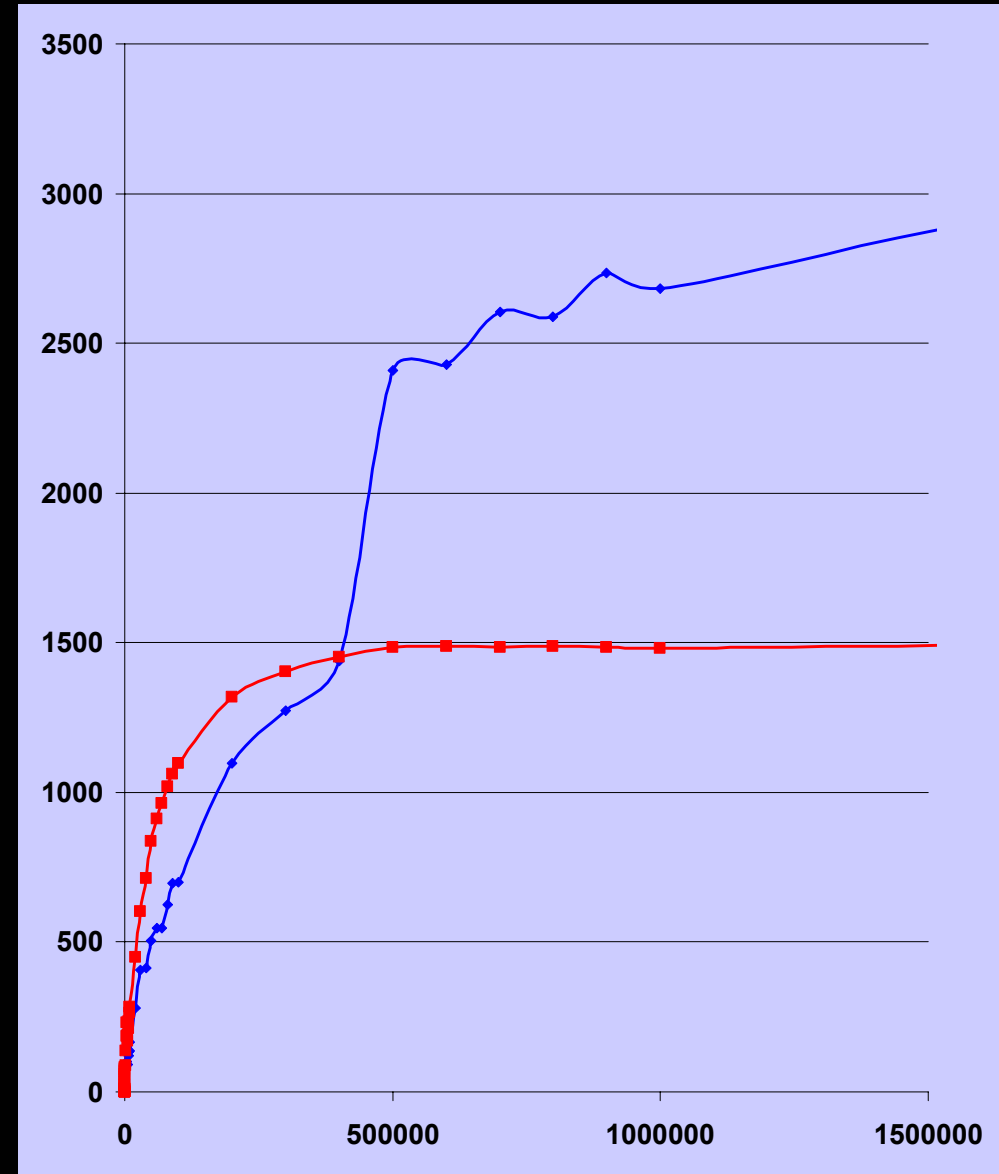
MPI Single Messages: RDMA vs. no RDMA



p655 1.5 GHz, HPS, RDMA

RDMA

- **Message striping starts at 500000 bytes**
 - Adjust with **MP_BULK_MIN_MSG_SIZE**



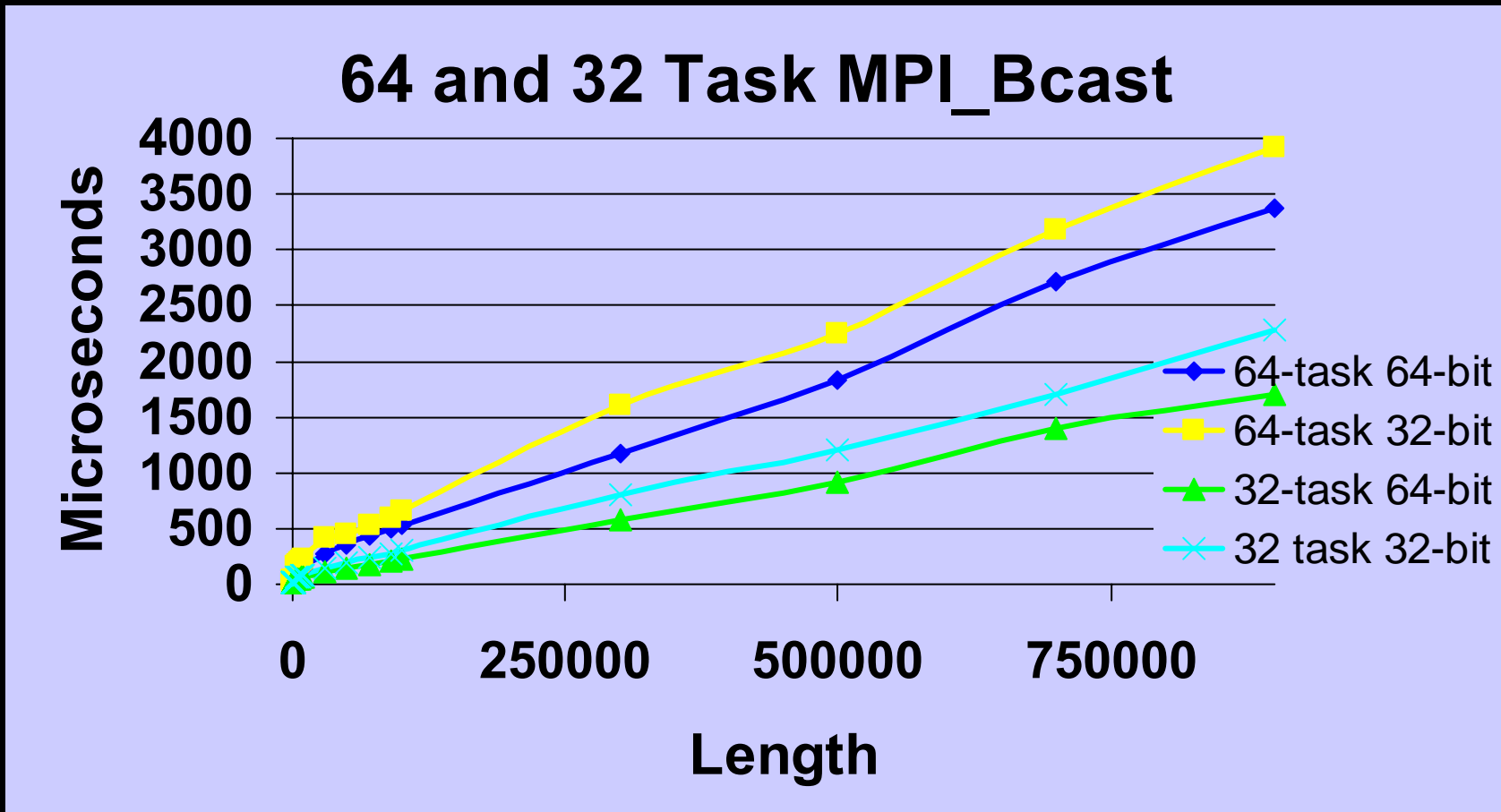
One more thing about Bulk Transfer...

- If running interactively
MP_USE_BULK_XFER=yes is sufficient
- If running via Load Leveler #@ bulkxfer =yes
is necessary

MPI Collectives

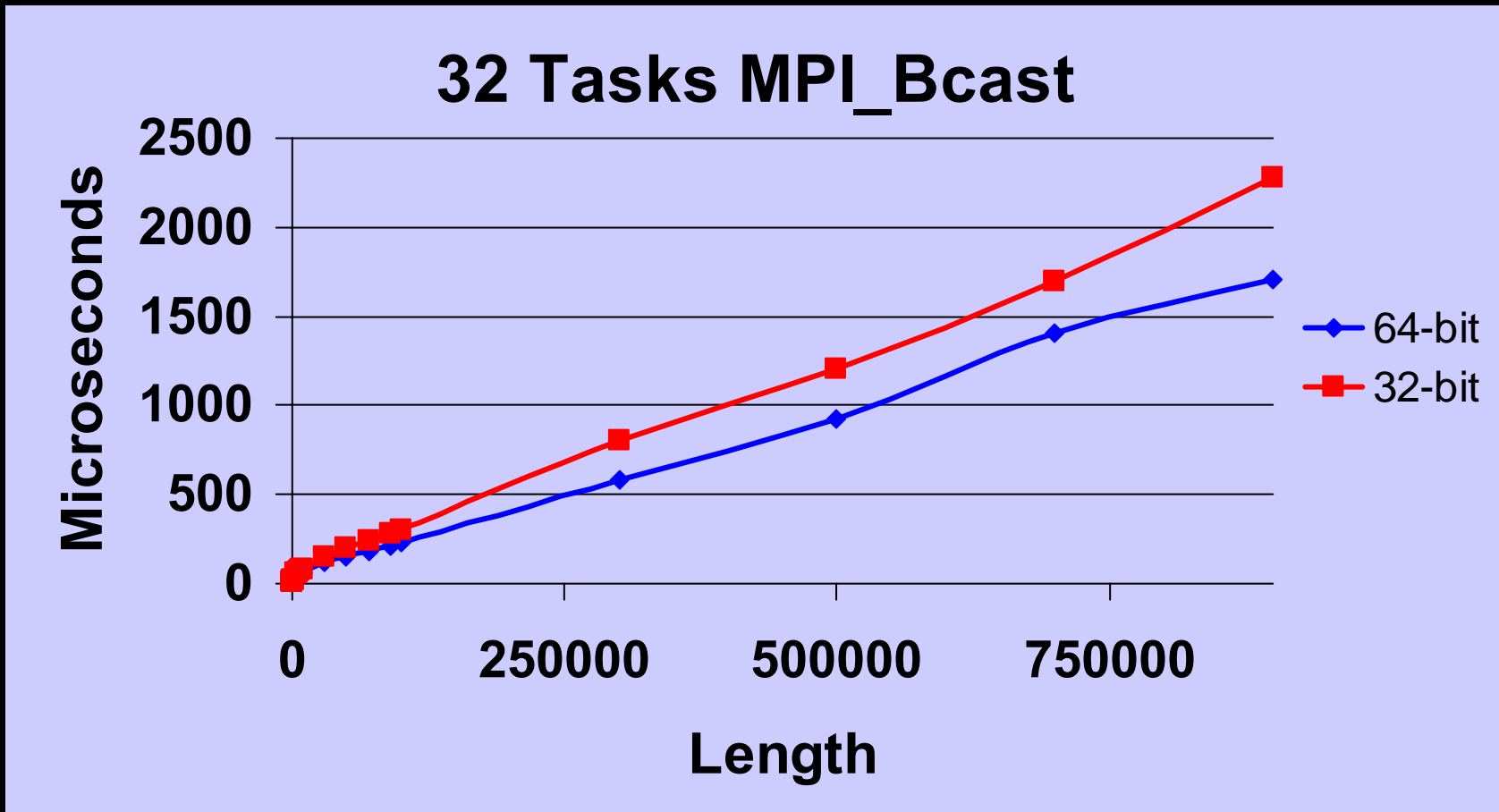
- **Take special advantage of 64-bit addressing**
 - More aggressive algorithms
- **Example:**
 - **MPI_Bcast, 32 tasks: 25% faster with 64-bit addressing**

MPI Bcast: 32-bit vs. 64-bit



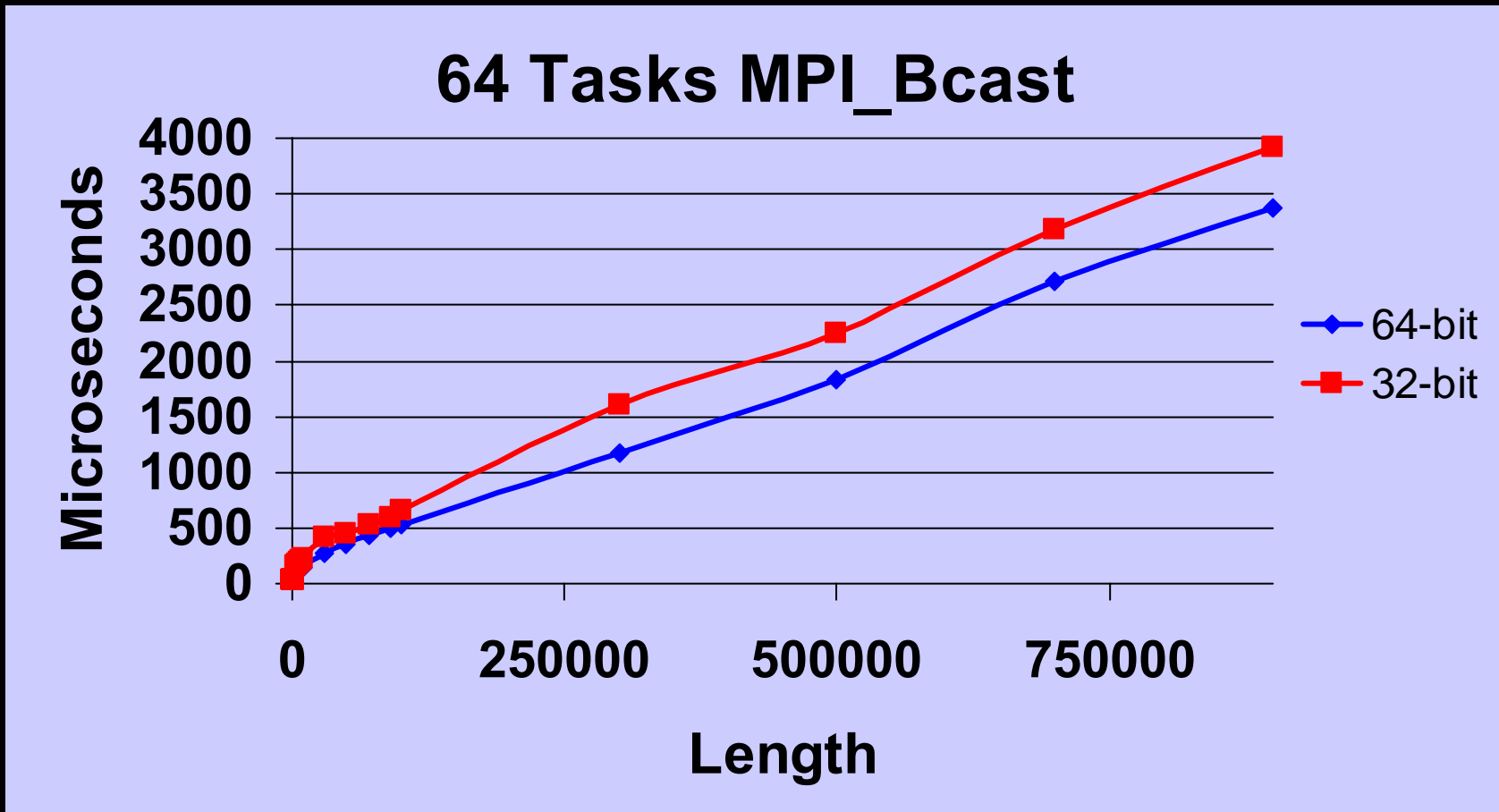
p655 1.5 GHz, HPS, RDMA

MPI_Bcast: 32-bit vs. 64-bit



p655 1.5 GHz, HPS, RDMA

MPI_Bcast: 32-bit vs. 64-bit



p655 1.5 GHz, HPS, RDMA

Application Considerations

- **MPI-LAPI has different architecture than prior version**
 - **Bulk transfer:**
 - Larger messages (>500K are used)
- **Set MP_SINGLE_THREAD=yes (if certain)**
- **32-bit applications:**
 - **Will not use LAPI shared memory for large messages**
 - Convert to 64-bit
 - **Will not use MPI Collective Communication optimizations**
 - Convert to 64-bit.
- **Applications that use signal handlers may require some changes**

MPI Environment Variables

Environment Variable	Recommend Value
MP_EUILIB	us
MP_EUIDEVICE	sn_single, sn_all
MP_SHARED_MEMORY	yes
MP_SINGLE_THREAD	yes*
MP_USE_BULK_XFER	yes
MP_BULK_MIN_MSG_SIZE	150 kbyte (default)

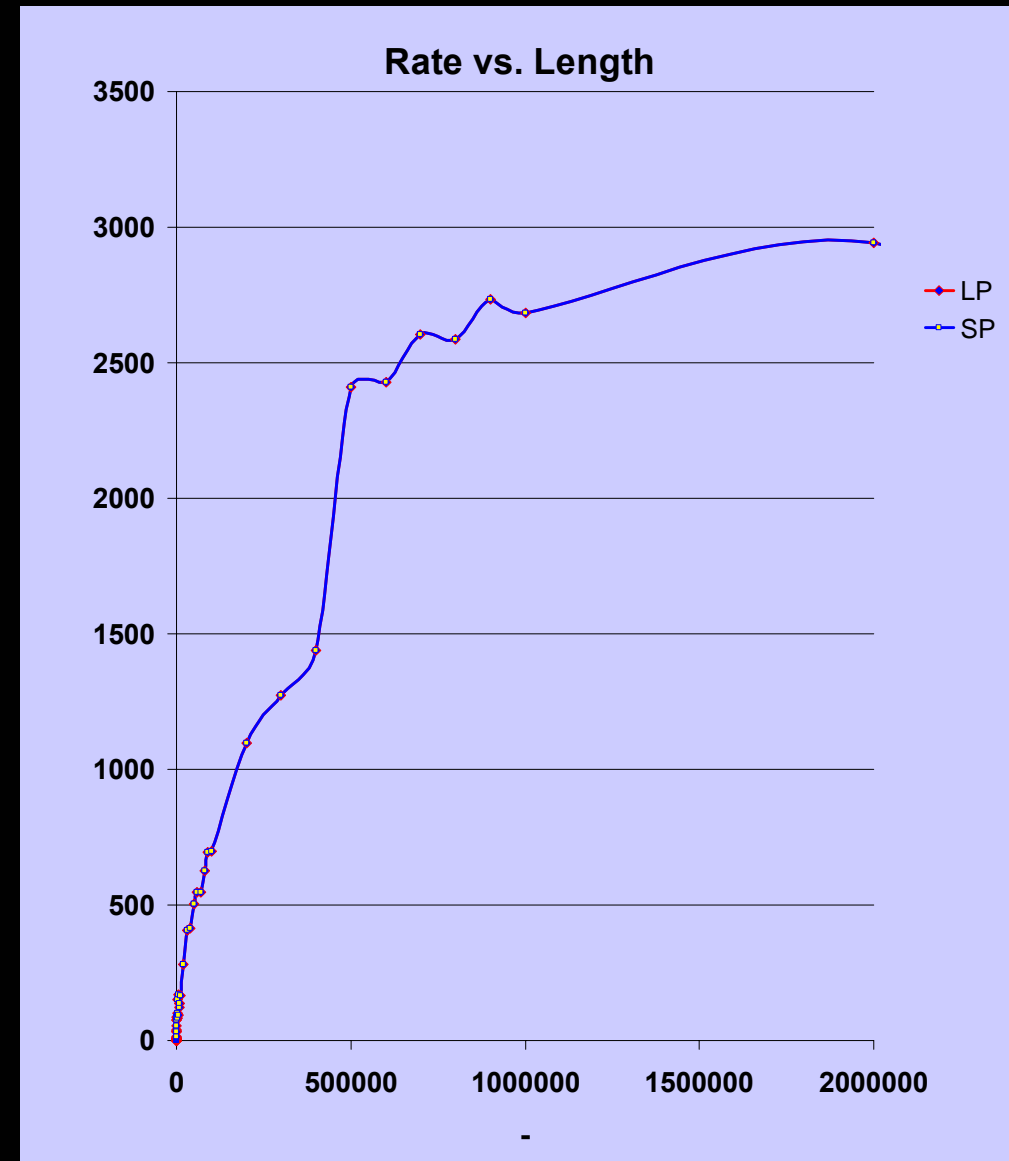
*** If certain**

Bandwidth Structure: Performance Aspects

- **Shared memory**
 - 3 Gbyte/s POWER4
 - 4 Gbyte/s POWER5
- **Large pages**
 - Reduced effect
- **Bulk Transfer**
 - 3 Gbyte/s for two adaptors
- **Eager Limit**
 - 15-20 microsec. → 5-7 microsec
- **Single threaded**
 - 1-2 microsec. reduced latency

HPS Performance Summary

- **High asymptotic peak bandwidth**
 - ~4x vs. Colony
- **Extra “kink” in performance curve**
 - Bulk Transfer
- **Small message performance improvement**
- **Low latency**
 - 5 microsecond.



Prescription For Use of RDMA

- Add to LL configuration file:
 - /usr/lpp/LoadL/full/samples/LoadL_config
 - SCHEDULE_BY_RESOURCES = RDMA
- Verification:
 - Run with MP_INFOLEVEL=2
 - Stderr must contain the text:
 - “Bulk Transfer is enabled”
- Running:
 - LoadLeveler switch
 - # @ bulkxfer = yes

Summary

- **HPS**
 - **Bandwidth**
 - **Bulk Transfers**
 - **4x higher bandwidth (large messages)**
 - **Latency**
 - **<4 microsecond**

Auxiliary