
MPICH-GM and VI-GM middlewares

Patrick Geoffray

Opinionated Software Developer

Myricom, Inc.

patrick@myri.com

Myrinet Users Group Conference

Vienna, Austria

13 May 2002

Current Choice of Myrinet Software Interfaces

- The GM API
 - Low level, but some applications are programmed at this level.
- TCP/IP
 - Actually, an Ethernet emulation over Myrinet.
- MPICH-GM
 - An implementation of the Argonne MPICH directly over GM.
- VI-GM
 - An implementation of the VI Architecture API directly over GM.
- Sockets-GM
 - An implementation of UNIX or Windows sockets (or DCOM) over GM.

The GM API to implement MPI or VI

- **Pros:**
 - Connectionless.
 - OS-bypass.
 - Fair.
 - Reliable.
- **Cons:**
 - Registration overhead.
 - Limited number of receive tokens.
 - Very limited number of send tokens.
 - Match incoming messages and receive buffers on (5 + 1) bits.
 - Reliable.
 - No scatter-gather support.
 - No collective communication support.

Challenge I: registration overhead

- True zero-copy yields high bandwidth and low CPU overhead but cannot provide low latency for small messages.
 - True full-copy provides lowest latency for small messages but wastes CPU cycles and reduces bandwidth for larger packets.
- ➔ Trade-off between the two protocols:
- Memory copies on send/receive sides to/from pre-registered memory areas: **EAGER** protocol (0 -> 16288 Bytes for MPICH-GM).
 - Zero-copy protocol with synchronization and dynamic memory registration: **RENDEZ-VOUS** protocol (16289 Bytes for MPICH-GM).

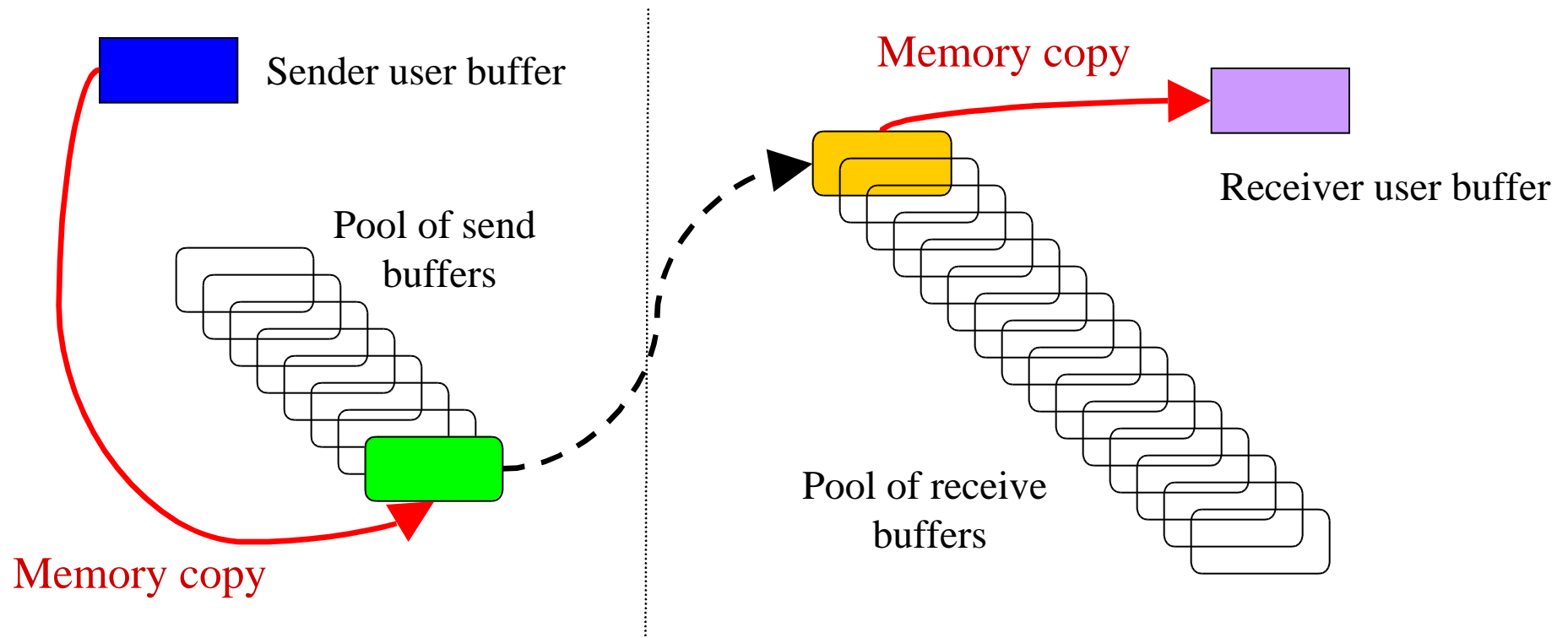
Challenge II: limited GM receive tokens

- In GM, incoming regular messages are delivered to receive buffers in registered user space memory.
- Each posted receive buffer uses a GM receive token.
- Number of posted MPI (asynchronous) Receives is unbounded.
- Number of posted VI Receives is bounded per VI, but much greater than the number of GM receive tokens.

- Impossible to map MPI or VI directly on top of regular GM messages:
 - **EAGER** protocol: regular GM messages.
 - **RENDEZ-VOUS** protocol: Directed sends, i.e. PUT.

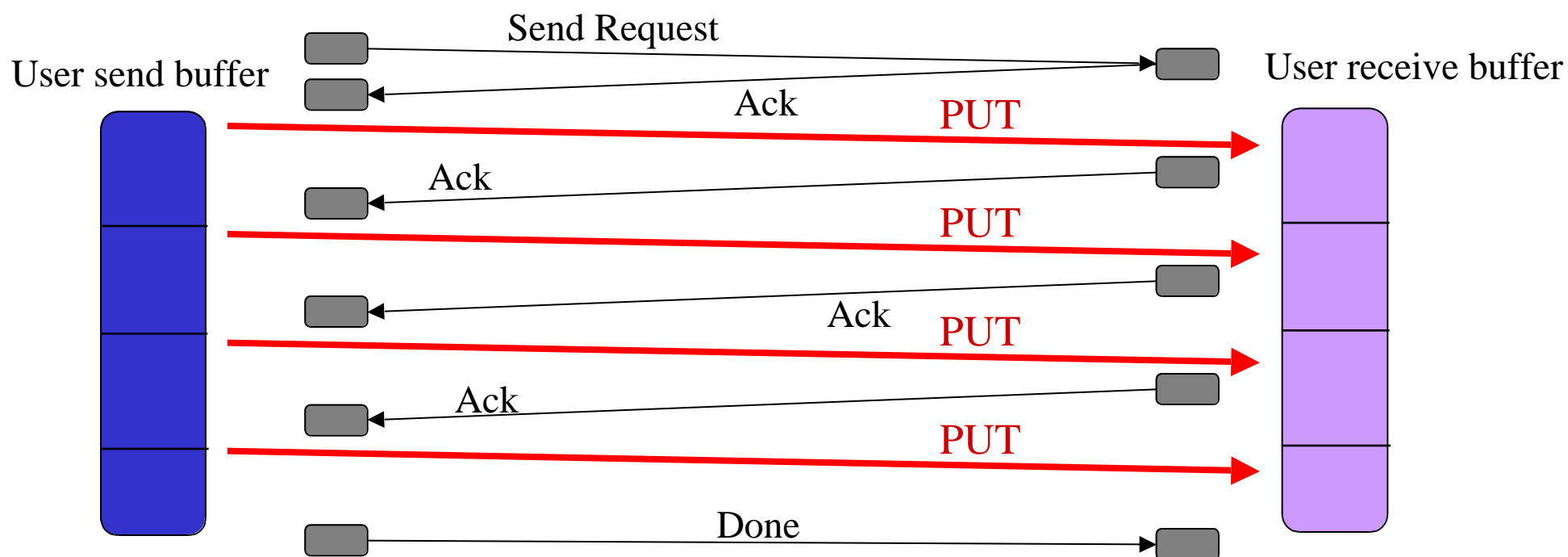
EAGER protocol

- Pools of pre-allocated, pre-registered, aligned and identical send buffers and receive buffers.
- Memory copy to a send buffer, send to posted receive buffer, memory copy to the user-space buffer when appropriate.



RENDEZ-VOUS protocol

- Synchronization via exchange of small messages.
- Memory registration on sender and receiver sides.
- One-sided communication for data.
- Pipelined to overlap registration and communication.

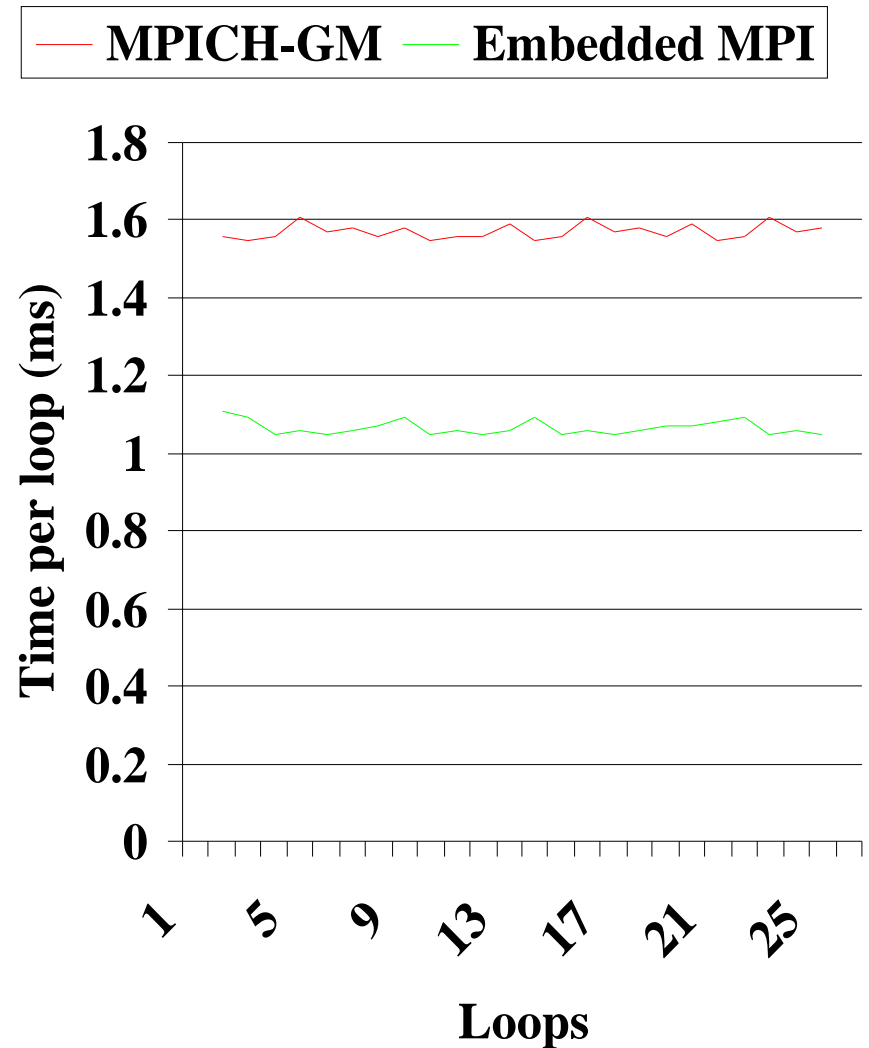


Main MPICH-GM flaw

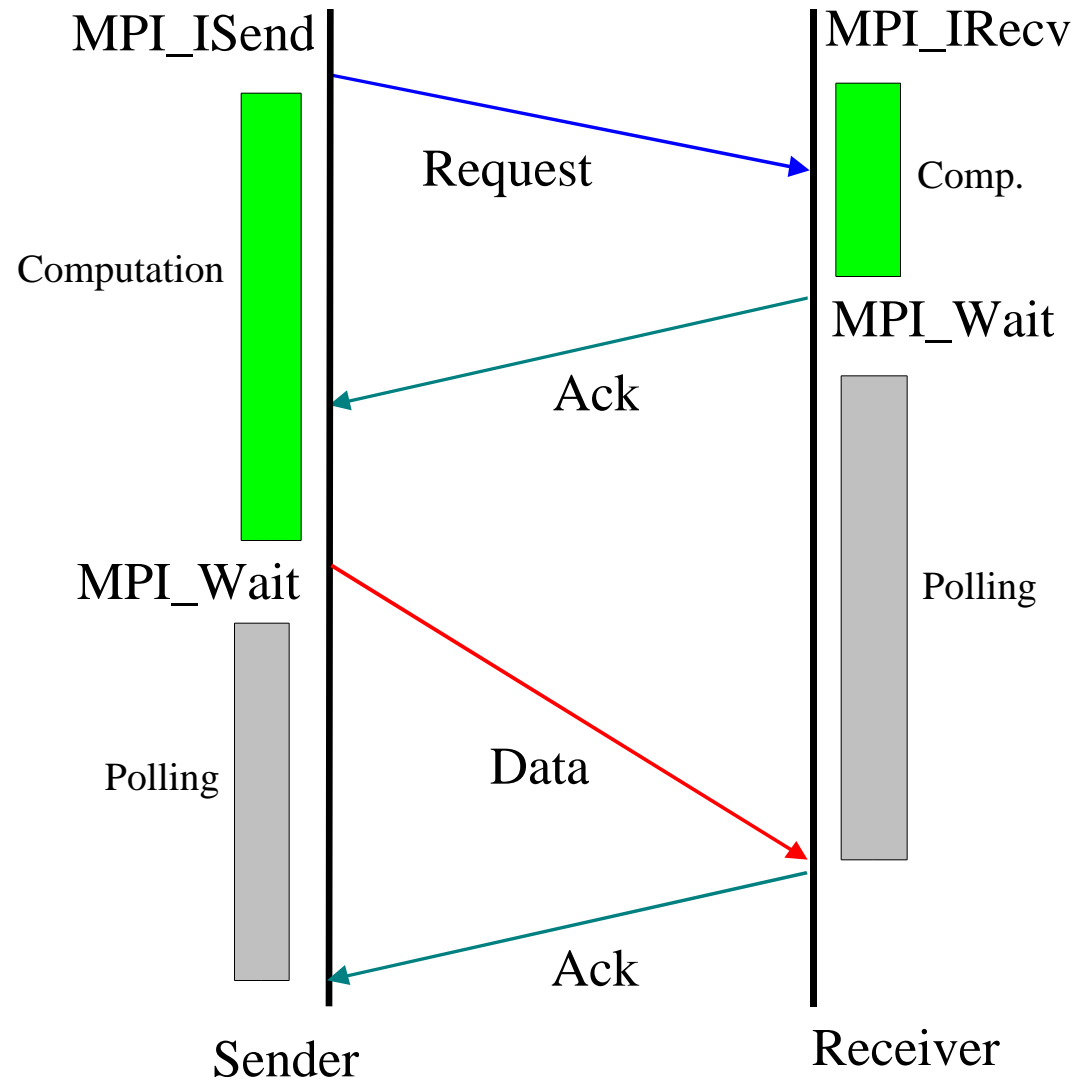
- Sender (loop):
 - MPI_Isend (100 KB).
 - Compute 1 ms.
 - MPI_Wait.
- Receiver (loop):
 - MPI_Irecv (100 KB).
 - Compute 1 ms
 - MPI_Wait.

Overlap is bad.

Most users do not care.



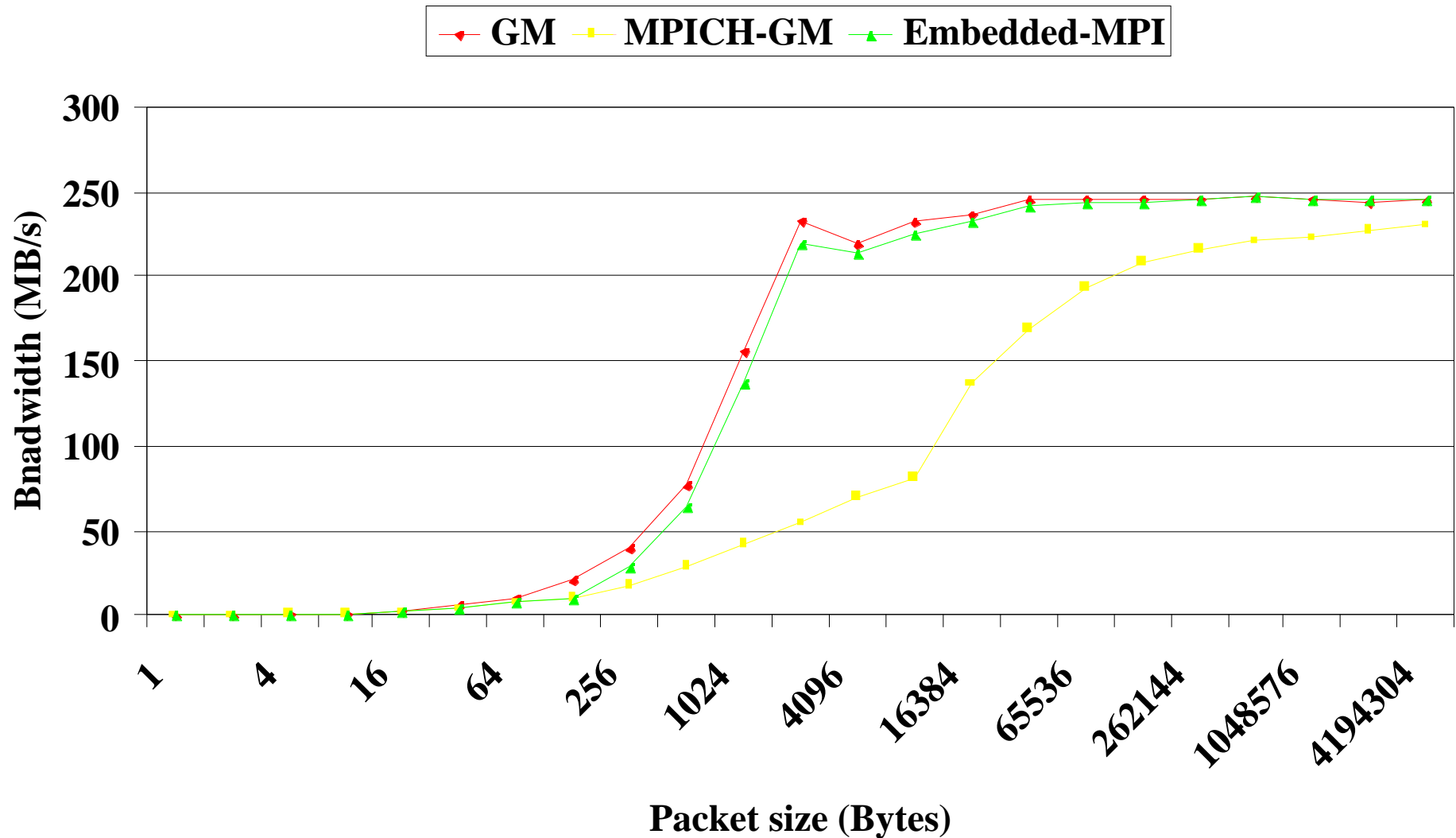
MPI Rendez-vous protocol inefficiency



MPICH-GM

- Port of MPICH from ANL, between Channel Interface and ADI.
- Initially ported by Loic Prylli in 1998, rewritten by Patrick Geoffray in 2001.
- Critical mass of users and applications: stability and portability.
- Short term: merge in Argonne's CVS, dynamic port allocation, MPD support, progression thread.
- Long term: requires firmware support.
 - MPI matching in the NIC (Tag, sender, context).
 - Unexpected messages retrieved by GET.
 - Collective communications in the NIC.
 - Complex datatypes using scatter-gather.
 - MPI specific operation (context allocation) in the NIC.

Very experimental Embedded-MPI results



VI-GM

- Original port of the VI Architecture specifications on top of GM.
- Released in 2002: VI-GM 1.0 Linux/Windows, IA32/IA64.
- Short term: testing and validation by key customers (Oracle, DB2, DAFS, etc.).
- Long term: requires firmware support, will inherit the MPI support in the GM-2 firmware.
 - VI matching (VI, size).
 - Scatter-gather by hardware.
 - Connection protocol assisted by firmware.

Future of middlewares on Myrinet

- Native MPI support in GM-2: “**Embedded MPI**”
- Design pre-requirements are in current GM-2.
- MPI: target MPICH-2 from ANL and/or proprietary MPI.
- VI: depends on customer needs.
- Other middlewares: SHMEM ? Parallel Filesystem ?
- Questions ?