



MPJ/Ibis, a Flexible and Efficient Message Passing Platform for Java

Rob van Nieuwpoort
rob@cs.vu.nl



MPI

- Message Passing Interface
- Language independent specification
- Language bindings
 - C, C++, Fortran, ...
- High performance
- Available for many platforms
- Widely used



MPI operations

```
MPI_Send(buf, BUFSIZE, MPI_CHAR, dest, TAG,  
         MPI_COMM_WORLD);
```

```
MPI_Recv(buf, BUFSIZE, MPI_CHAR, from, TAG,  
         MPI_COMM_WORLD, &status);
```

- Point-to-point
 - Send / receive (only explicit!)
 - Synchronous / asynchronous
- Collective operations
 - broadcast, reduce, scatter, gather, ...
- Closed world



MPI bindings for Java

- Many Java/MPI bindings:
 - JavaMPI, JMPI, MPIJ, CCJ, etc.
- MPJ: Proposed by the Java Grande Forum
 - A Java language binding for MPI 1.1
 - Developed benchmark suite
- Implementations:
 - MPIJava, built on top of native MPI library
 - MPJ/Ibis, built on top of Ibis



MPJ

```
void Comm.send(Object buf, int offset, int count,  
               Datatype type, int dest, int tag)  
    throws MPJException
```

- No status objects, but exceptions
- Separate versions for primitive types
- Parameter “buf” can be
 - Array of a primitive type
 - Array of objects
 - Multidimensional arrays
 - Arbitrarily complex data structure -> object serialization



MPJ/Ibis

- First 100% Java MPJ implementation
- Uses Ibis IPL for communication
- Ibis provides highly efficient object serialization
- Special grid connectivity support in Ibis
 - Heterogeneous networks
 - Communicate through firewalls
- Very portable, ideal for grid computing

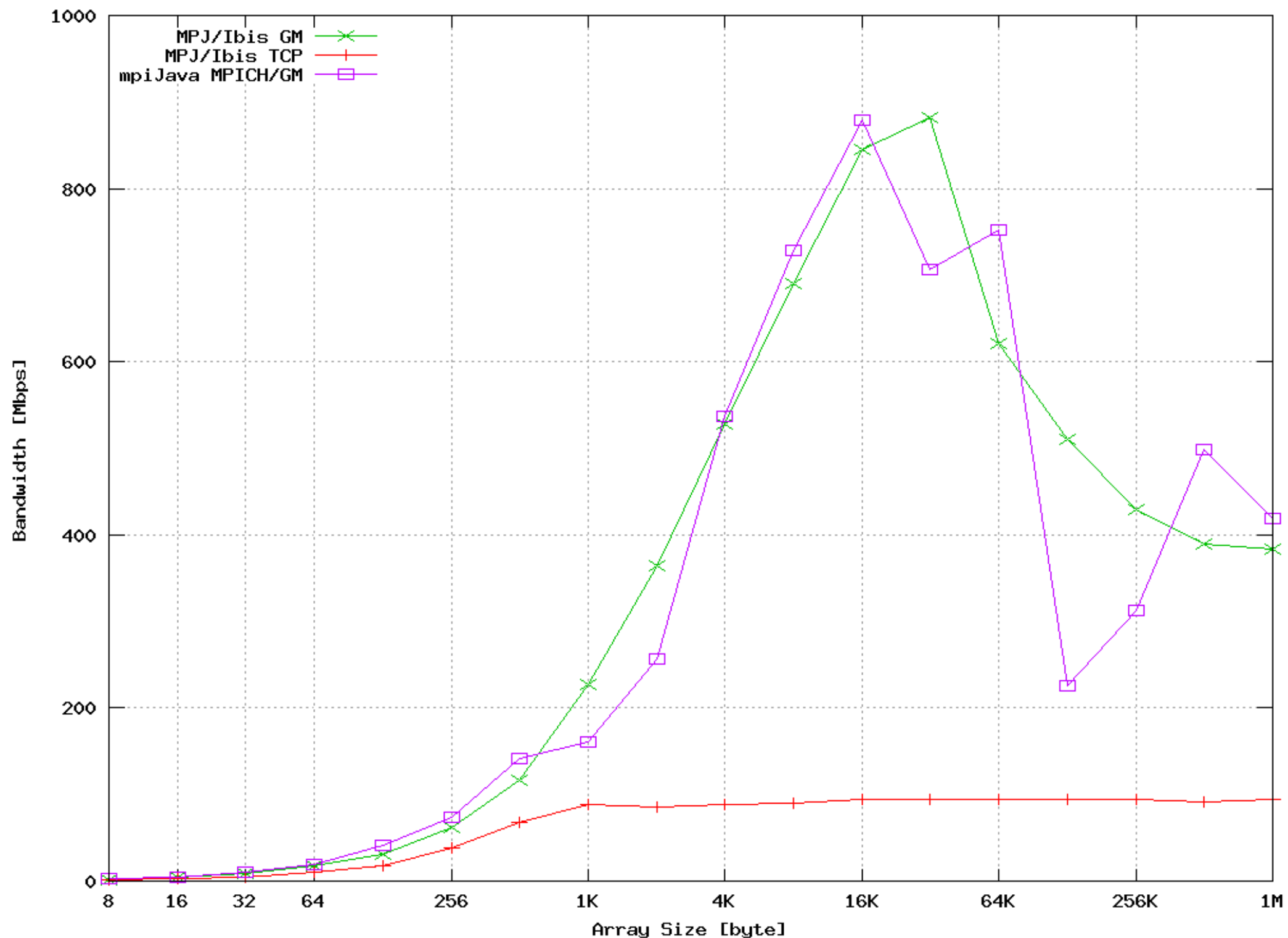


MPJ/Ibis latency P-III 1 GHz

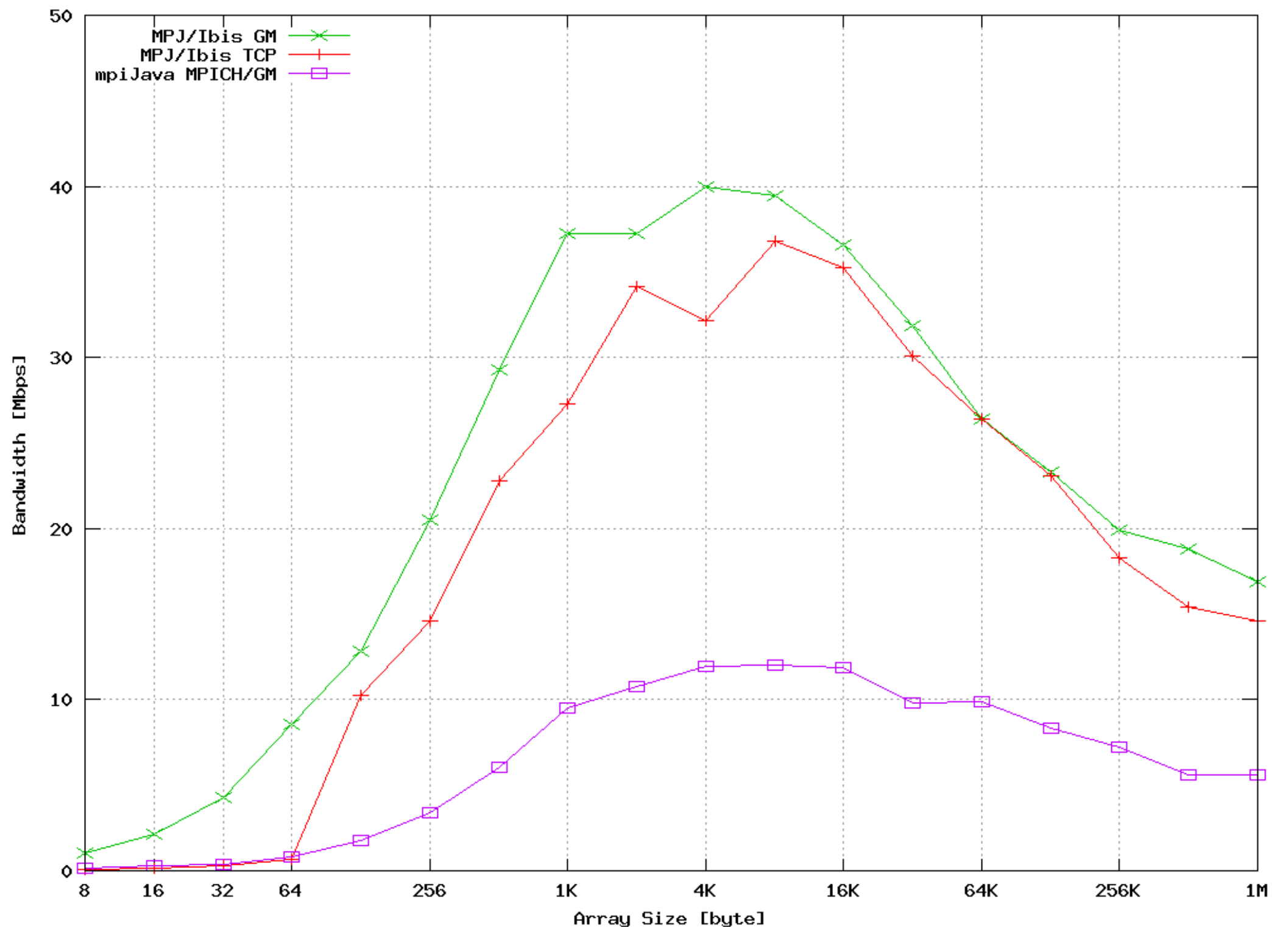
Implementation	round-trip latency (us)
MYRINET	
mpijava (MPICH 1.2.6/GM)	28
Ibis (GM)	44
MPJ/Ibis (GM)	50
FAST ETHERNET	
Ibis (TCP)	113
MPJ/Ibis (TCP)	120



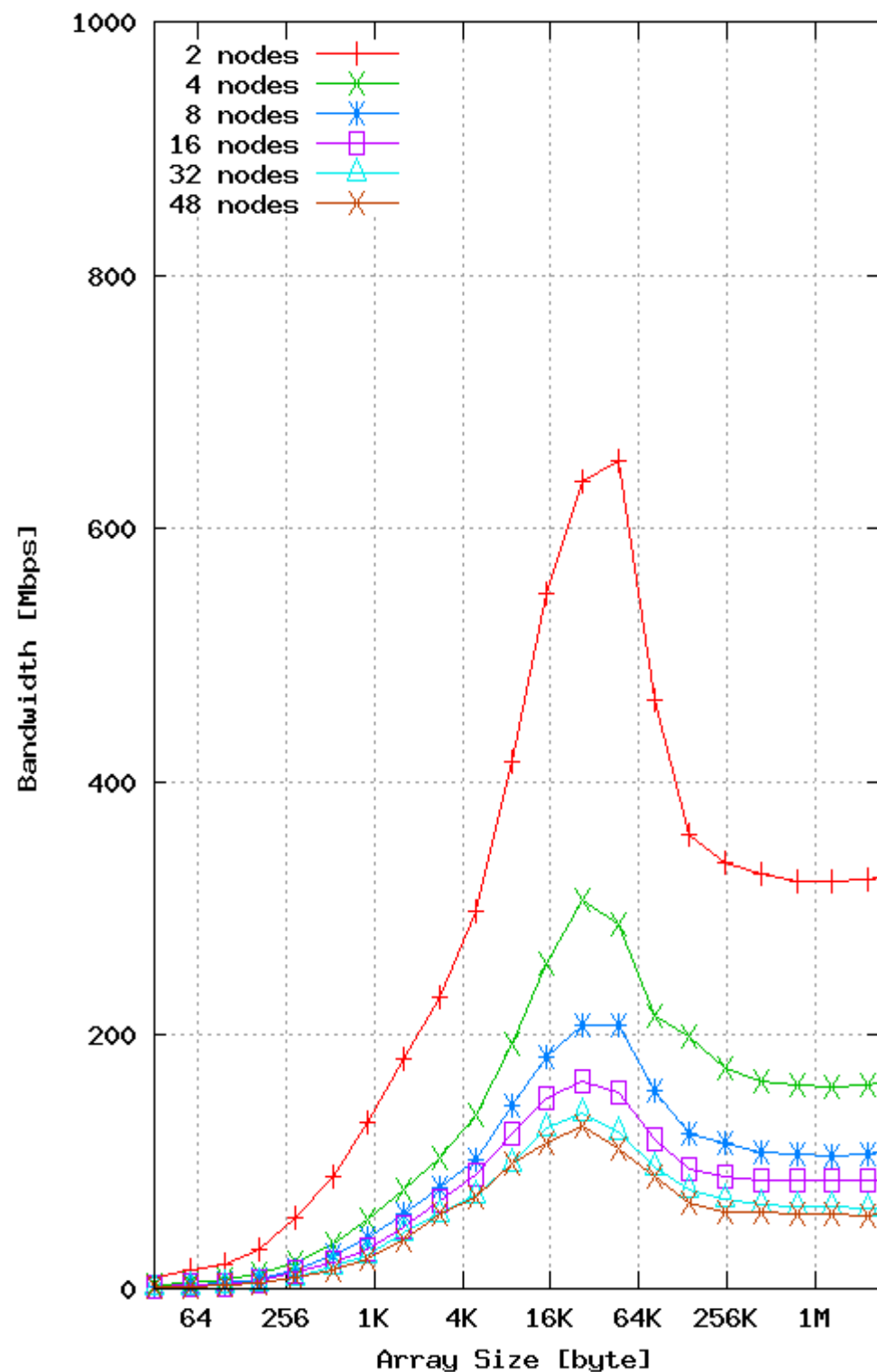
Throughput Double Arrays



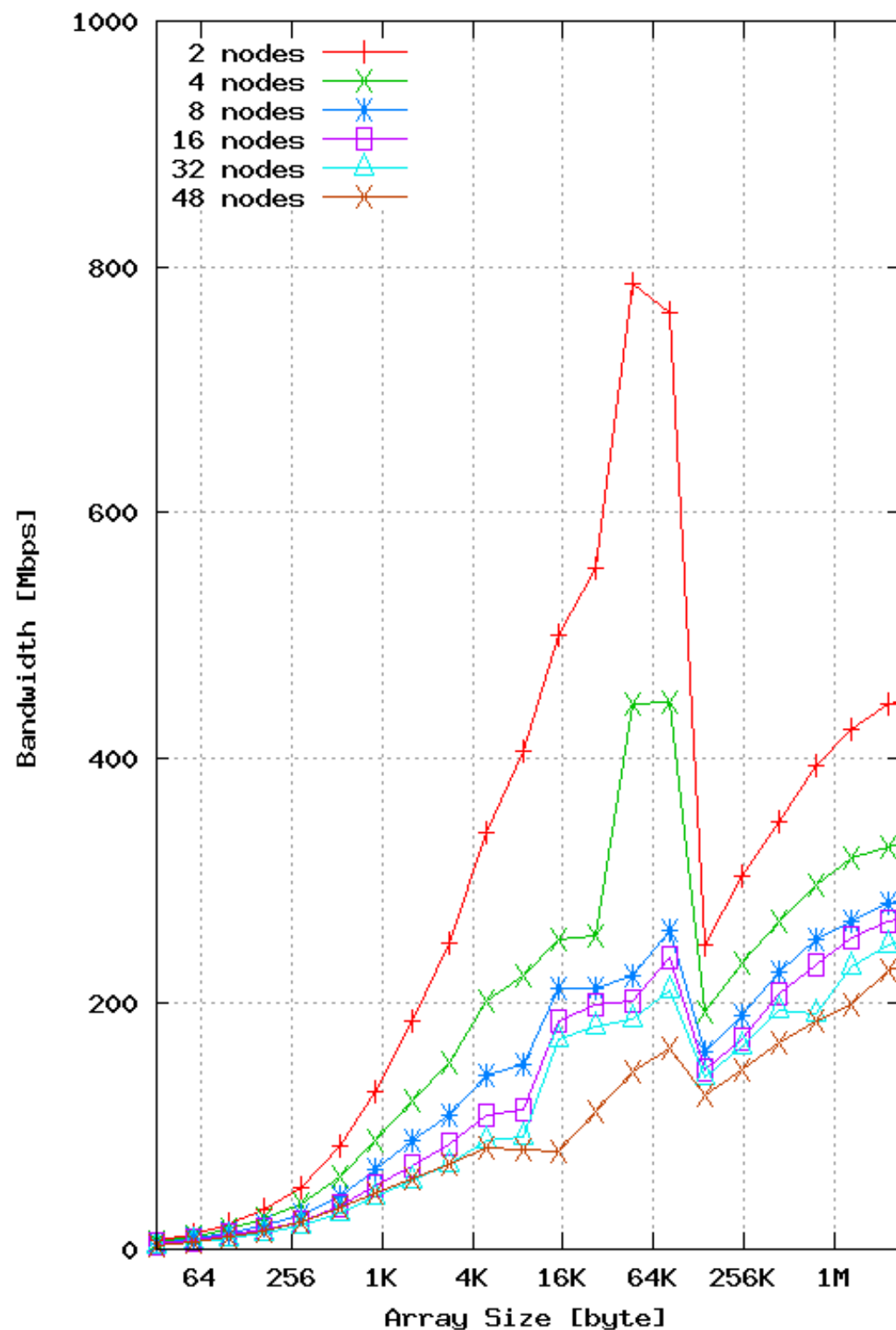
Throughput Object Arrays



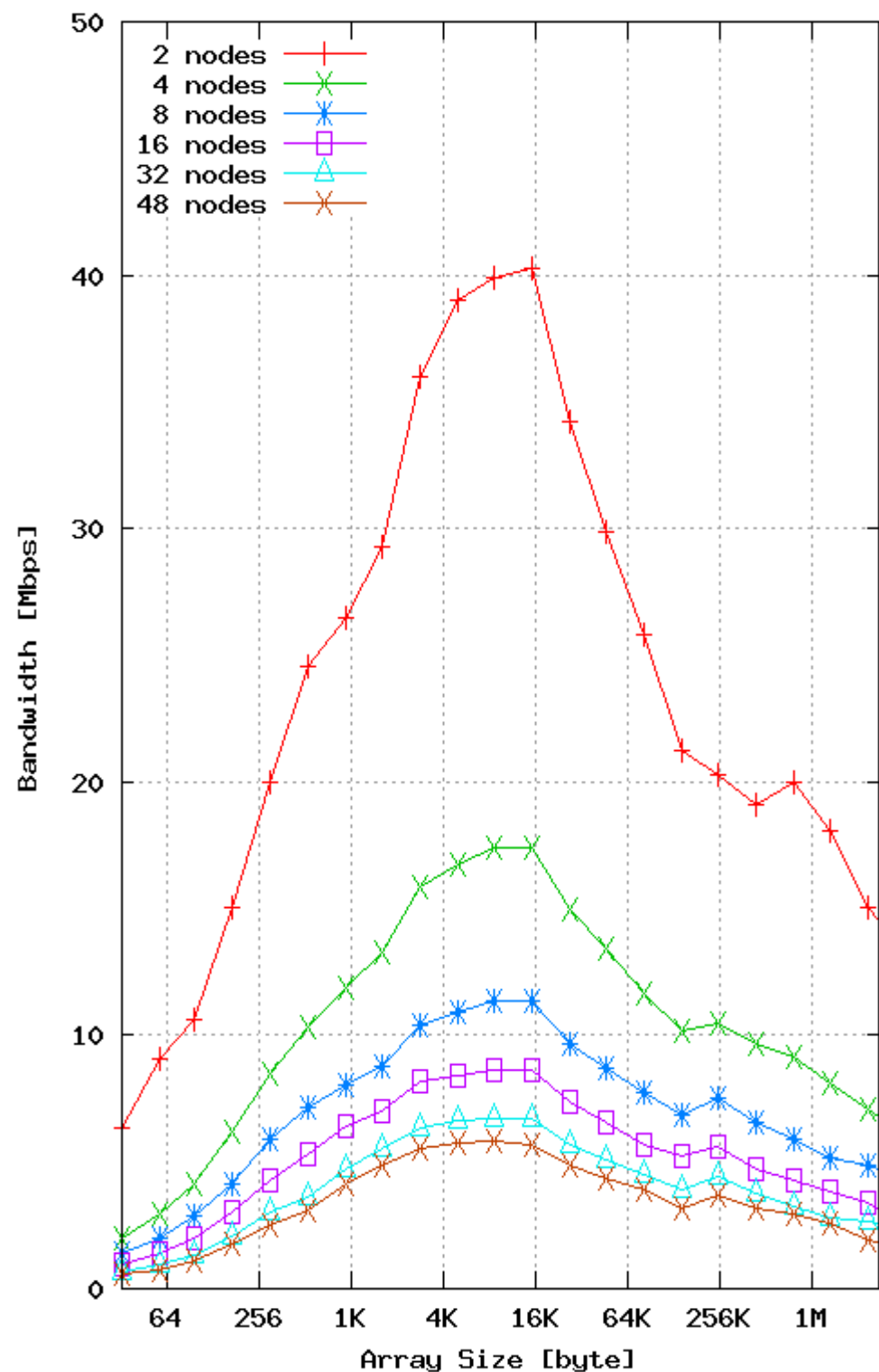
MPJ/Ibis GM (Double arrays)



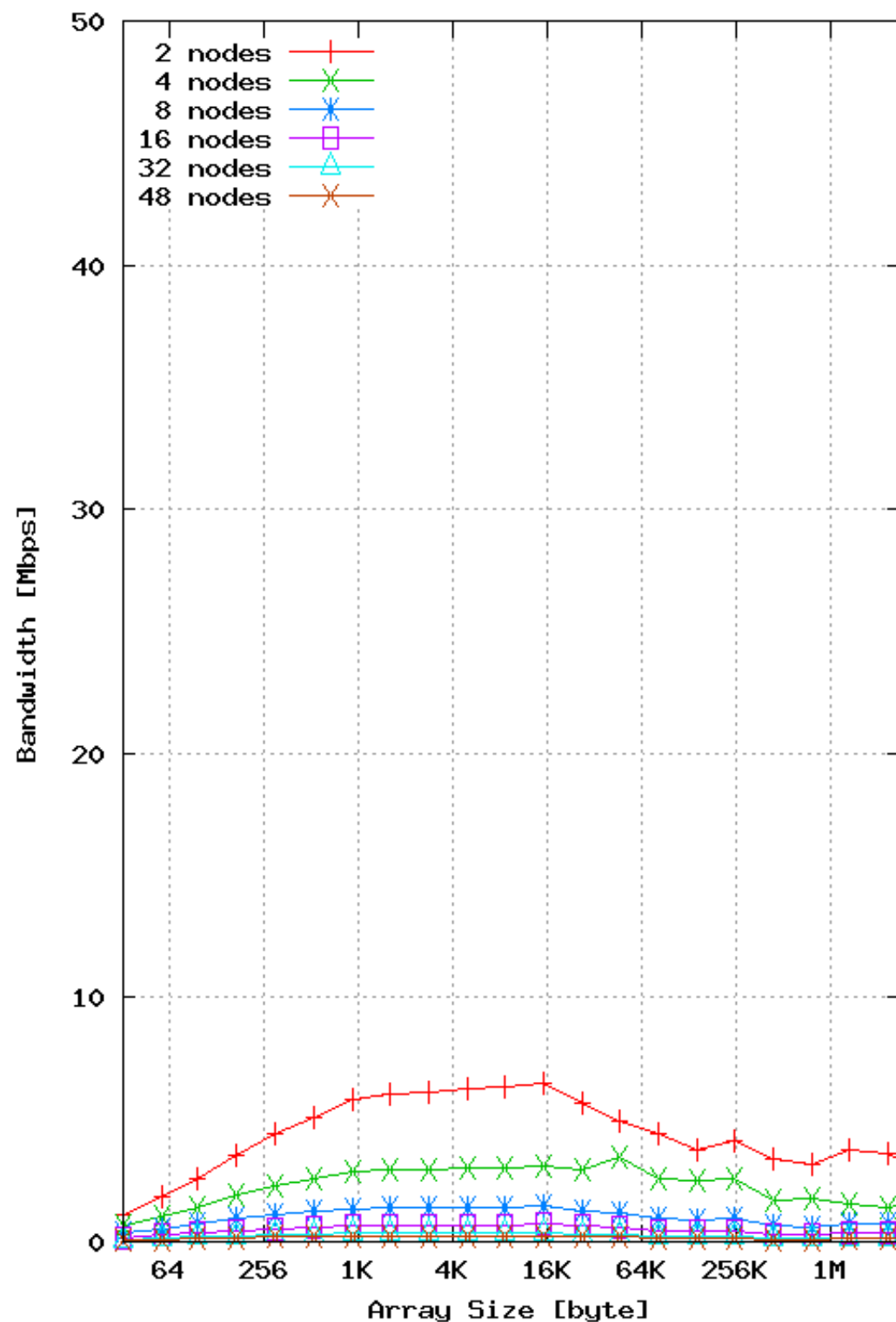
mpiJava MPICH/GM (Double arrays)



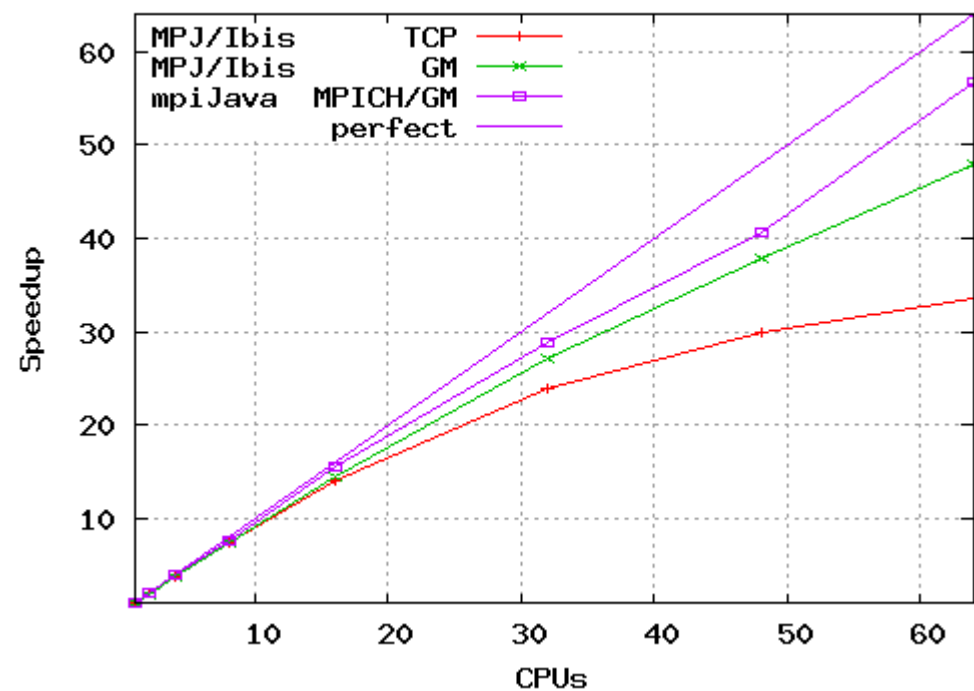
MPJ/Ibis GM (Object arrays)



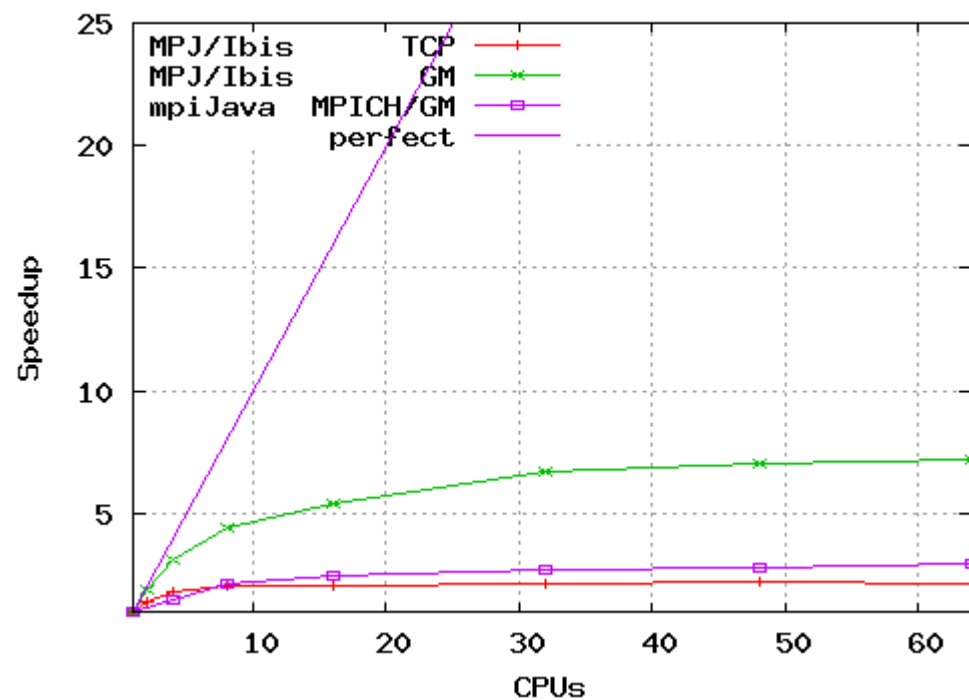
mpiJava MPICH/GM (Object arrays)



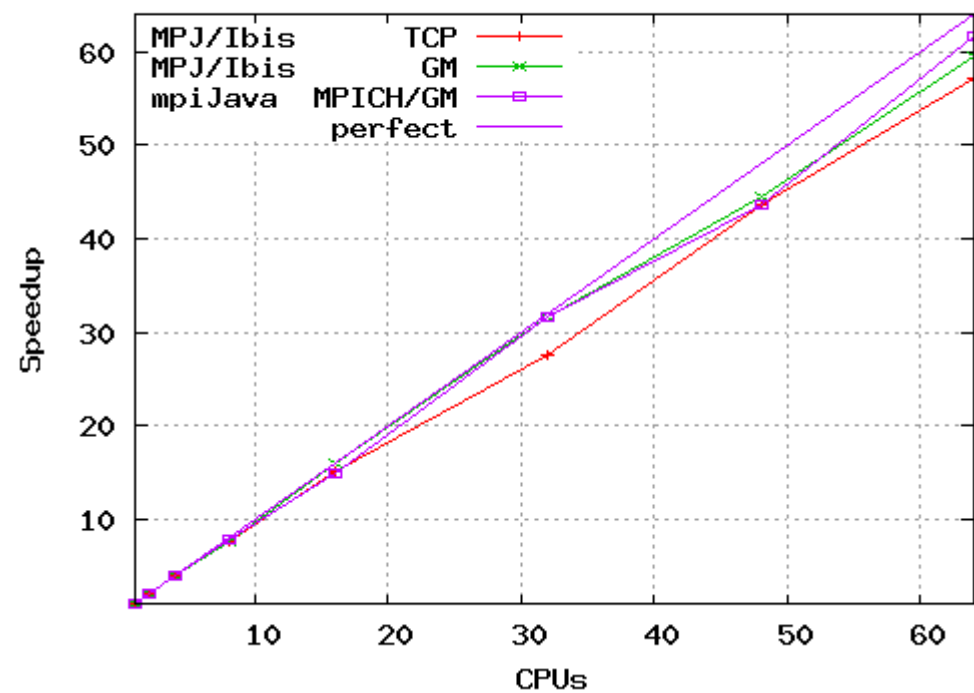
Molecular Dynamics



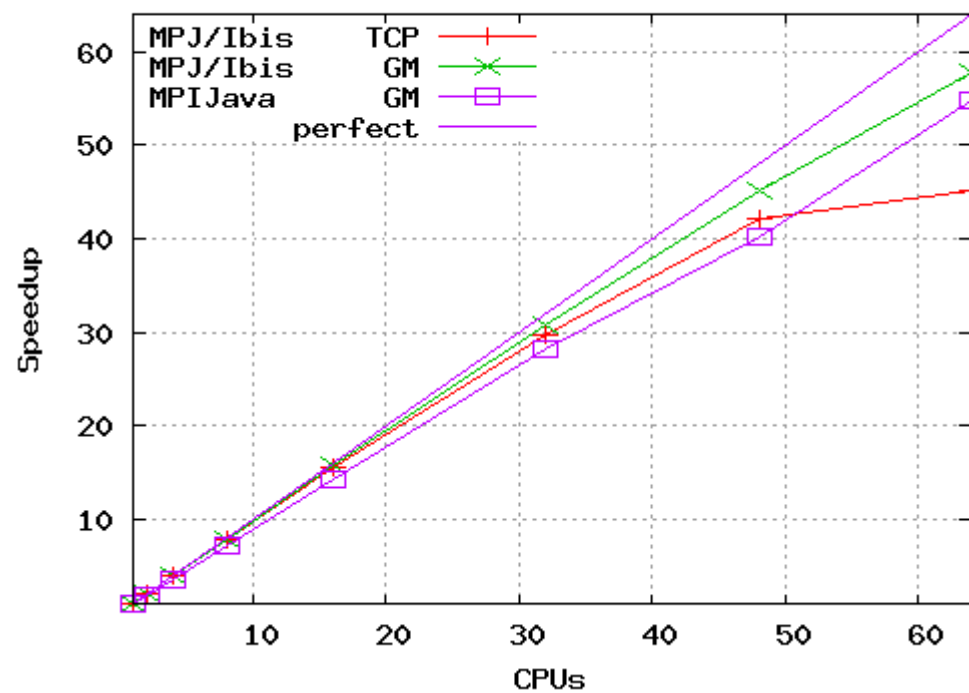
MonteCarlo



RayTracer



ASP



Conclusions

- Targeted at grid environments
- MPJ/Ibis is extremely flexible
 - “run everywhere”
 - Heterogeneous networks
 - Communicate through Firewalls
- Competitive performance
 - Latency and Collectives are a bit slower than native implementation
 - Object serialization is much faster
 - Application-level performance is similar



MPJ/Ibis collectives

Collective Operation	Algorithm	Upper Complexity Borders
<i>allgather</i>	double ring	$O(n)$
<i>allgatherv</i>	single ring	$O(n)$
<i>allreduce</i>	recursive doubling	$O((\log n) + 2)$
<i>alltoall</i>	flat tree	$O(n^2)$
<i>alltoallv</i>	flat tree	$O(n^2)$
<i>barrier</i>	flat tree	$O(2n)$
<i>broadcast</i>	binomial tree	$O(\log n)$
<i>gather</i>	flat tree	$O(n)$
<i>gatherv</i>	flat tree	$O(n)$
<i>reduce</i>	commutative op: binomial tree non-commutative op: flat tree	$O(\log n)$ $O(n)$
<i>reduceScatter</i>	phase 1: reduce phase 2: scatterv	commutative op: $O((\log n) + n)$ non-commutative op: $O(2n)$
<i>scan</i>	flat tree	$O(n)$
<i>scatter</i>	flat tree	$O(n)$
<i>scatterv</i>	flat tree	$O(n)$



Ibis RMI

Rob van Nieuwpoort
rob@cs.vu.nl



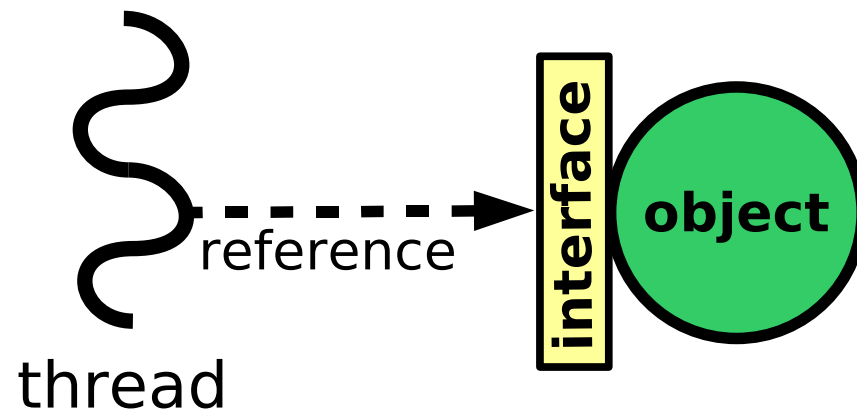
vrije Universiteit



vl·e

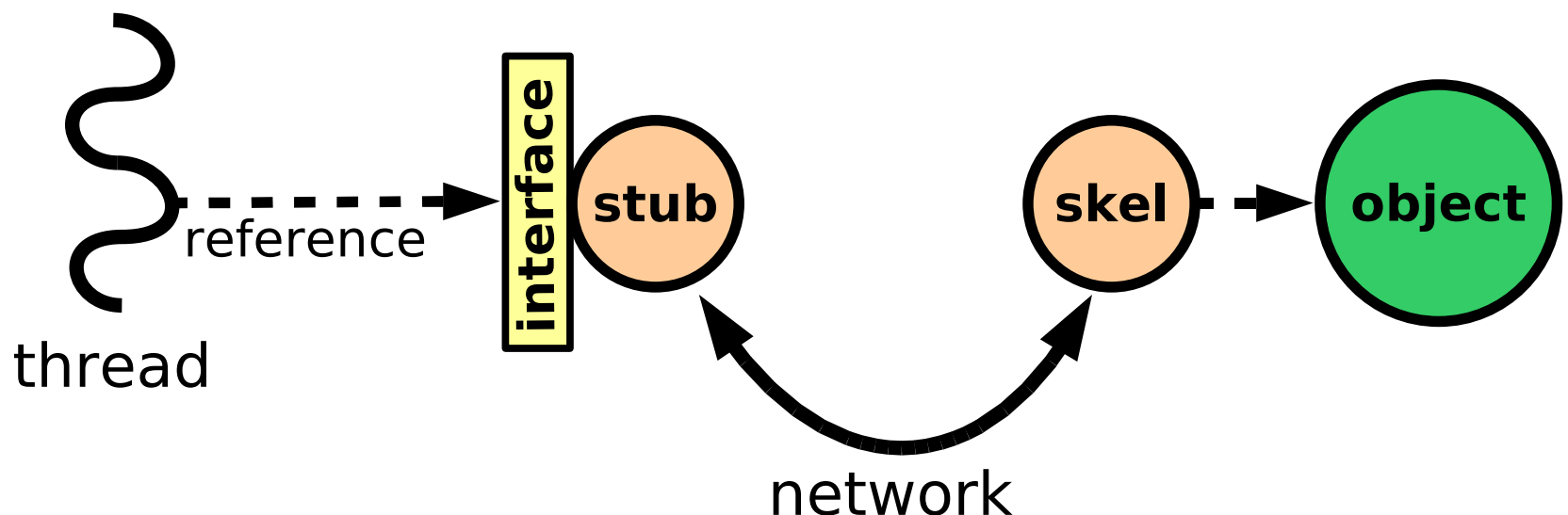
Ibis RMI

- Drop-in replacement for Sun RMI
 - Has the same interface
 - Used different stub compiler (rmic)
 - Generates Ibis specific stubs/skeletons



Ibis RMI

- Drop-in replacement for Sun RMI
 - Has the same interface
 - Used different stub compiler (rmic)
 - Generates Ibis specific stubs/skeletons



Ibis RMI Limitations

- Not interoperable with Sun RMI
 - uses a different protocol
- No socket factories
 - Ibis doesn't have to use sockets!
- No activatable objects



Ibis RMI Advantages

- Uses Ibis / SmartSockets
 - Can communicate through firewalls
 - Does not suffer from NAT / multihoming problems
 - Can use fast networks transparently (Myrinet, etc)
 - Uses Ibis serialization
- Much faster!



MPJ/Ibis structure

