Flow control: a prerequisite for Ethernet scaling

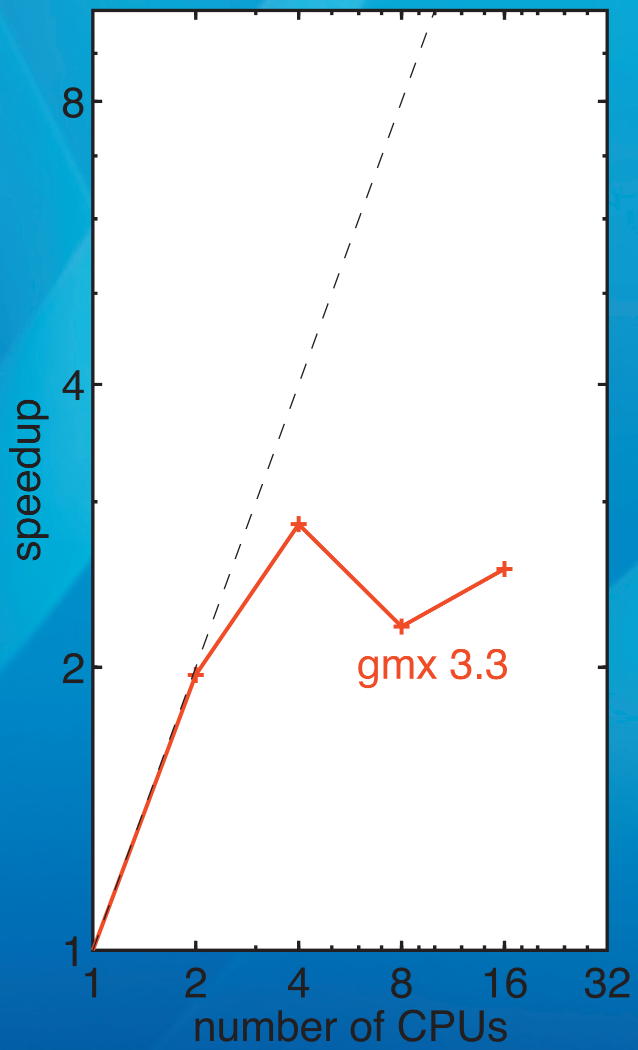
Carsten Kutzner, David van der Spoel, Erik Lindahl,
Martin Fechner, Bert L. de Groot, Helmut Grubmüller

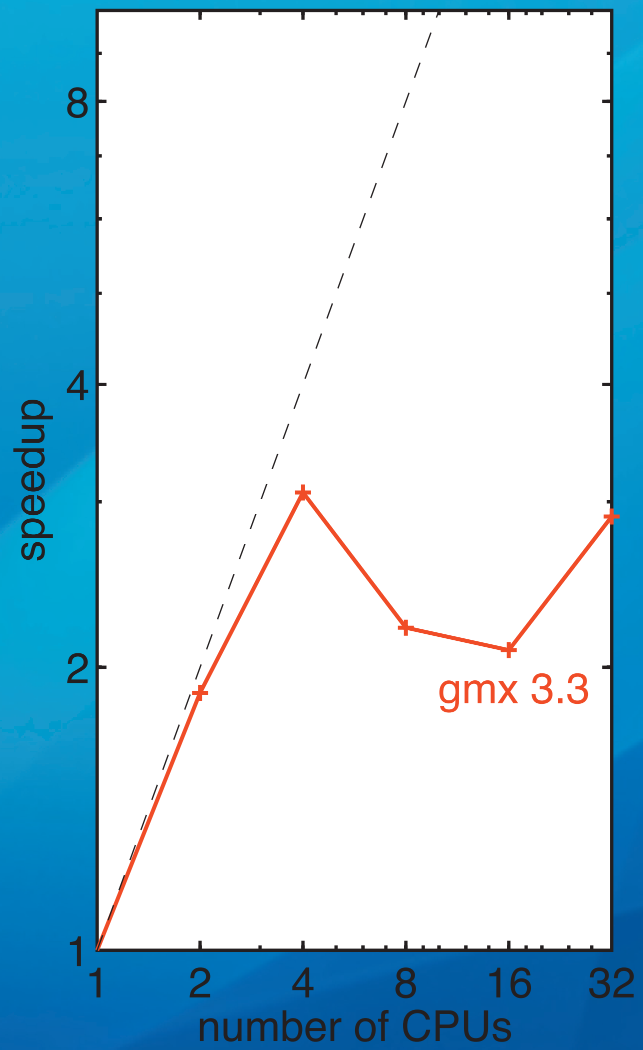
- GROMACS and network congestion
- a network stress-test with MPI_Alltoall
- IEEE 802.3x = ???
- high-level flow control / ordered communication

GROMACS 3.3 GigE speedup

Aquaporin-1 benchmark (80k atoms)

1 CPU/node



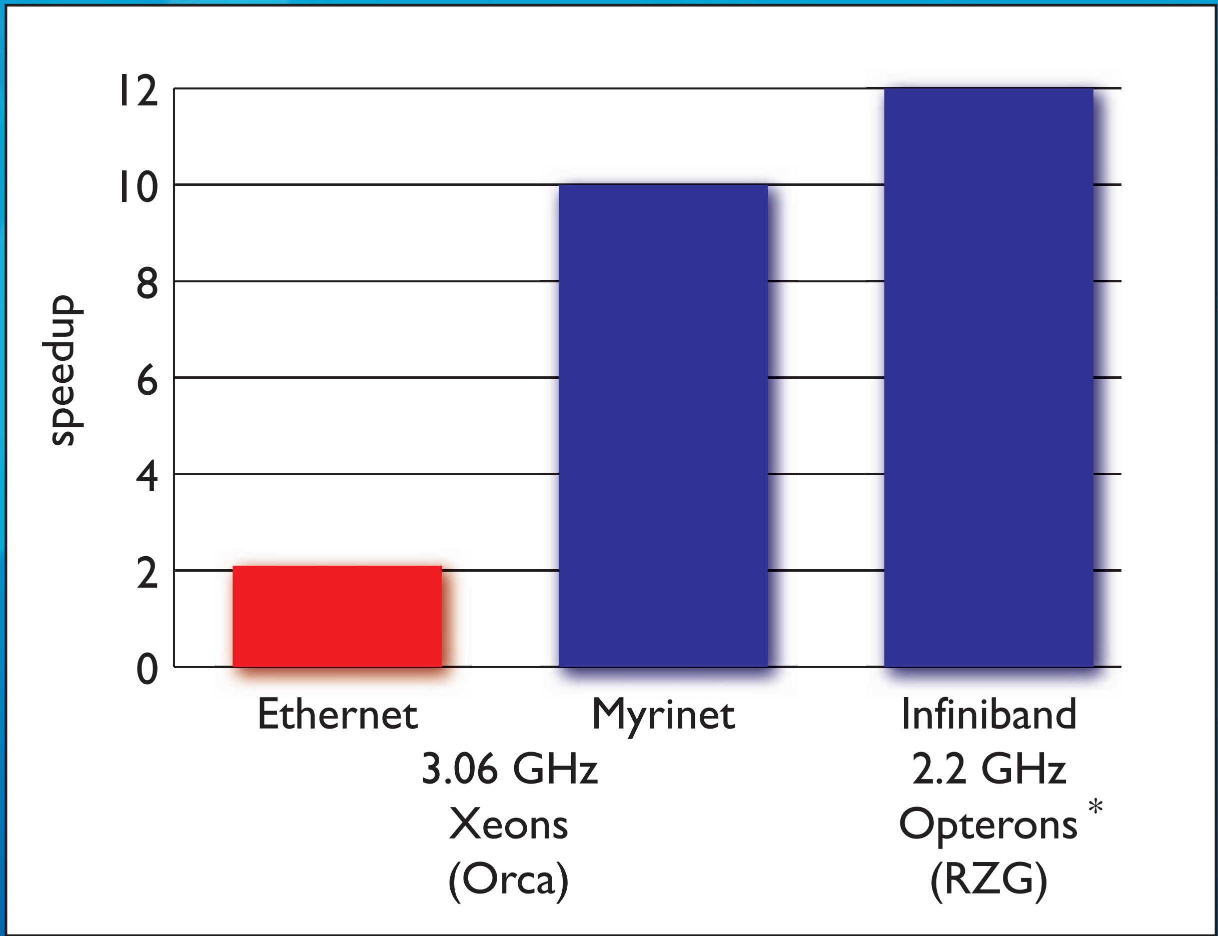


2 CPUs/node

Orca,
Coral,
Slomo,
...

GROMACS 3.3 GigE speedup

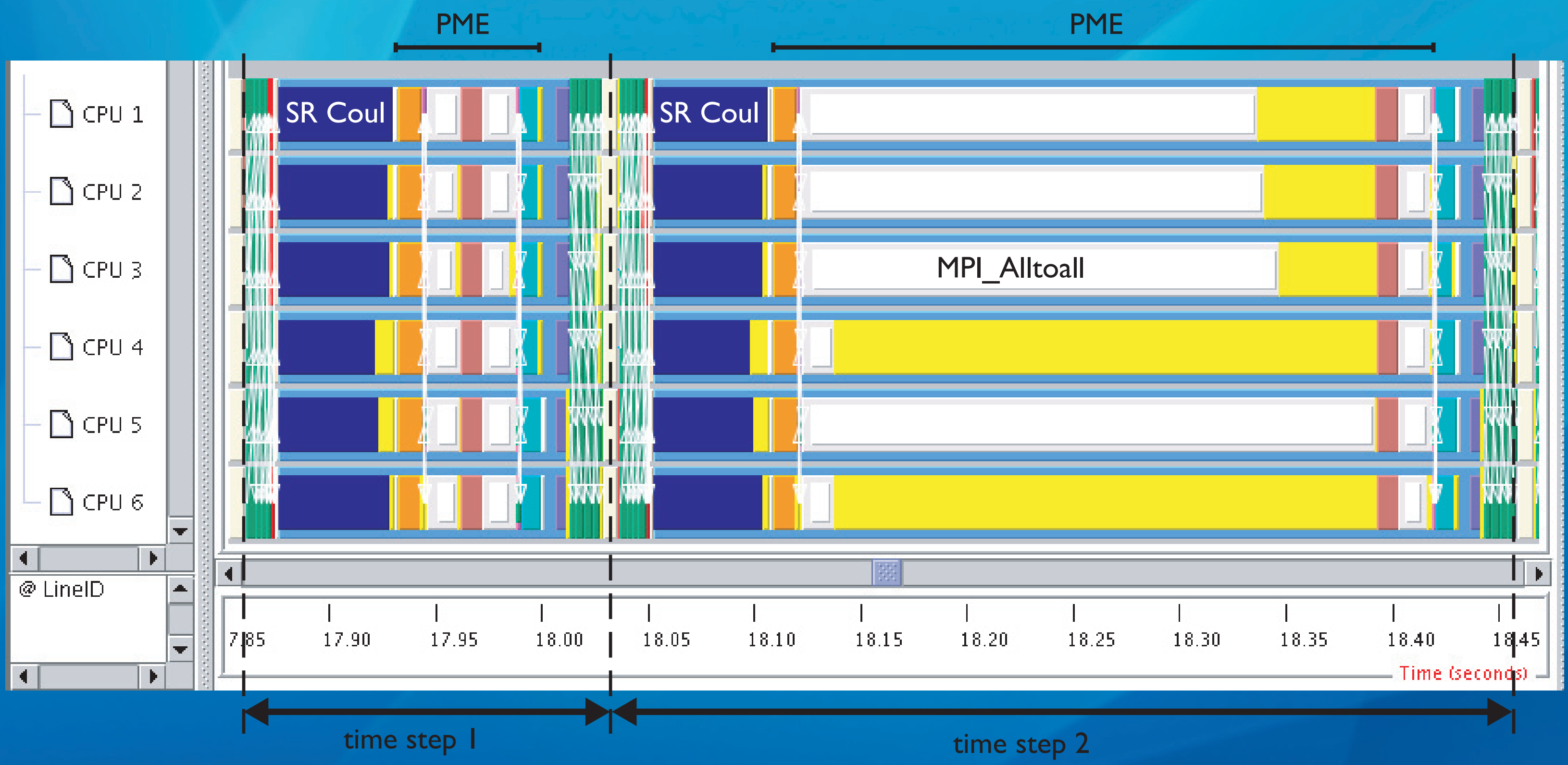
compared to other interconnects (8x2 CPUs)



* benchmark done by Renate Dohmen, RZG

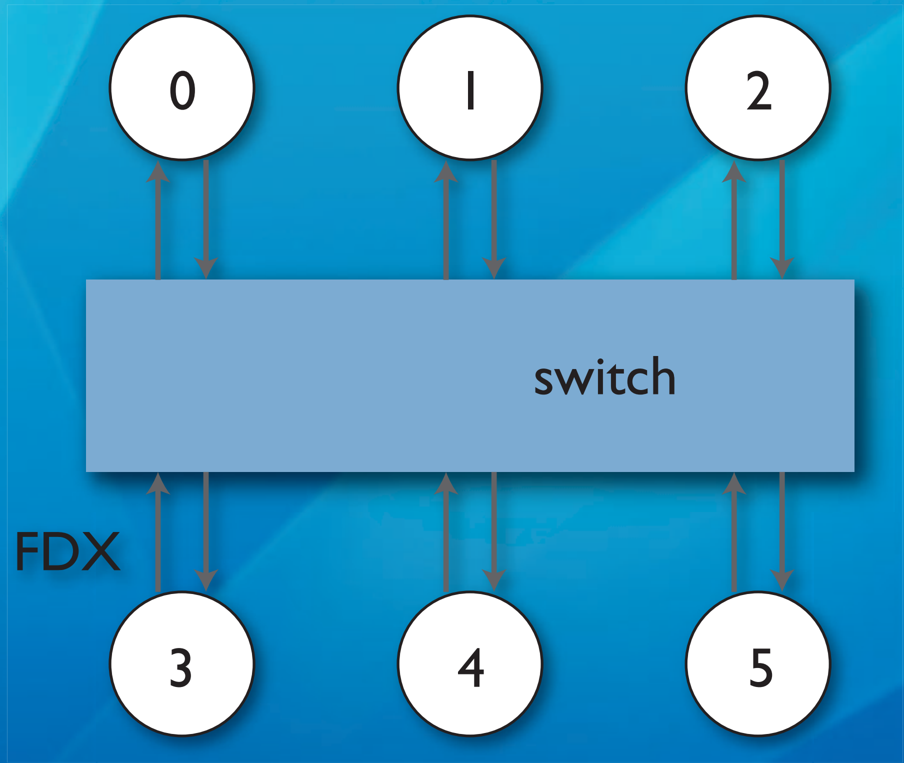
Where is the problem?

- delays of 0.25s in comm-intensive program parts (MPI_Alltoall, MPI_Allreduce, MPI_Bcast, consecutive MPI_Sendrecvs)



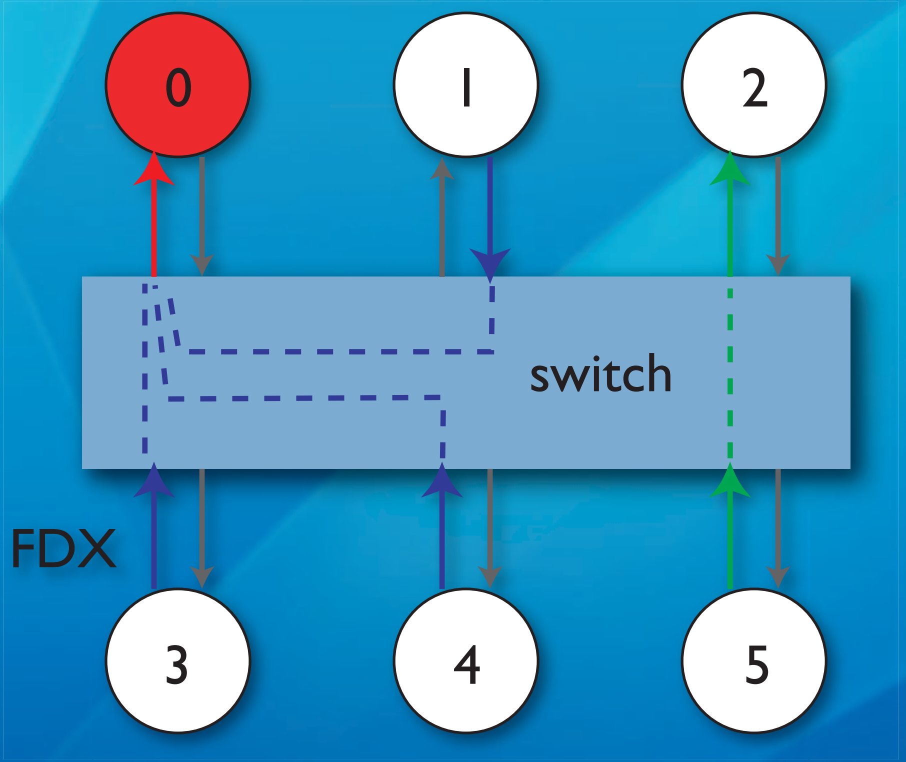
Network congestion

- If more packets arrive than a receiver can process, packets are dropped (and later retransmitted)



Network congestion

- If more packets arrive than a receiver can process, packets are dropped (and later retransmitted)

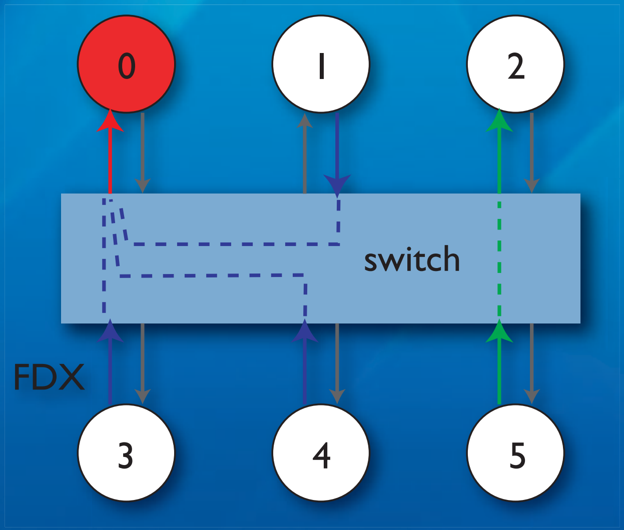


common MPI_Alltoall implementations

order of message transfers is random!

LAM:

```
/* Initiate all send/recv to/from others. */
for (i loops over processes)
{
  MPI_Recv_init(from i ...);
  MPI_Send_init(to i ...);
}
/* Start all the requests. */
MPI_Startall(...);
/* Wait for them all. */
MPI_Waitall(...);
```

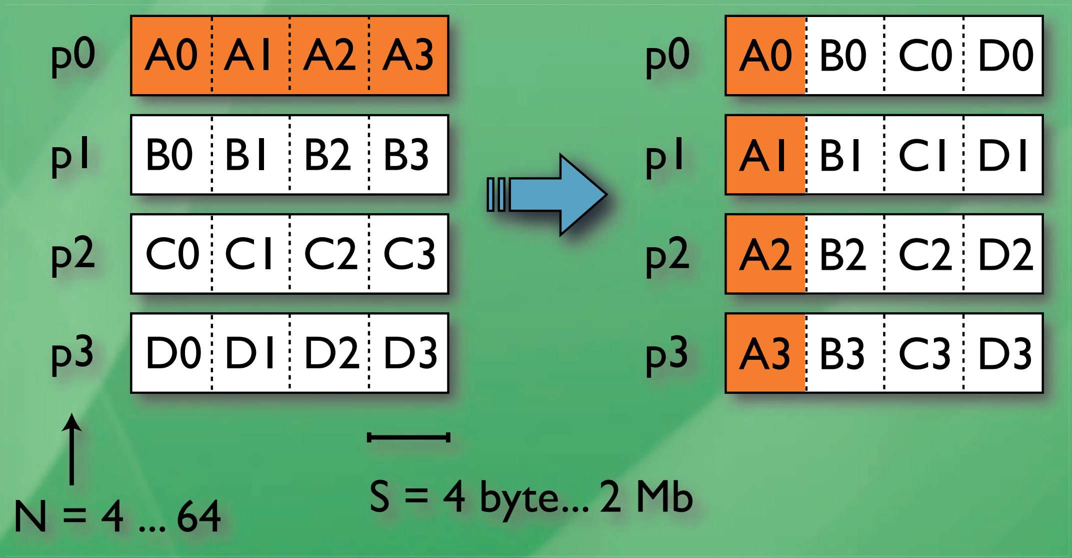


MPICH:

```
/* do the communication -- post *all* sends and re-
ceives: */
for ( i=0; i<size; i++ )
{
/* Performance fix sent in by Duncan Grove
<duncan@cs.adelaide.edu.au>. Instead of posting
irecvs and isends from rank=0 to size-1, scatter the
destinations so that messages don't all go to rank 0
first. Thanks Duncan! */
  dest = (rank+i) % size;
  MPI_Irecv(from dest)
}
for ( i=0; i<size; i++ )
{
  dest = (rank+i) % size;
  MPI_Isend(to dest)
}
/* ... then wait for *all* of them to finish: */
MPI_Waitall(...);
```

A network stress-test with MPI_Alltoall

- MPI_Alltoall, a parallel matrix transpose

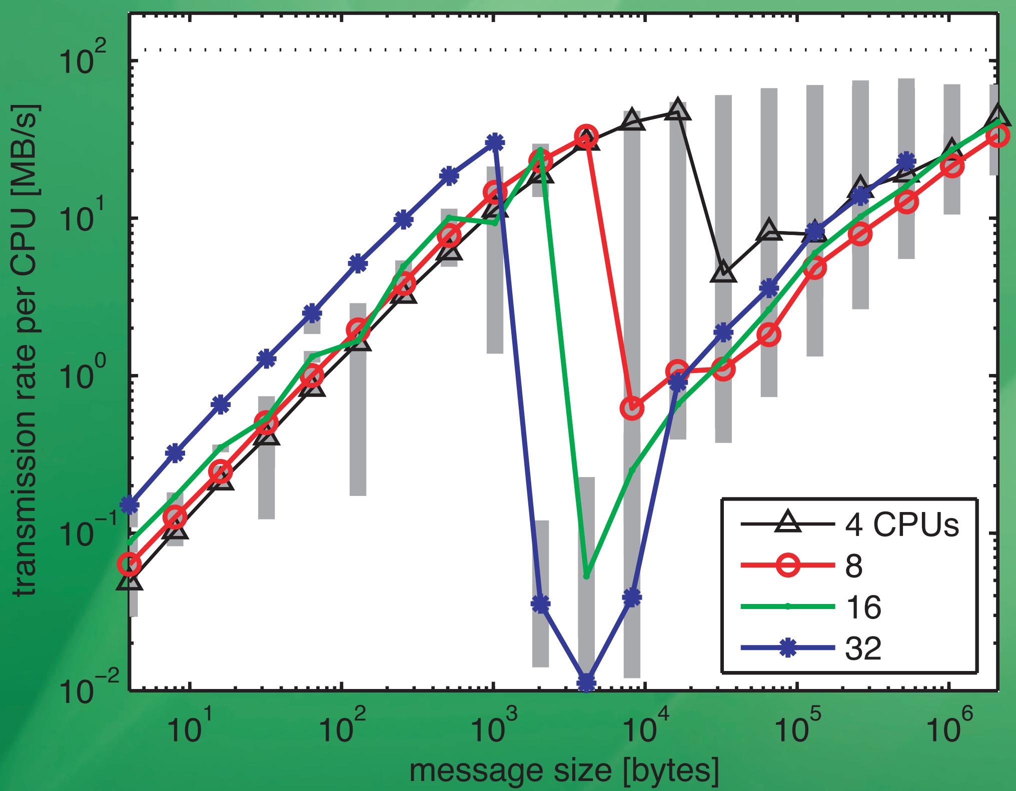


- each of N procs sends N-1 messages of size S
- N*(N-1) messages transferred in time Δt
- transmission rate T per proc: $T = (N-1)S / \Delta t$

LAM MPI_Alltoall

T averaged over 25 calls (bars: min/max)

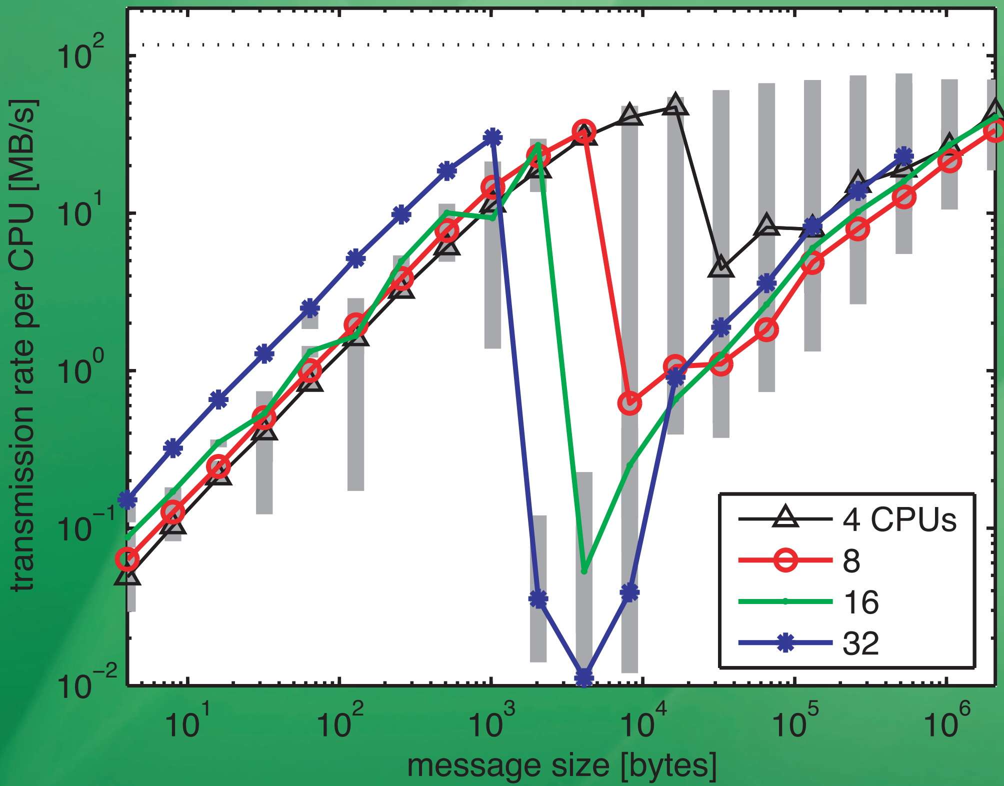
I CPU/node



LAM MPI_Alltoall

T averaged over 25 calls (bars: min/max)

1 CPU/node



data transferred within the FFTW all-to-all:

Aquaporin-1, PME grid 90x88x80=633600 floats

CPUs	byte to each CPU
2	649440
4	173184
6	73800
8	43296
16	11808
32	2952

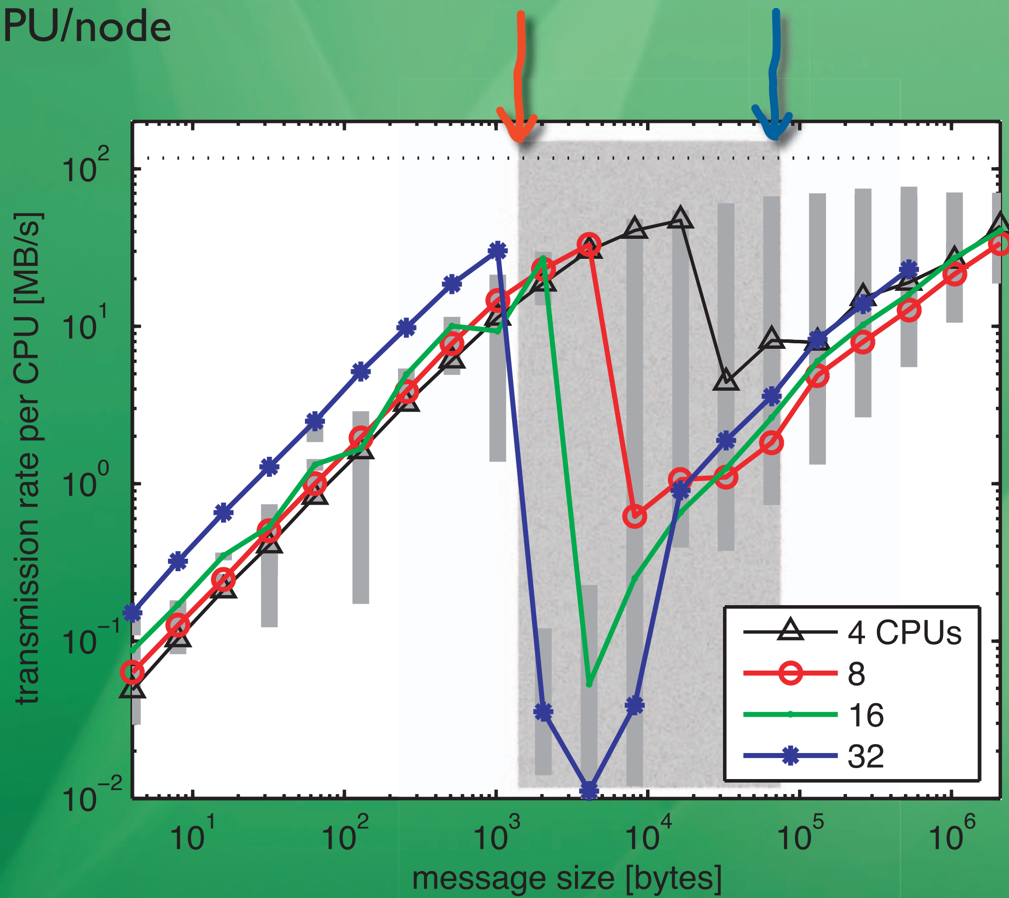
Guanylin, PME grid 30x30x21=18900 floats

CPUs	byte to each CPU
2	19800
4	5632
6	2200
8	1408

LAM MPI_Alltoall

T averaged over 25 calls (bars: min/max)

1 CPU/node



data transferred within the FFTW all-to-all:

Aquaporin-1, PME grid
90x88x80=633600 floats

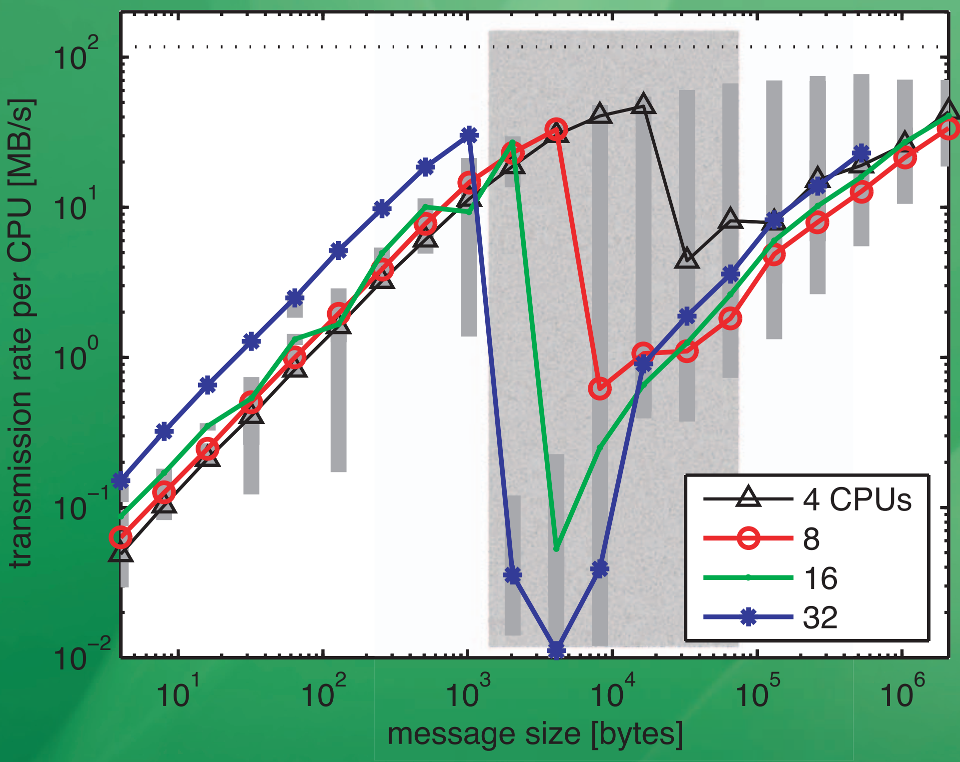
CPUs	byte to each CPU
2	649440
4	173184
6	73800
8	43296
16	11808
32	2952

Guanylin, PME grid
30x30x21=18900 floats

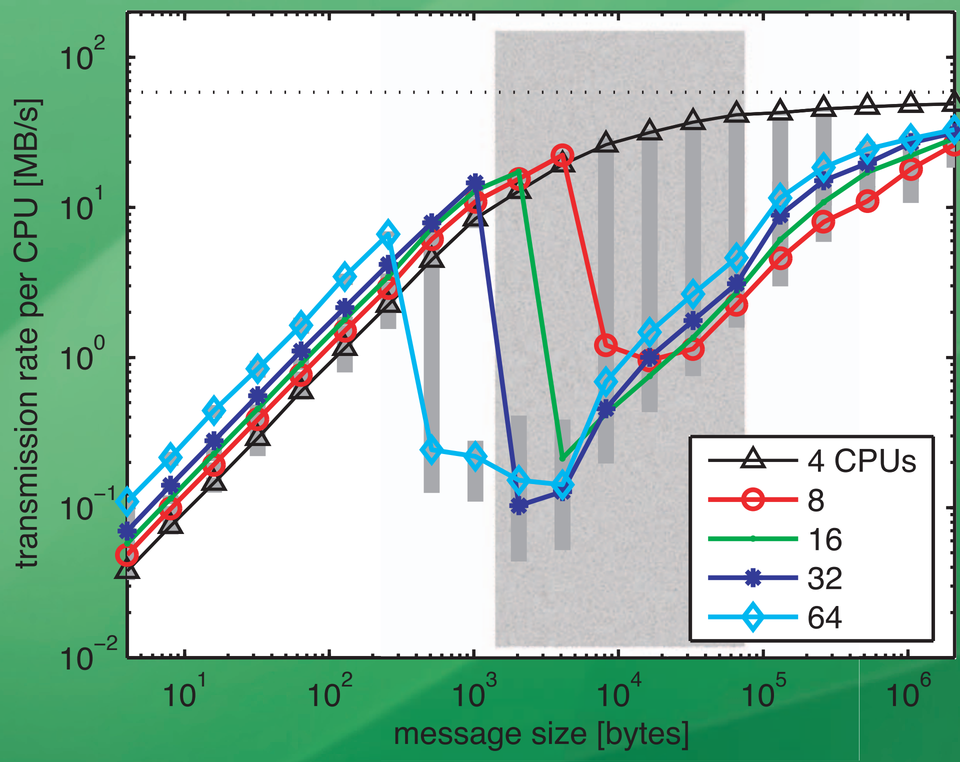
CPUs	byte to each CPU
2	19800
4	5632
6	2200
8	1408

LAM MPI_Alltoall

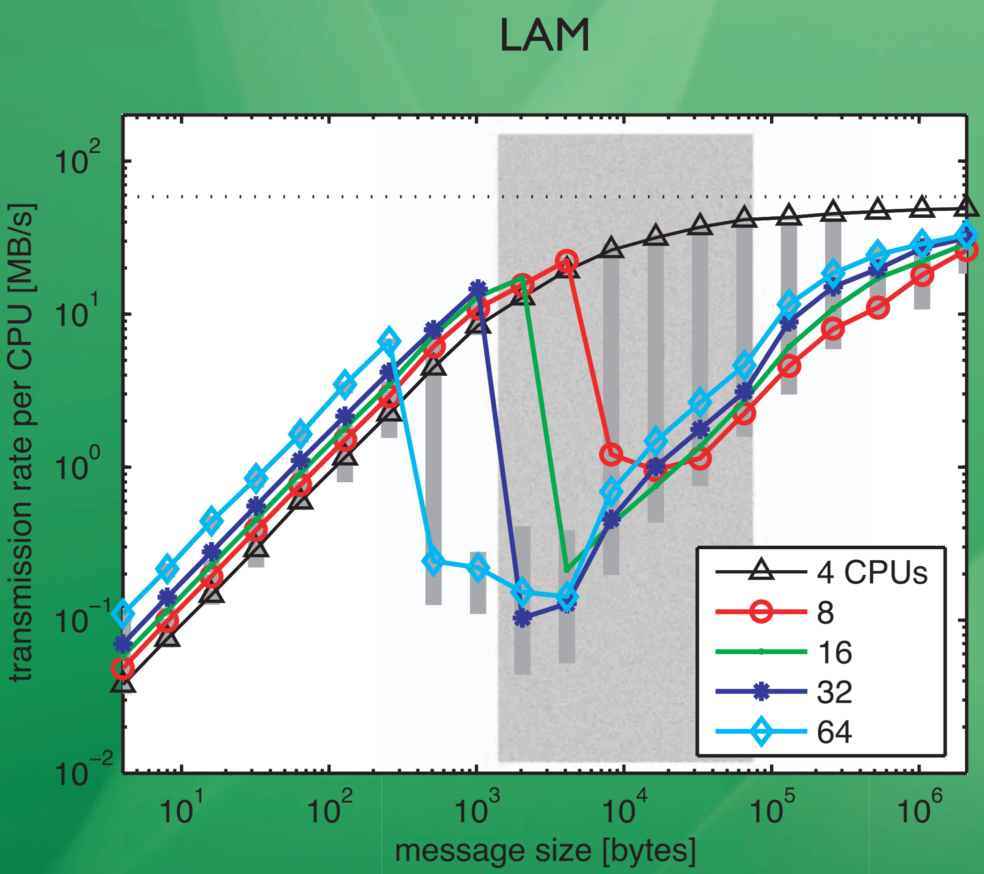
1 CPU/node

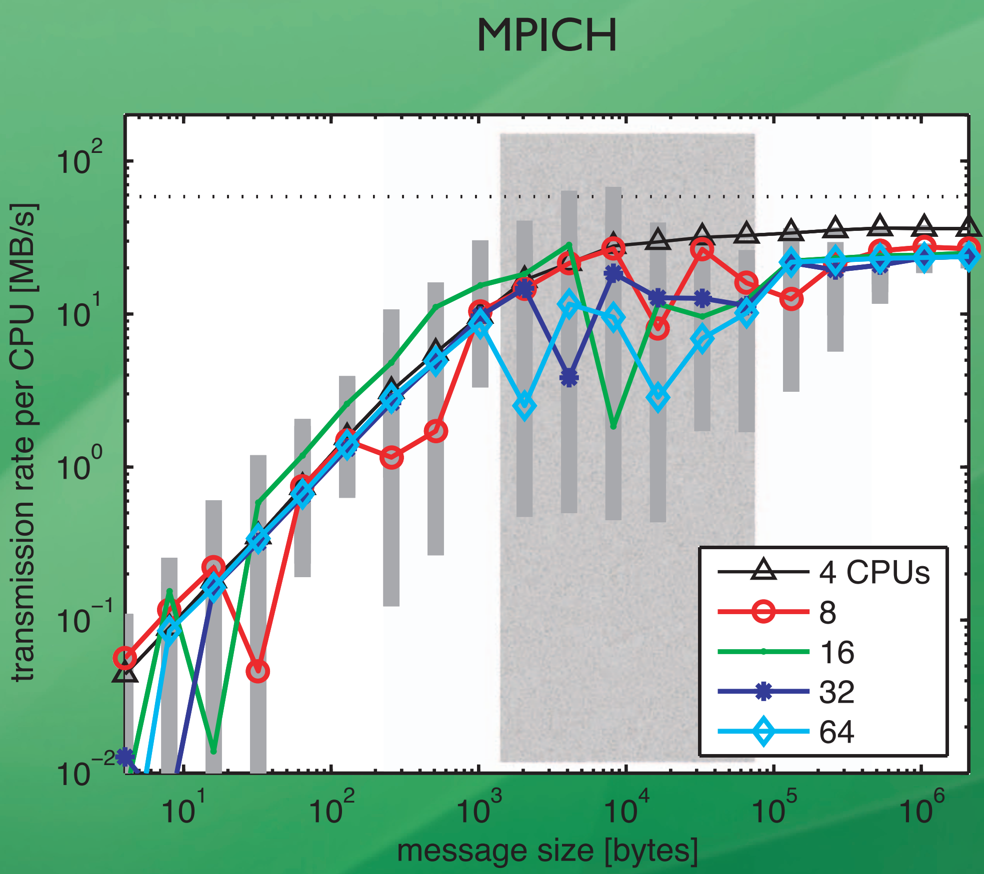


2 CPUs/node

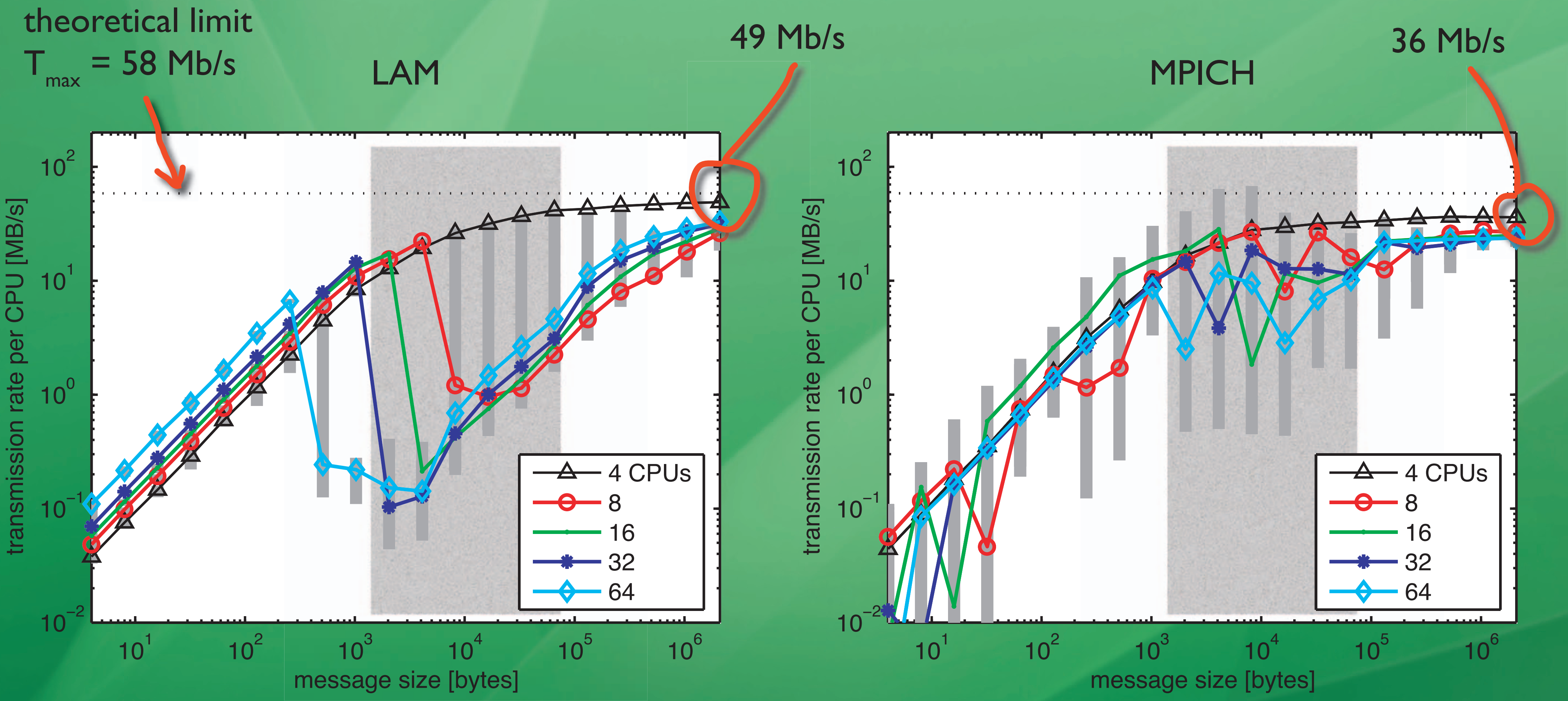


LAM vs. MPICH (2 CPUs/node)



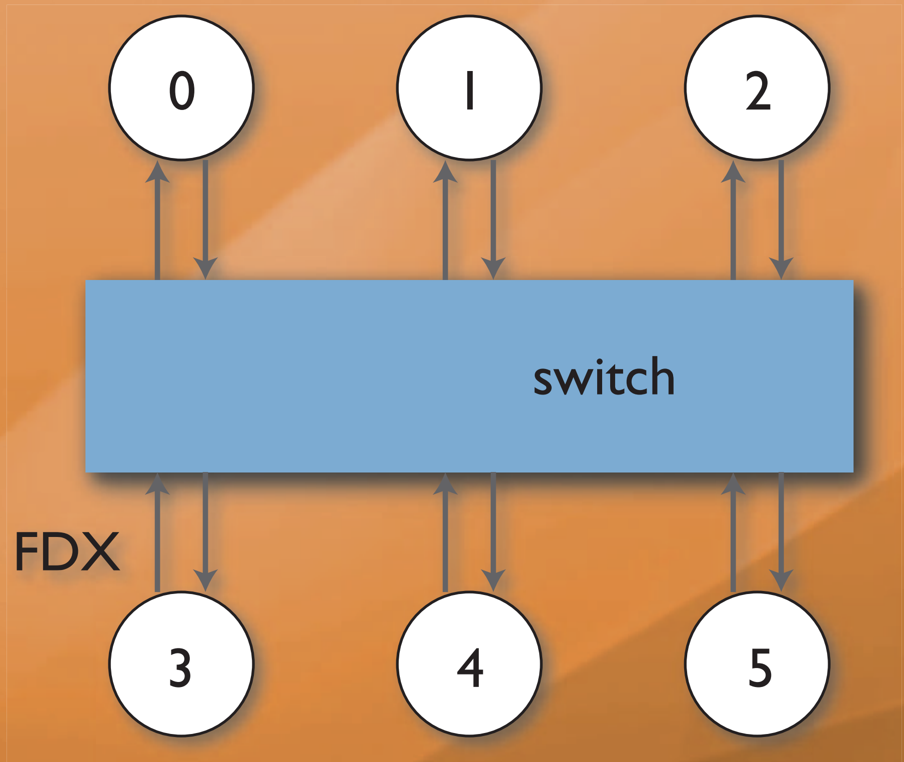


LAM vs. MPICH (2 CPUs/node)



IEEE 802.3x flow control

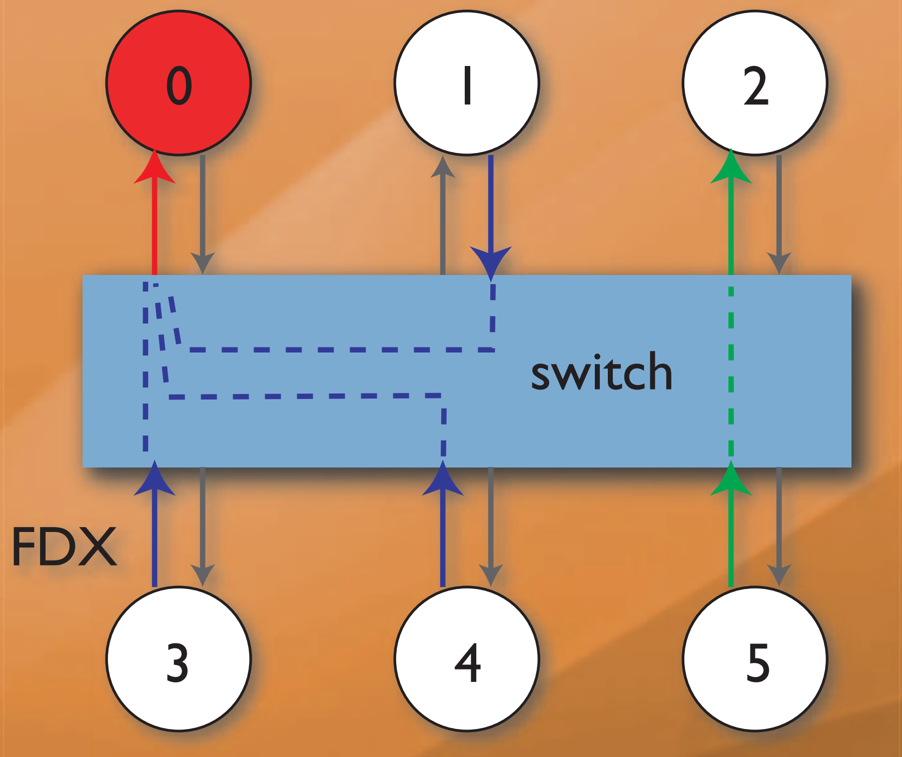
- IEEE 802.3x[1] defines low level flow control mechanism to prevent network congestion
- flow of ... messages/data/TCP packets
- receivers may send PAUSE frames to tell source to stop sending
- reduce packet loss!



1 IEEE, 2000. Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements–Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. IEEE 802.3, New York, Annex 31B: MAC control PAUSE operation.

IEEE 802.3x flow control

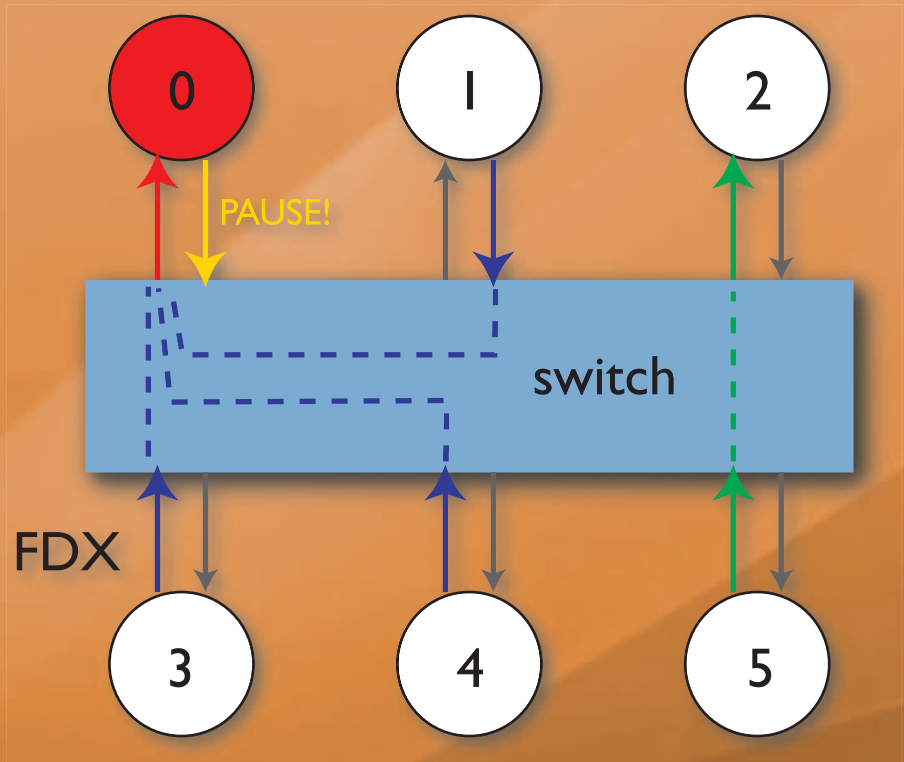
- IEEE 802.3x[1] defines low level flow control mechanism to prevent network congestion
- flow of ... messages/data/TCP packets
- receivers may send PAUSE frames to tell source to stop sending
- reduce packet loss



1 IEEE, 2000. Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements–Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. IEEE 802.3, New York, Annex 31B: MAC control PAUSE operation.

IEEE 802.3x flow control

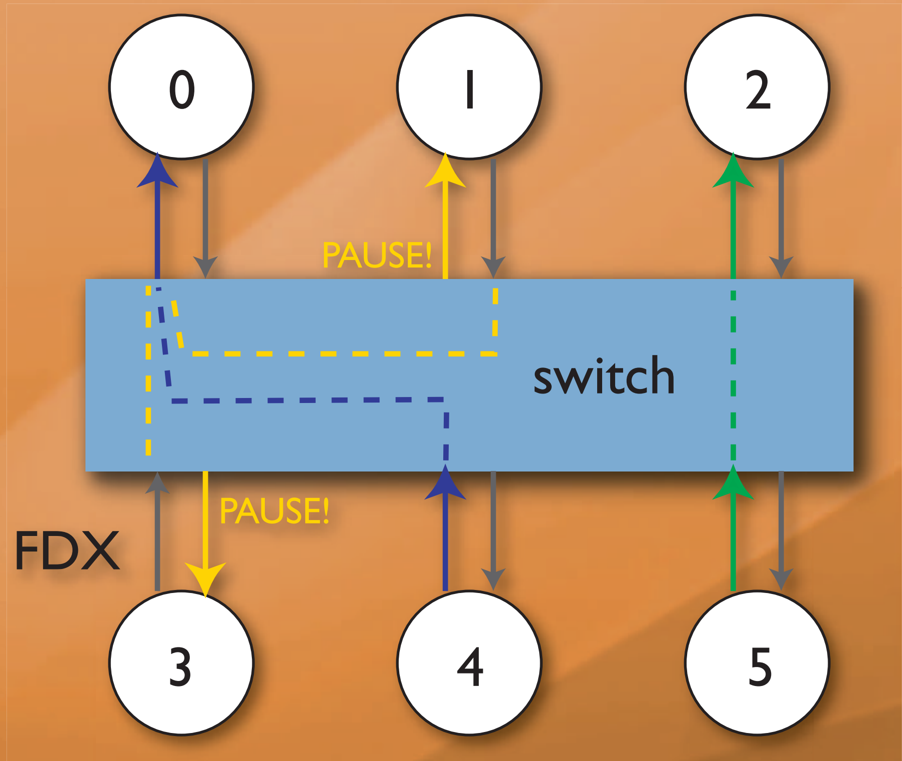
- IEEE 802.3x[1] defines low level flow control mechanism to prevent network congestion
- flow of ... messages/data/TCP packets
- receivers may send PAUSE frames to tell source to stop sending
- reduce packet loss



1 IEEE, 2000. Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements–Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. IEEE 802.3, New York, Annex 31B: MAC control PAUSE operation.

IEEE 802.3x flow control

- IEEE 802.3x[1] defines low level flow control mechanism to prevent network congestion
- flow of ... messages/data/TCP packets
- receivers may send PAUSE frames to tell source to stop sending
- reduce packet loss



1 IEEE, 2000. Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements–Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. IEEE 802.3, New York, Annex 31B: MAC control PAUSE operation.

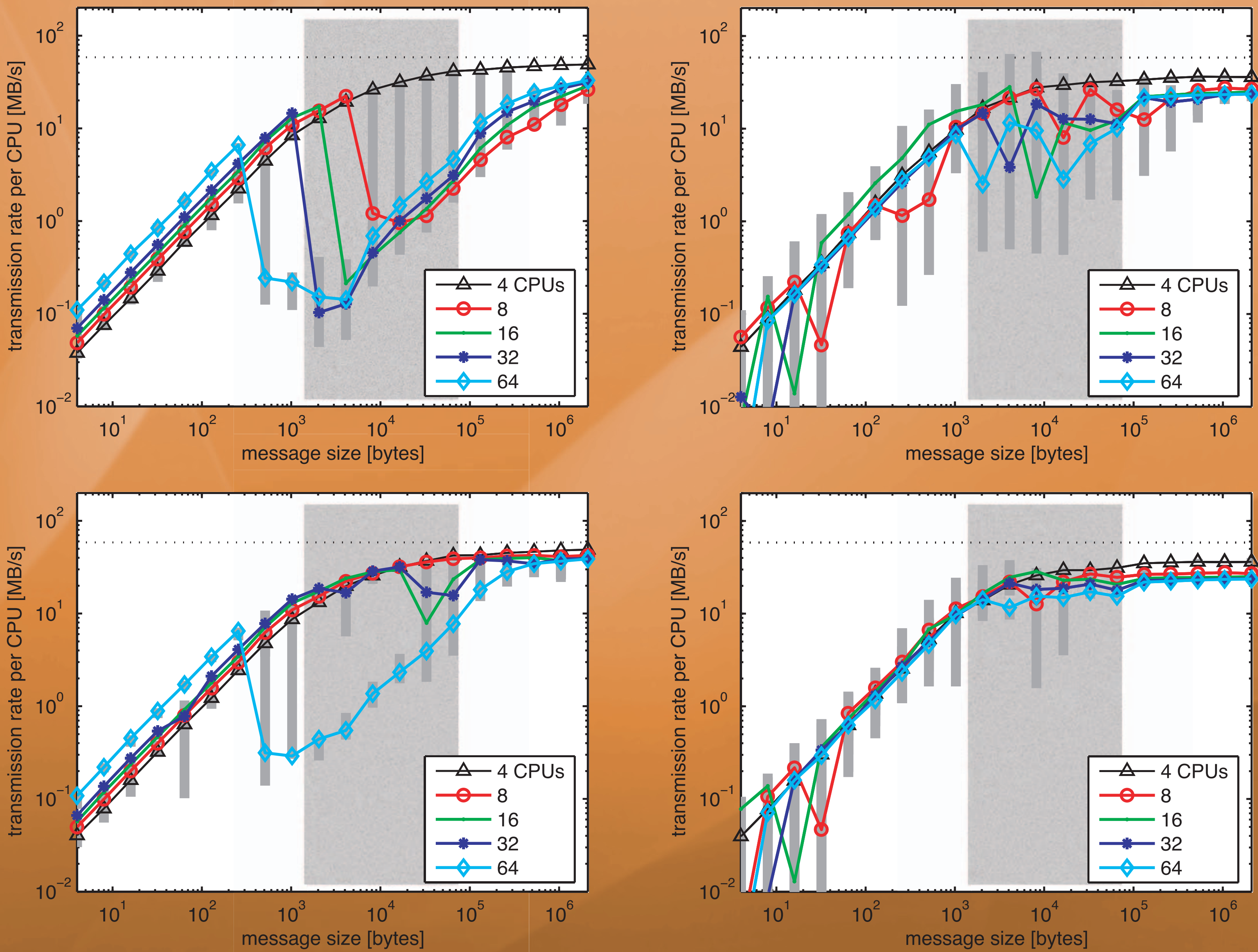
MPI_Alltoall performance (2 CPUs/node)

LAM

MPICH

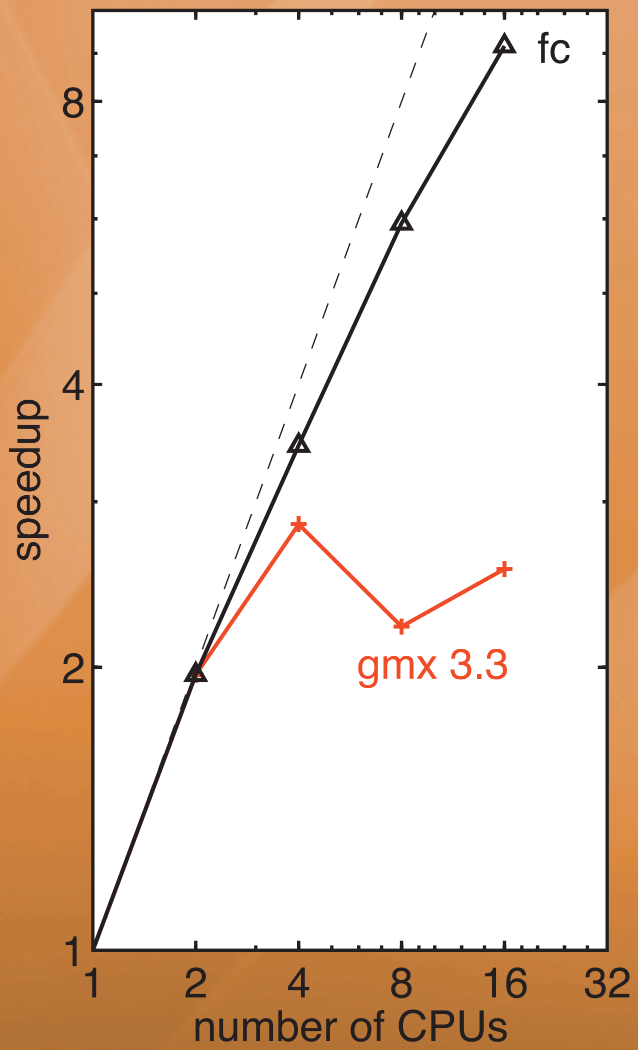
no flow control

with IEEE 802.3x flow control

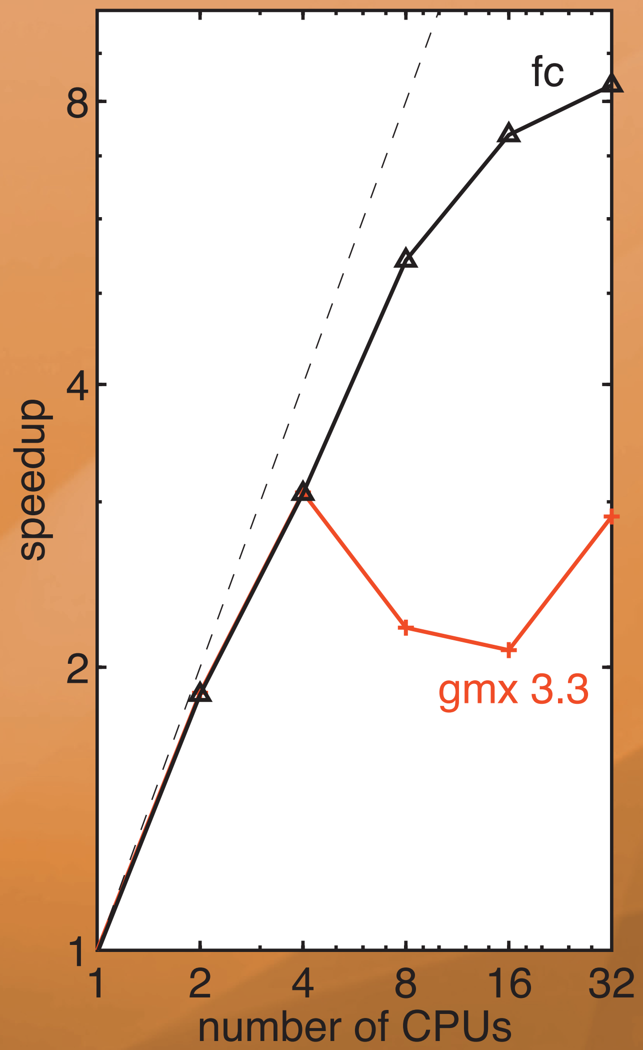


GROMACS 3.3 GigE speedup

1 CPU/node



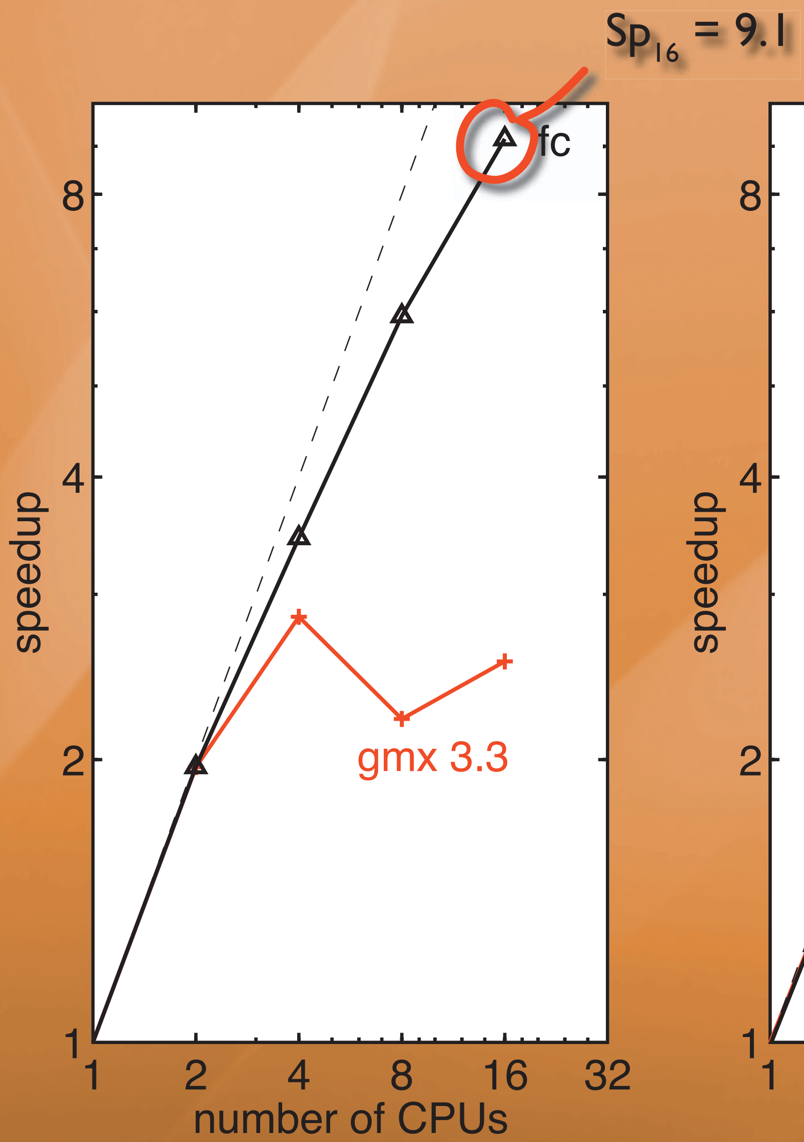
2 CPUs/node

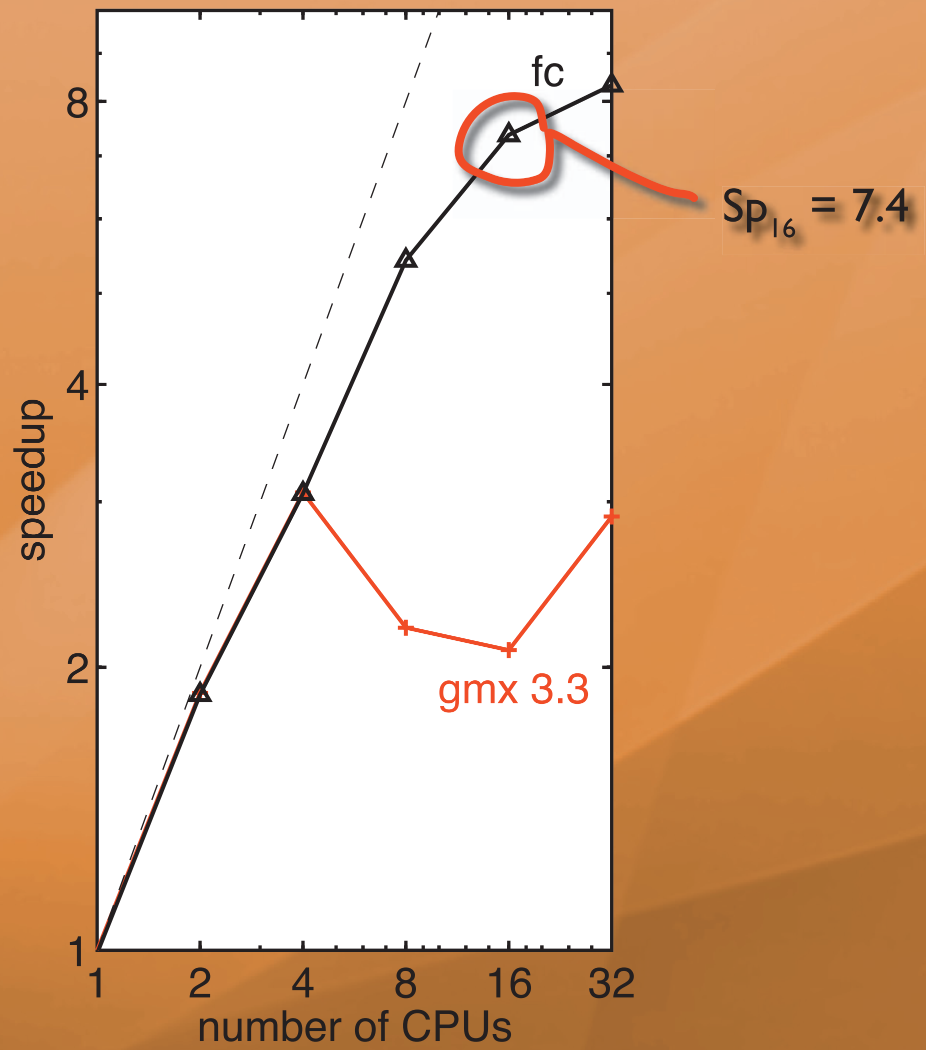


GROMACS 3.3 GigE speedup

1 CPU/node

2 CPUs/node

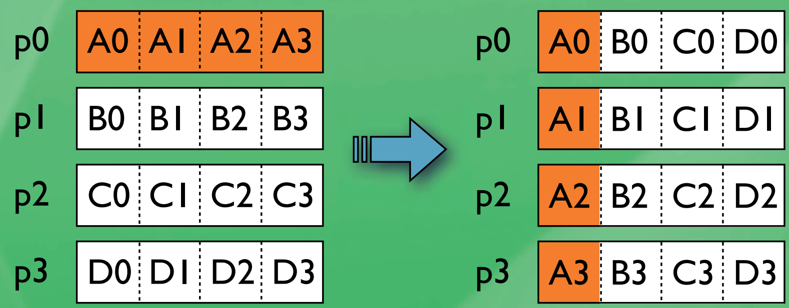




High-level flow control

- low-level flow control does not guarantee good performance for 16+ CPUs

- implement own all-to-all



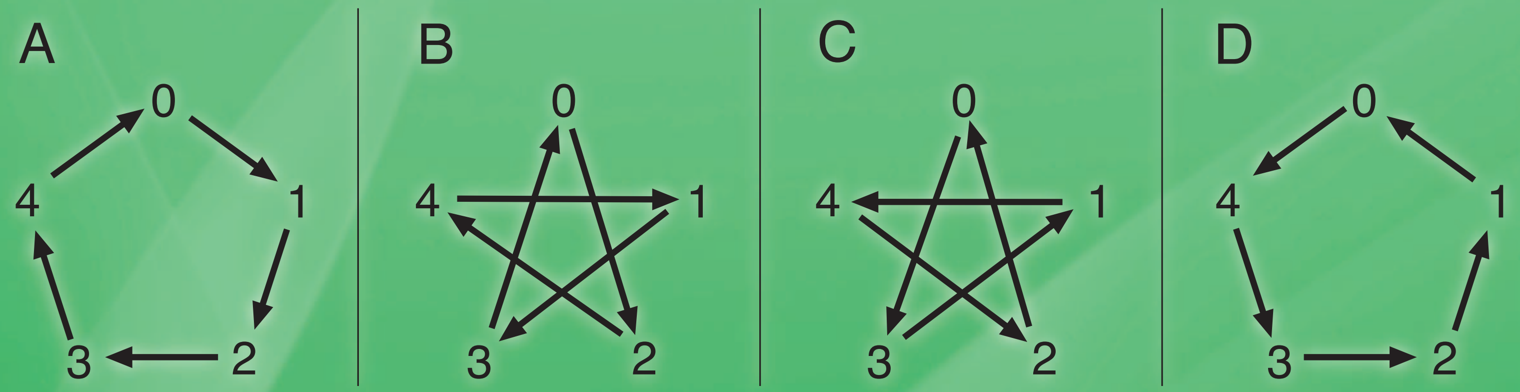
- idea:
  arrange communication in separated phases, each free of congestion

- experimental MPI prototype: CC-MPI[2]

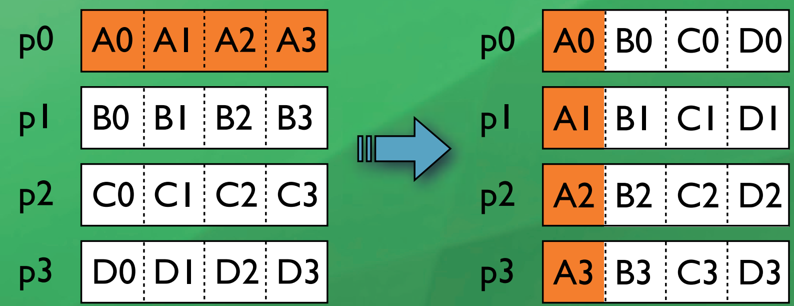
2 Karwande et al., 2005, An MPI prototype for compiled communication on Ethernet switched clusters, J Parallel Distrib Comput 65

High-level flow control

ordered all-to-all for single-CPU nodes



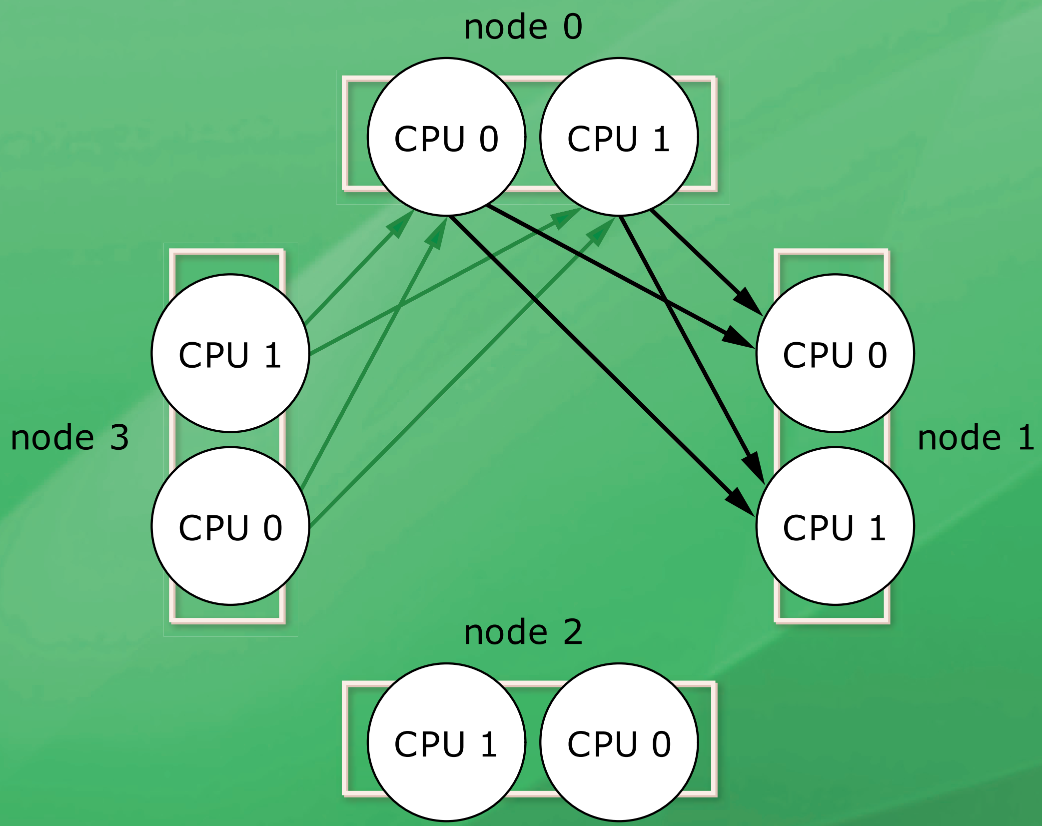

```
for (i=0; i<ncpu; i++)   /* loop over all CPUs */
{
   /* send to destination CPU
      while receiving from source CPU: */
   dest = (cpuid+i) % ncpu;
   source = (ncpu+cpuid-i) % ncpu;
   MPI_Sendrecv(send to dest, recv from source);
   /* separate the communication phases: */
   if (i<ncpu-1)
      MPI_Barrier(comm);
}
```



High-level flow control

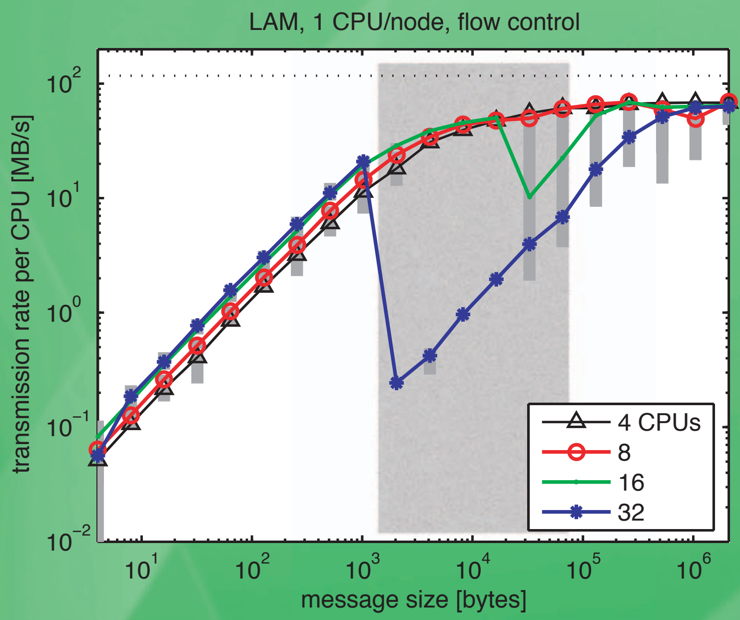
ordered all-to-all for multi-CPU nodes

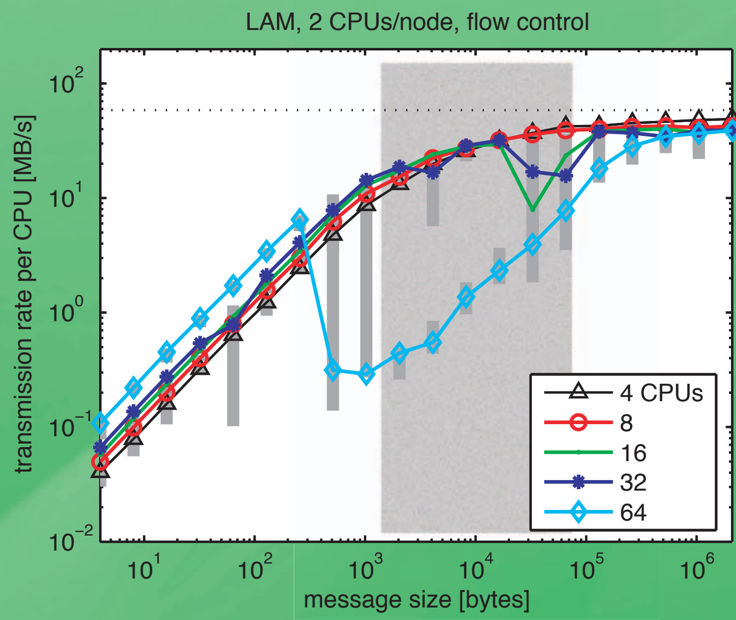
```
/* loop over all nodes */
for (i=0; i<nnodes; i++)
{
   /* send to destination node
      while receiving from source node: */
    destnode = (nodeid+i) % nnodes;
    sourcenode = (nnodes+nodeid-i) % nnodes;
    /* loop over CPUs on a node: */
    for (j=0; j < cpus_per_node; j++)
    {
       /* source and destination CPU: */
       destcpu  =destnode*cpus_per_node + j;
       sourcecpu=sourcenode*cpus_per_node+j;
       MPI_Irecv(from sourcecpu ...);
       MPI_Isend(to destcpu ...);
    }
    /* wait for communication to finish: */
    MPI_Waitall(...);
    MPI_Waitall(...);
    /* separate the communication phases: */
    if (i<nnodes-1)
       MPI_Barrier(...);
}
```



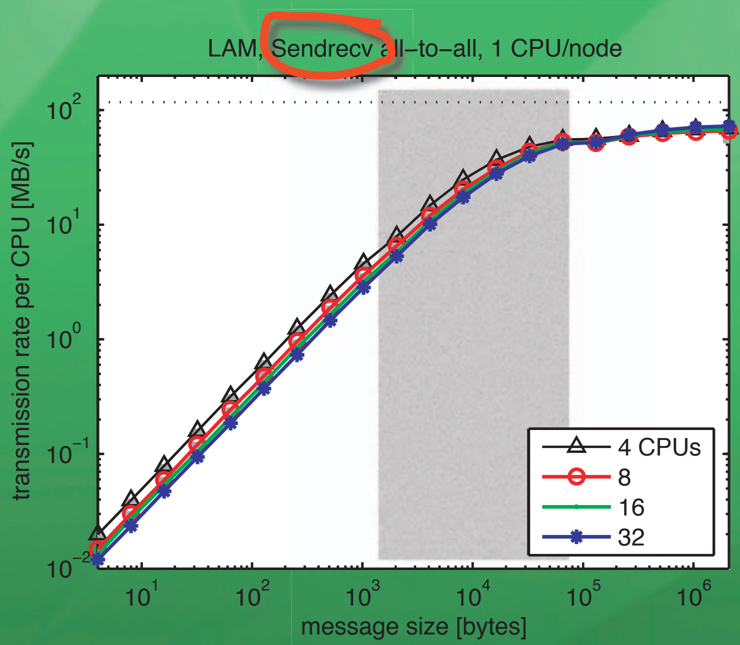
High- vs. low level flow control – LAM

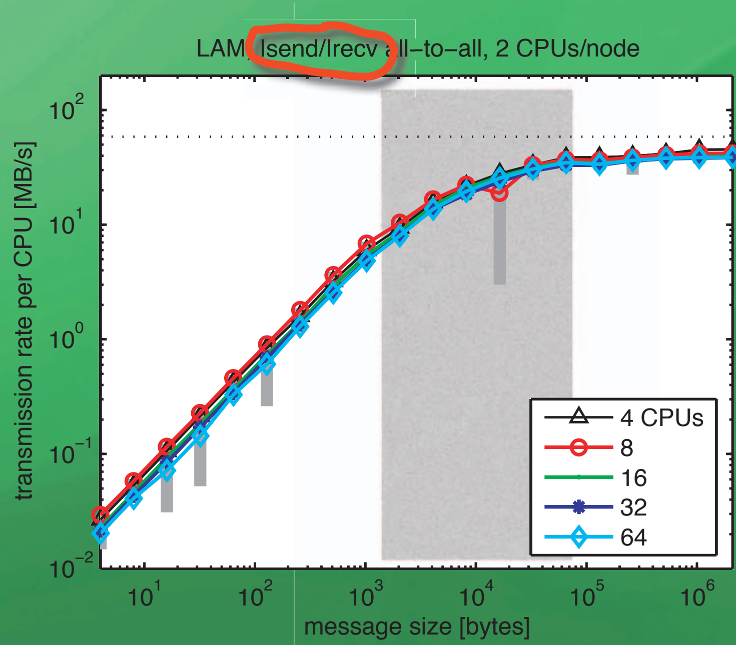
with
IEEE 802.3x
flow control





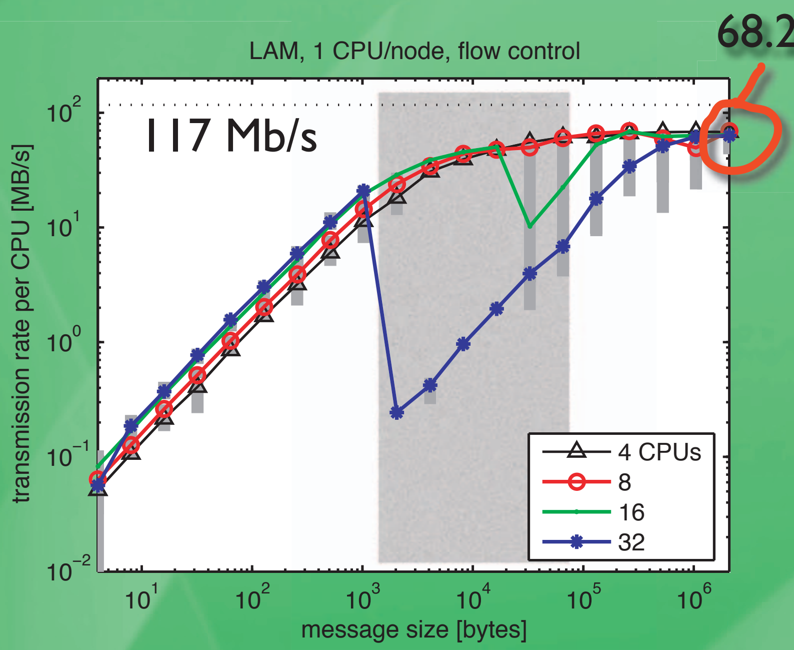
with
high-level
flow control

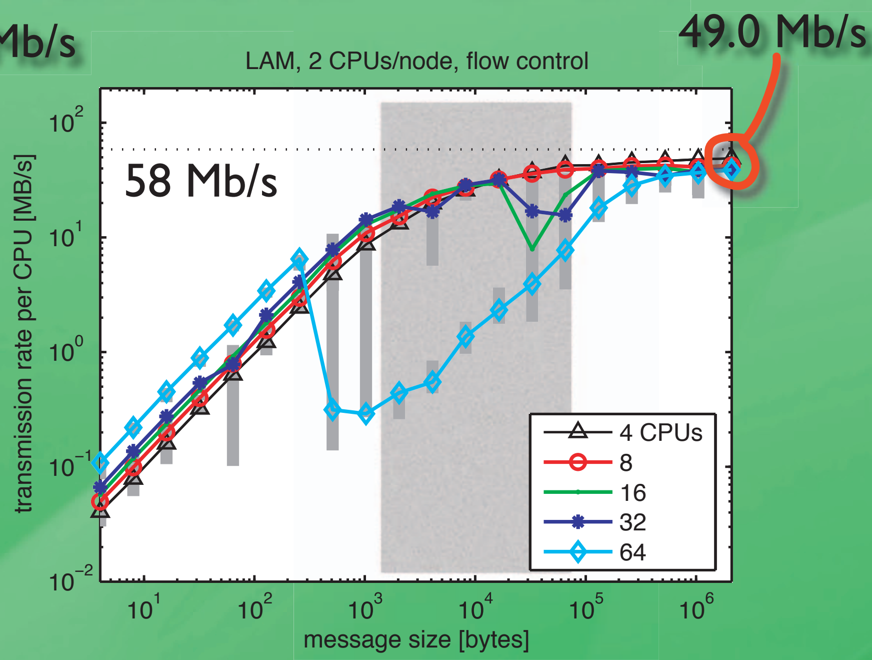




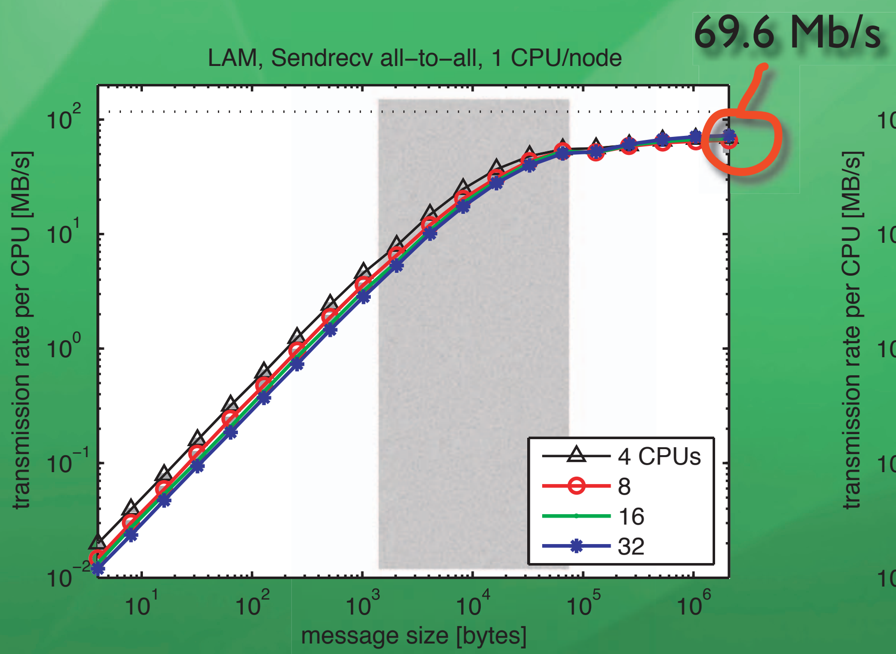
High- vs. low level flow control – LAM

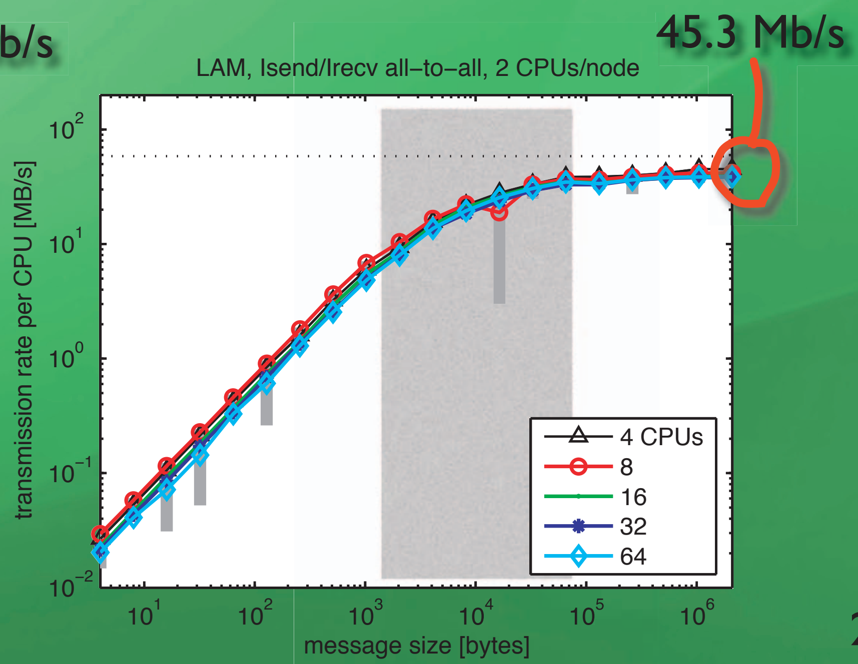
with
IEEE 802.3x
flow control





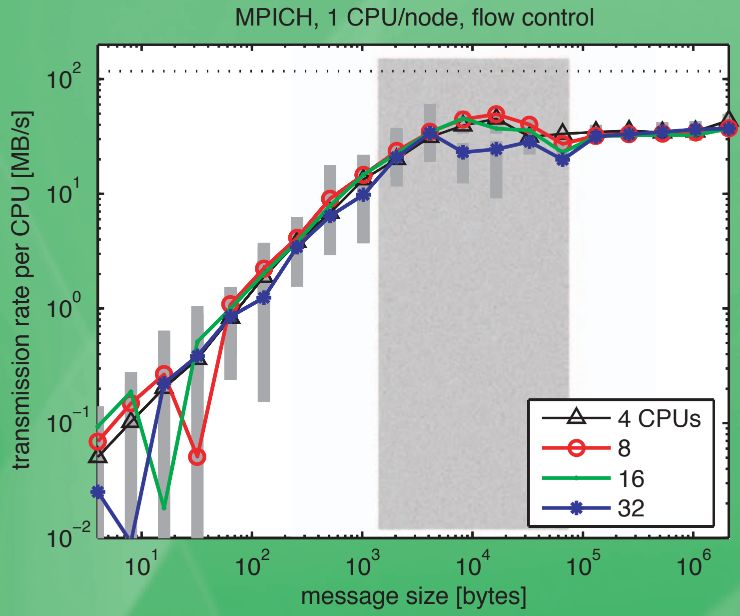
with
high-level
flow control

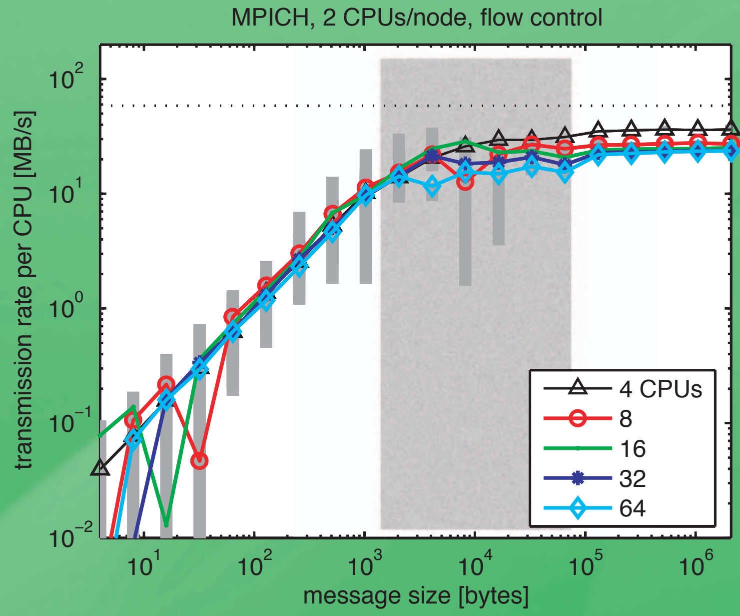




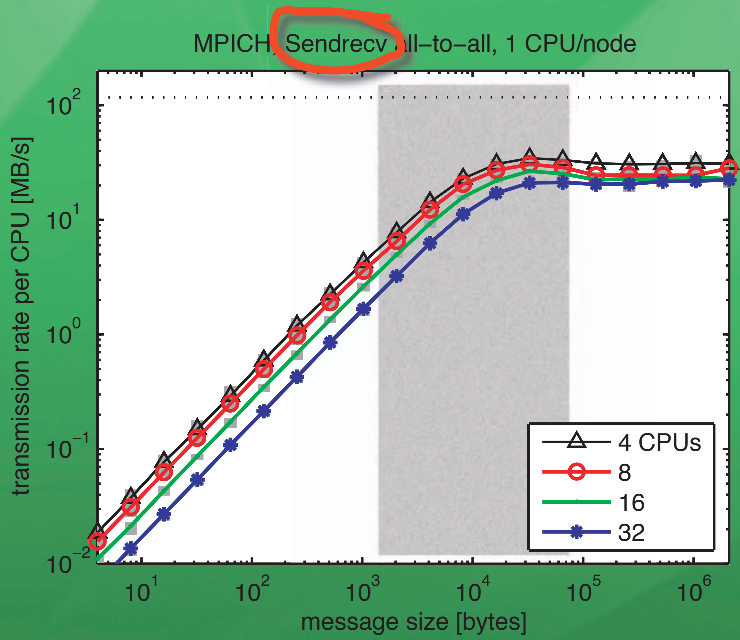
High- vs. low level flow control – MPICH

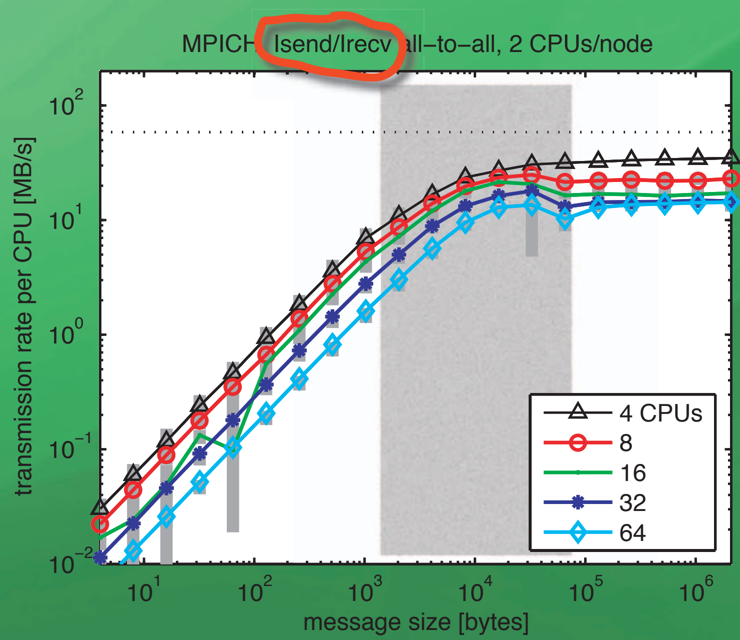
with
IEEE 802.3x
flow control





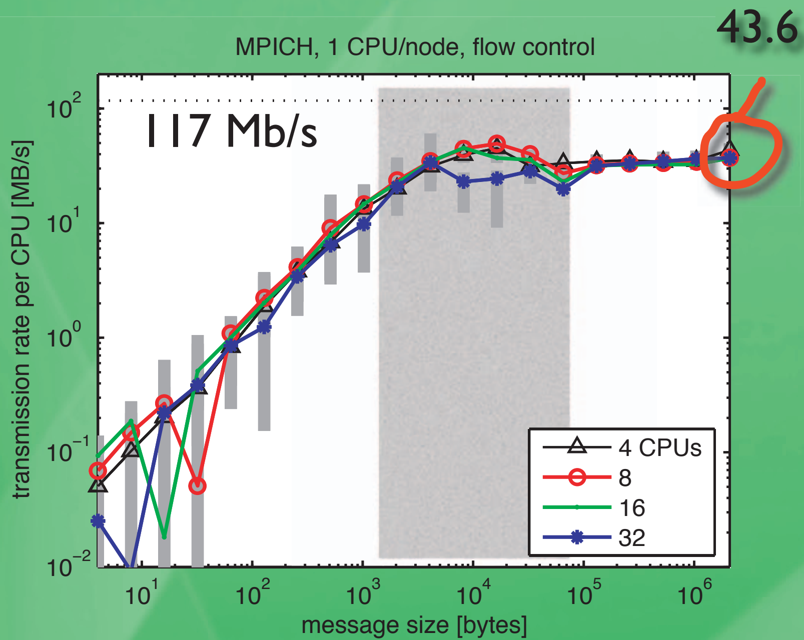
with
high-level
flow control

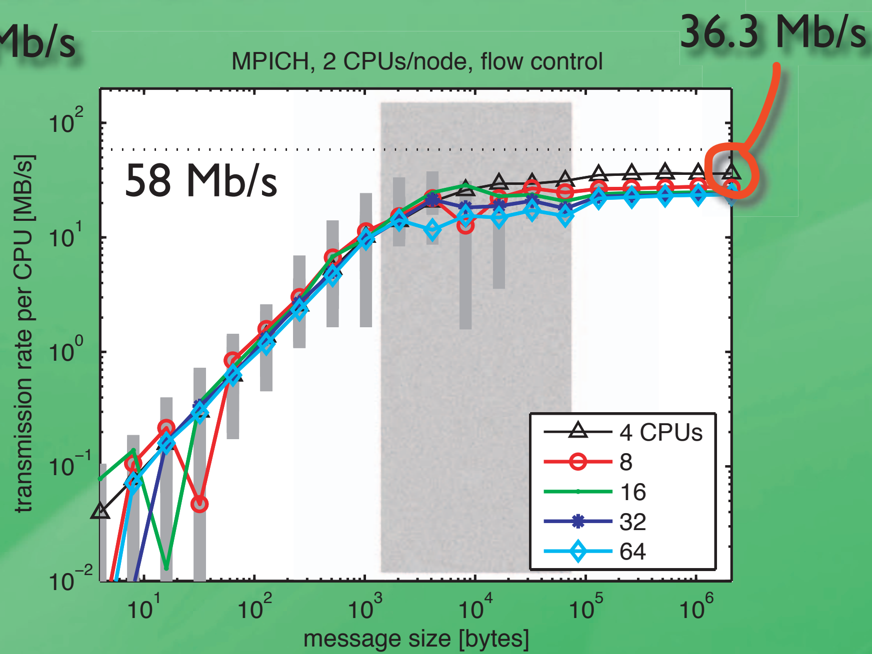




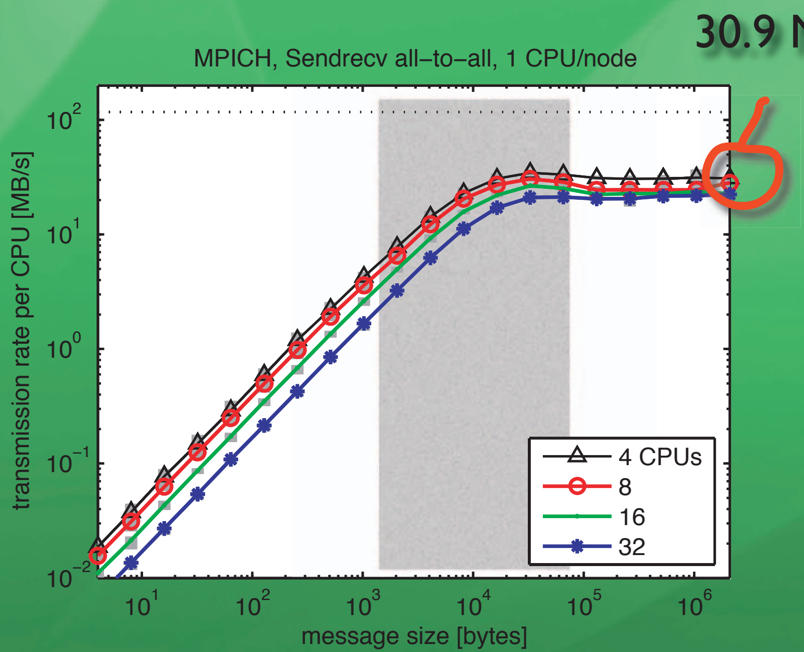
High- vs. low level flow control – MPICH

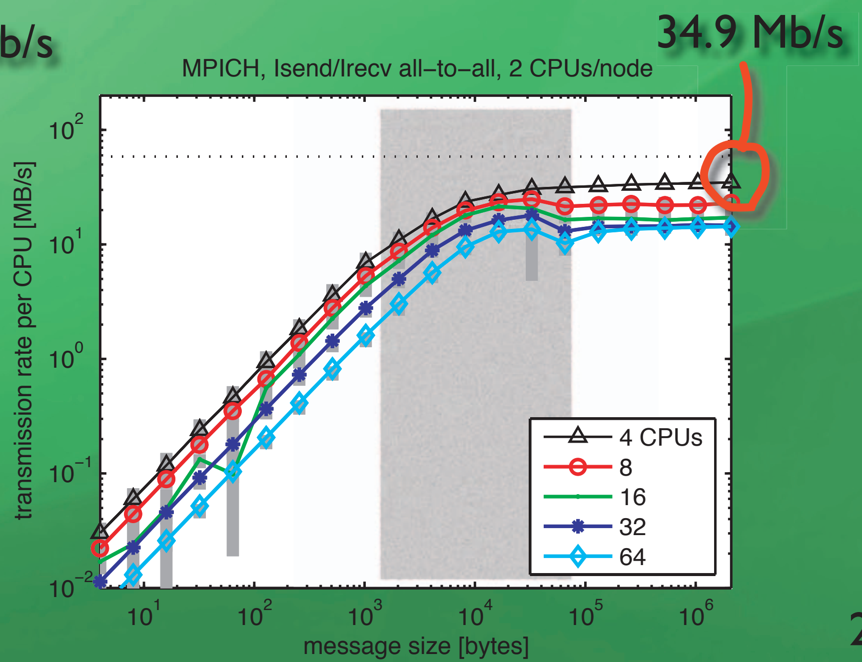
with
IEEE 802.3x
flow control





with
high-level
flow control





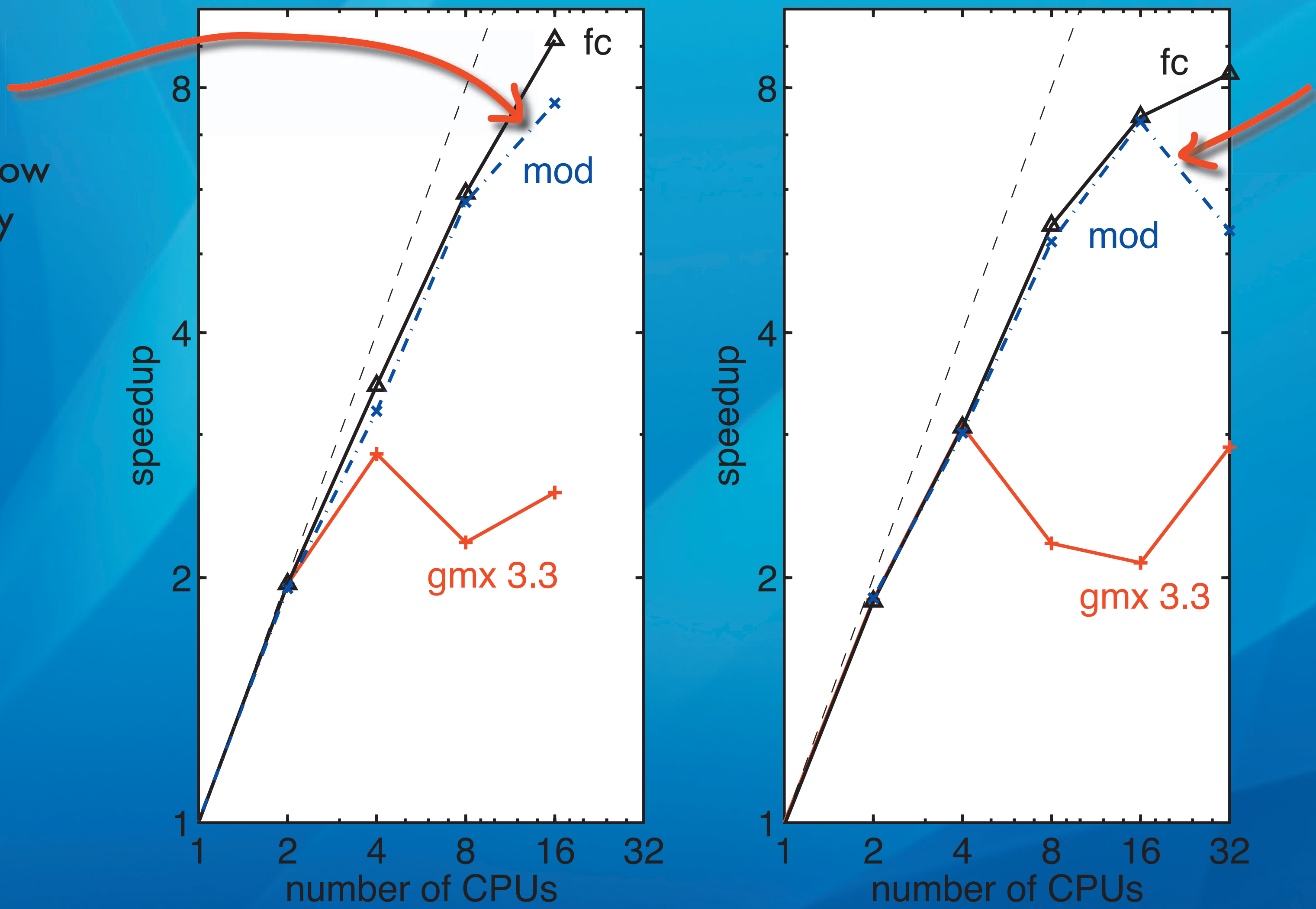
GROMACS 3.3 GigE speedup

MPI_Alltoall vs. ordered all-to-alls

1 CPU/node

2 CPUs/node

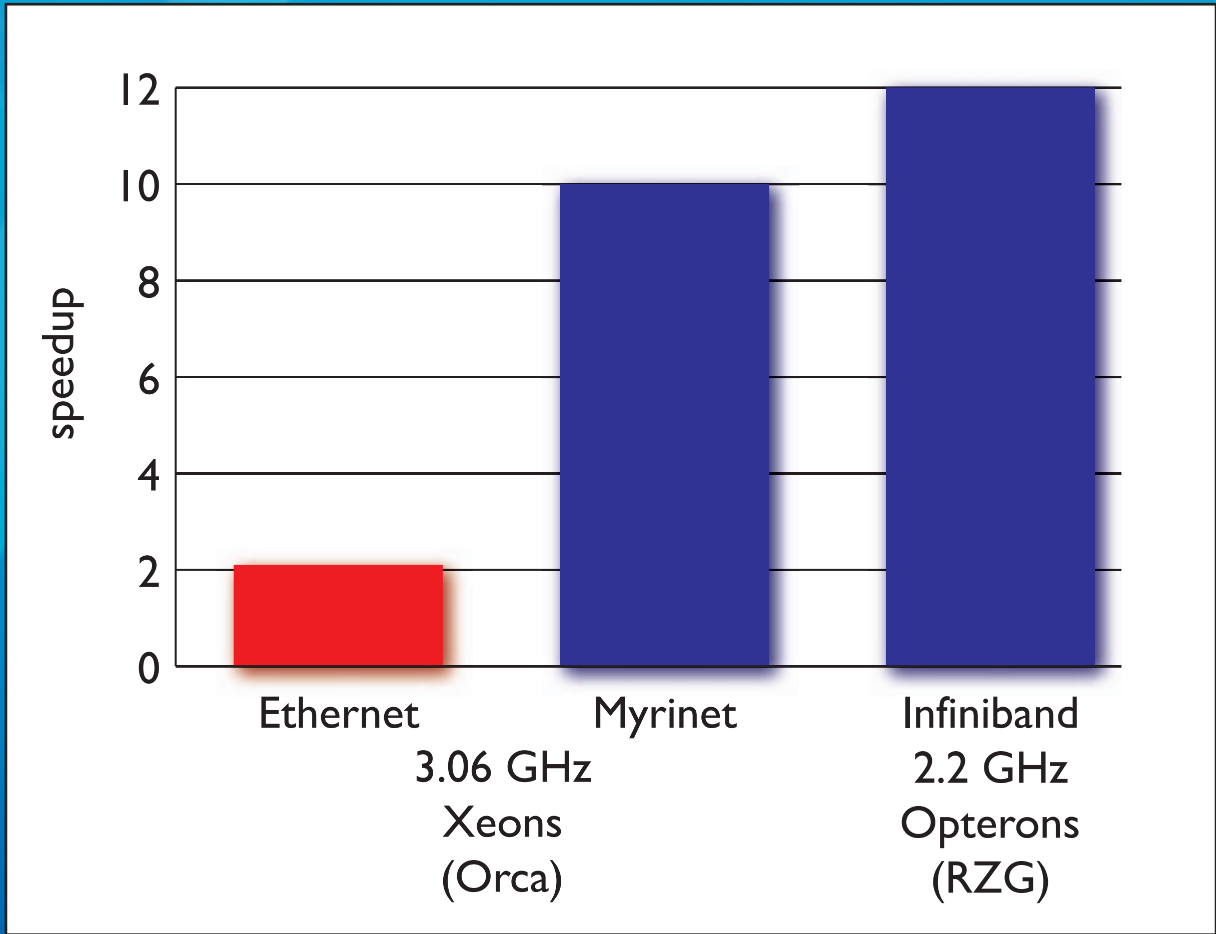
Sendrecv-scheme, high-level flow control only



Isend/Irecv scheme, high-level flow control only

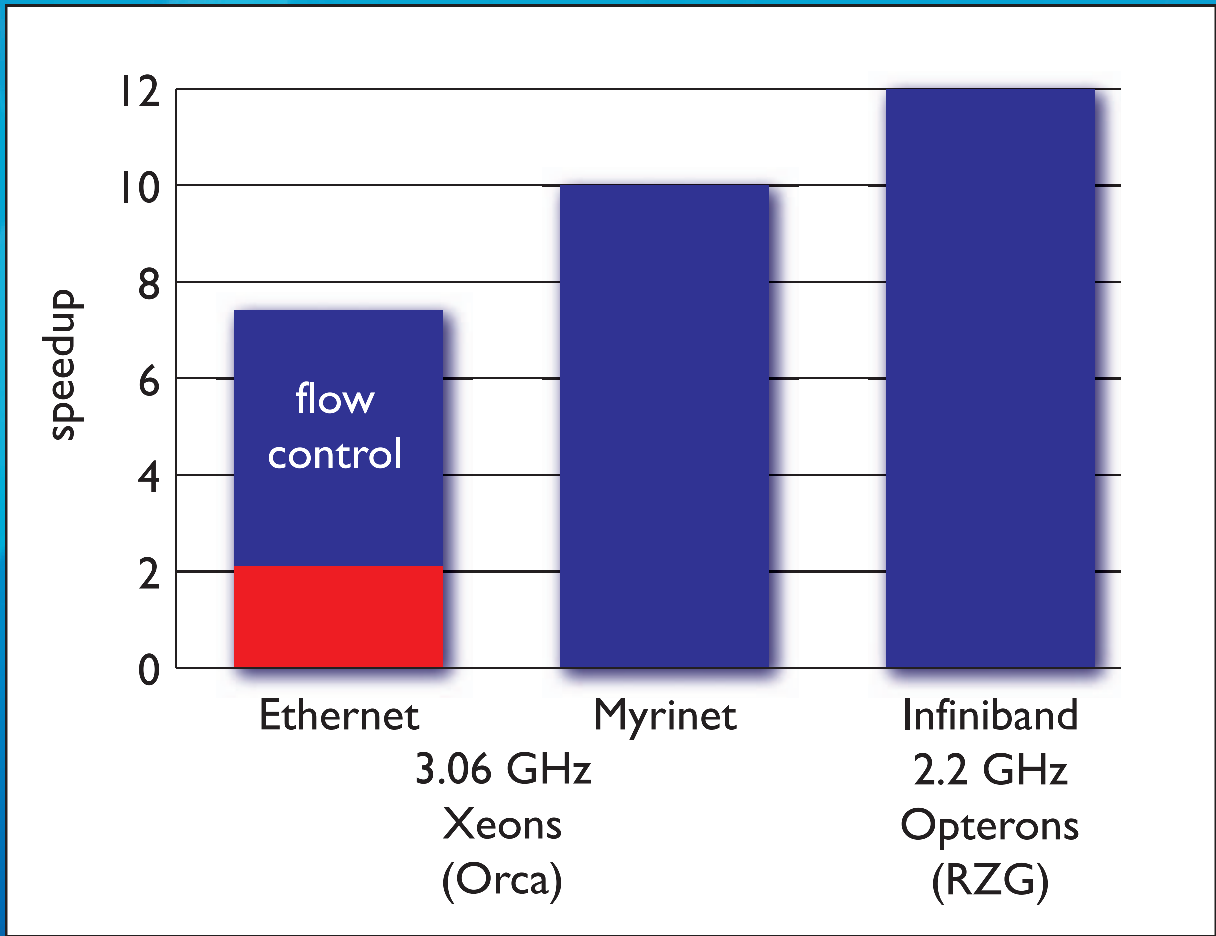
GROMACS 3.3 GigE speedup

compared to other interconnects (8x2 CPUs)



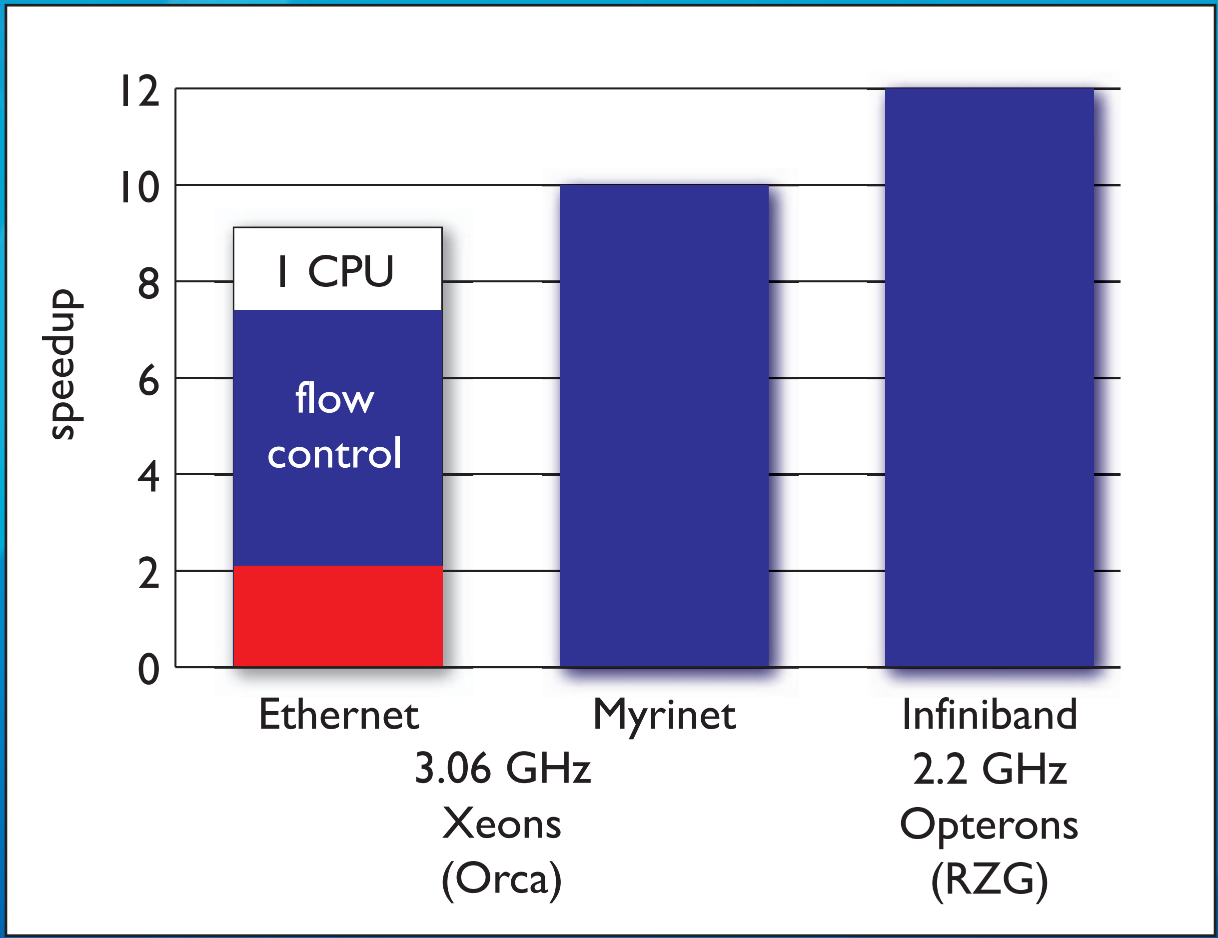
GROMACS 3.3 GigE speedup

compared to other interconnects (8x2 CPUs)



GROMACS 3.3 GigE speedup

compared to other interconnects (8x2 CPUs)



Conclusions

- when running GROMACS on >2 Ethernet-connected nodes, flow control is needed to avoid network congestion
- implemented a congestion-free all-to-all for multi-CPU nodes
- highest network throughput probably for combination of high+low level flow control
- findings can be applied to other parallel applications that need all-to-all communication

Outlook

- MPICH1-1.2.6 → MPICH2-1.0.2
- LAM-7.1.1 → OpenMPI-1.0rc1

- allocation technique for PME/PP splitting on multi-CPU nodes