# Switch “Jitter”

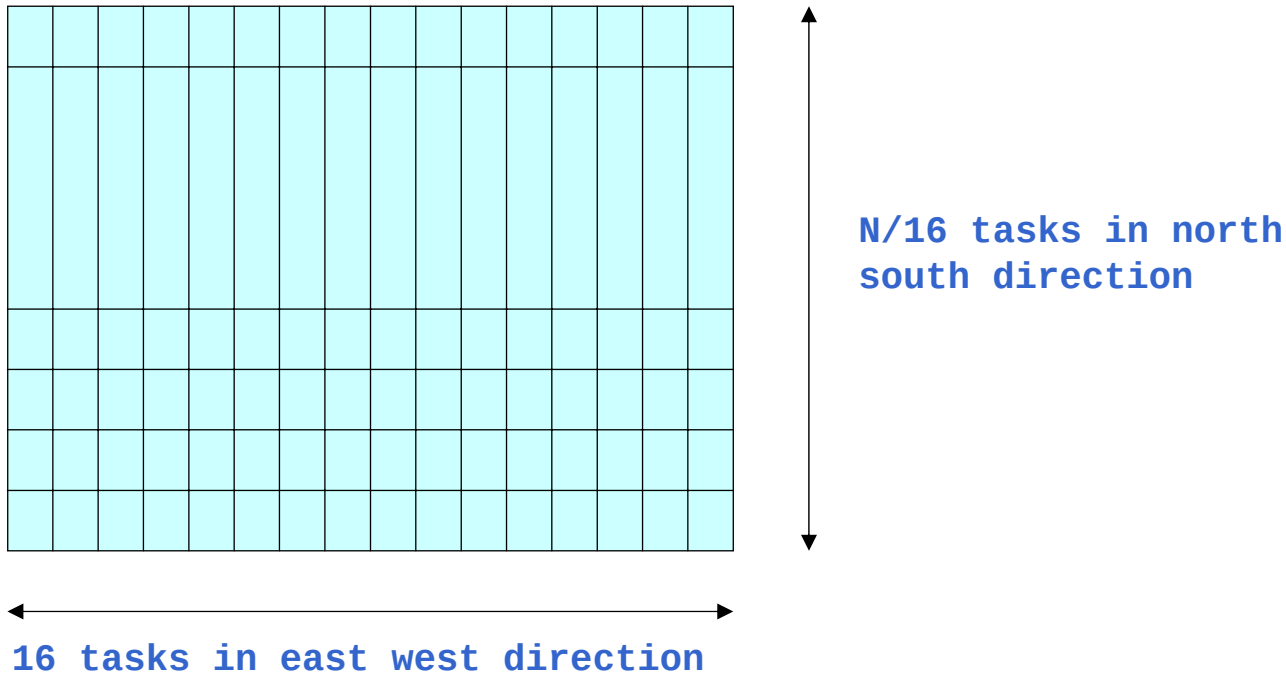
John Hague
IBM consultant
Nov/08

# Introduction

- Investigate Halo exchange time
  - One of simplest communication patterns
  - Expect increase with number of MPI tasks
  - Will not identity cause of “jitter”
- Run on P5+ system (hpce)
  - 16 cores per node
  - 2 “Federation” switch links per node
- Run on P6 system (c1a)
  - 32 cores per node
  - 8 InfiniBand switch links per node

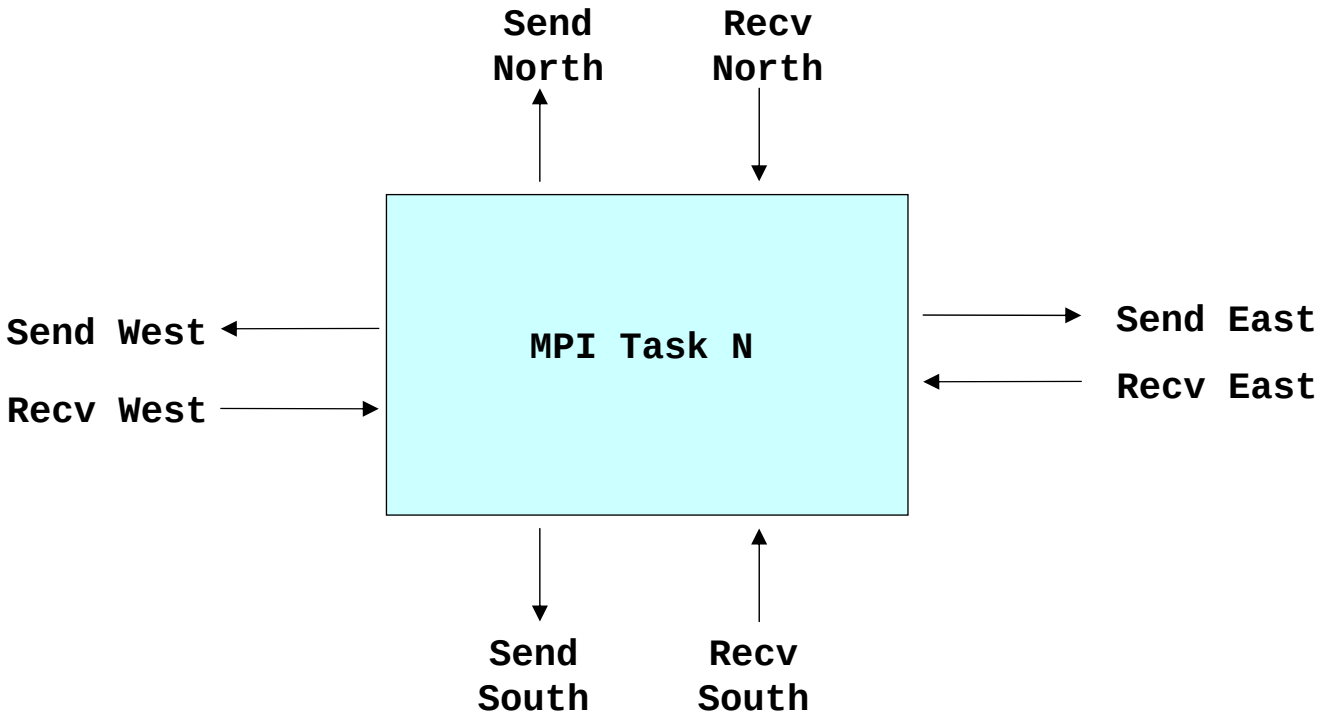
# Decomposition

- Global grid points split into 2D north/south east/west configuration
- For N MPI tasks
  - Each task allocated set of east/west, north/south grid points
  - Each task bound to core in node



# Halo exchange

- Halo exchange for each MPI task:



# Measurements

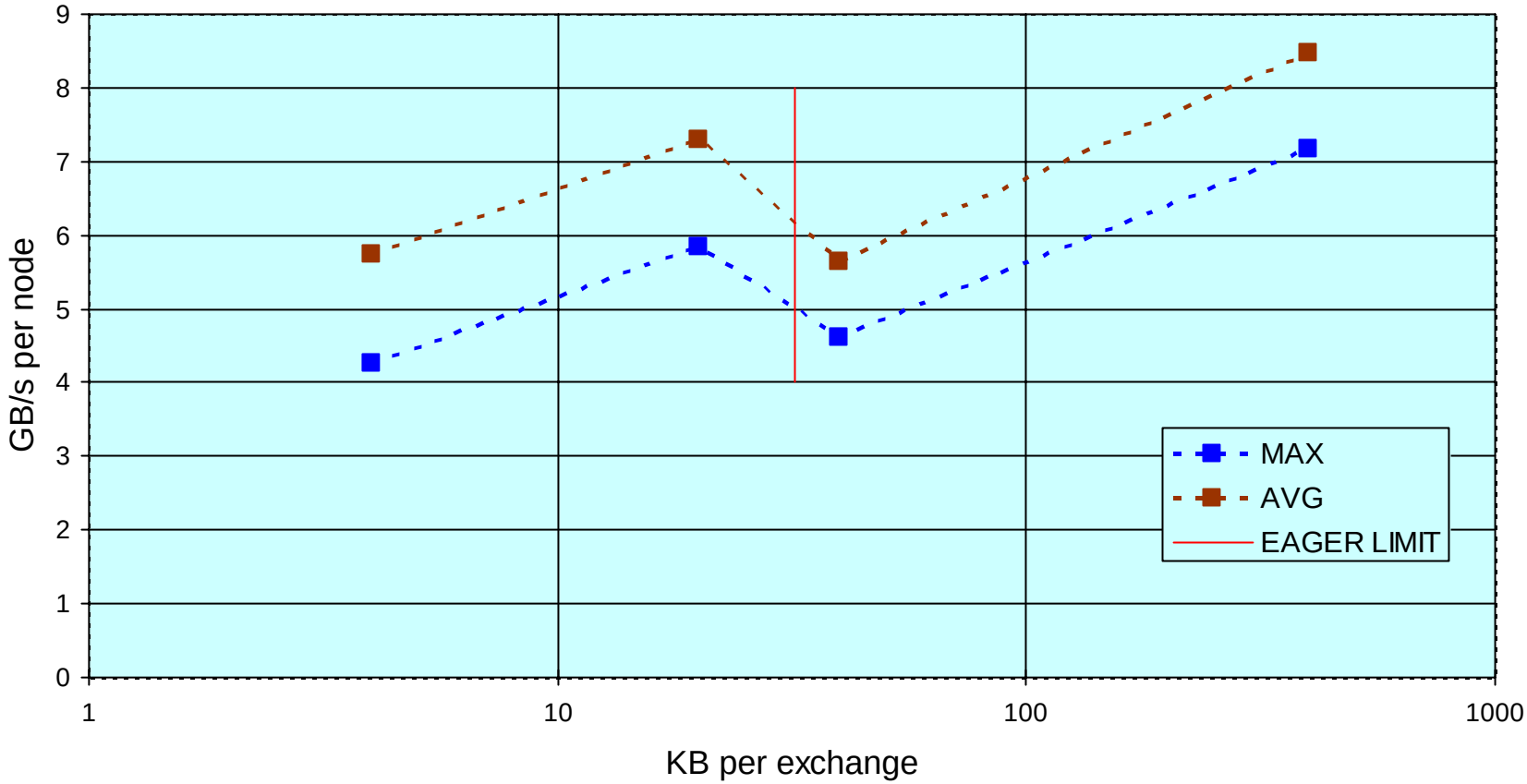
- For Various number of tasks
  - Issue 1000 halo calls in each task consisting of
    - Barrier
    - 2 north/south isends, 2 north/south irecvs, wait
    - 2 east/west isends, 2 east/west irecvs, wait
  - Measure each individual halo time on each task
  - Get average MAX time per halo call
    - Take maximum time over all tasks for each call to halo
    - Sum all maximums and take average
  - Get average AVG time per halo call
    - Take average time over all tasks for each call to halo
    - Sum all averages and take average

# Computation

Computation of MAX and AVG

| Repetition | 1 | 2 | 3 | 4 | | 1000 | Avg |
|---|---|---|---|---|---|---|---|
| | nnn | nnn | nnn | nnn | . . . | nnn | |
| | nnn | nnn | nnn | nnn | . . . | nnn | |
| | --- | --- | --- | --- | | --- | |
| Avg | aaa | aaa | aaa | aaa | . . . | aaa | -> AVG |
| Max | mmm | mmm | mmm | mmm | . . . | mmm | -> MAX |

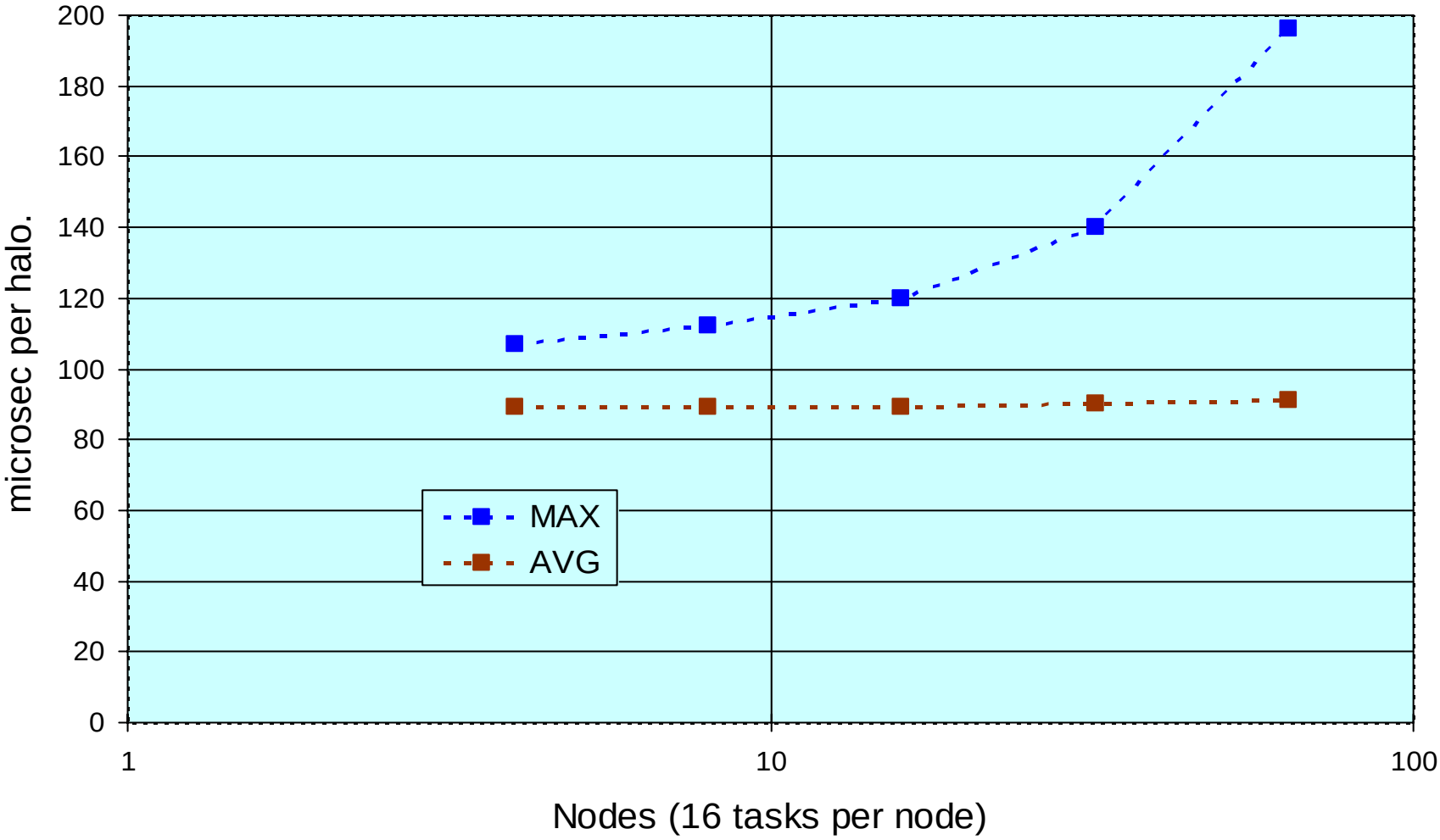
# GB/s per node on P5 (16 nodes, 256 tasks)




**2*(2*irecv+2*isend)**

**EAGER_LIMIT=32K**

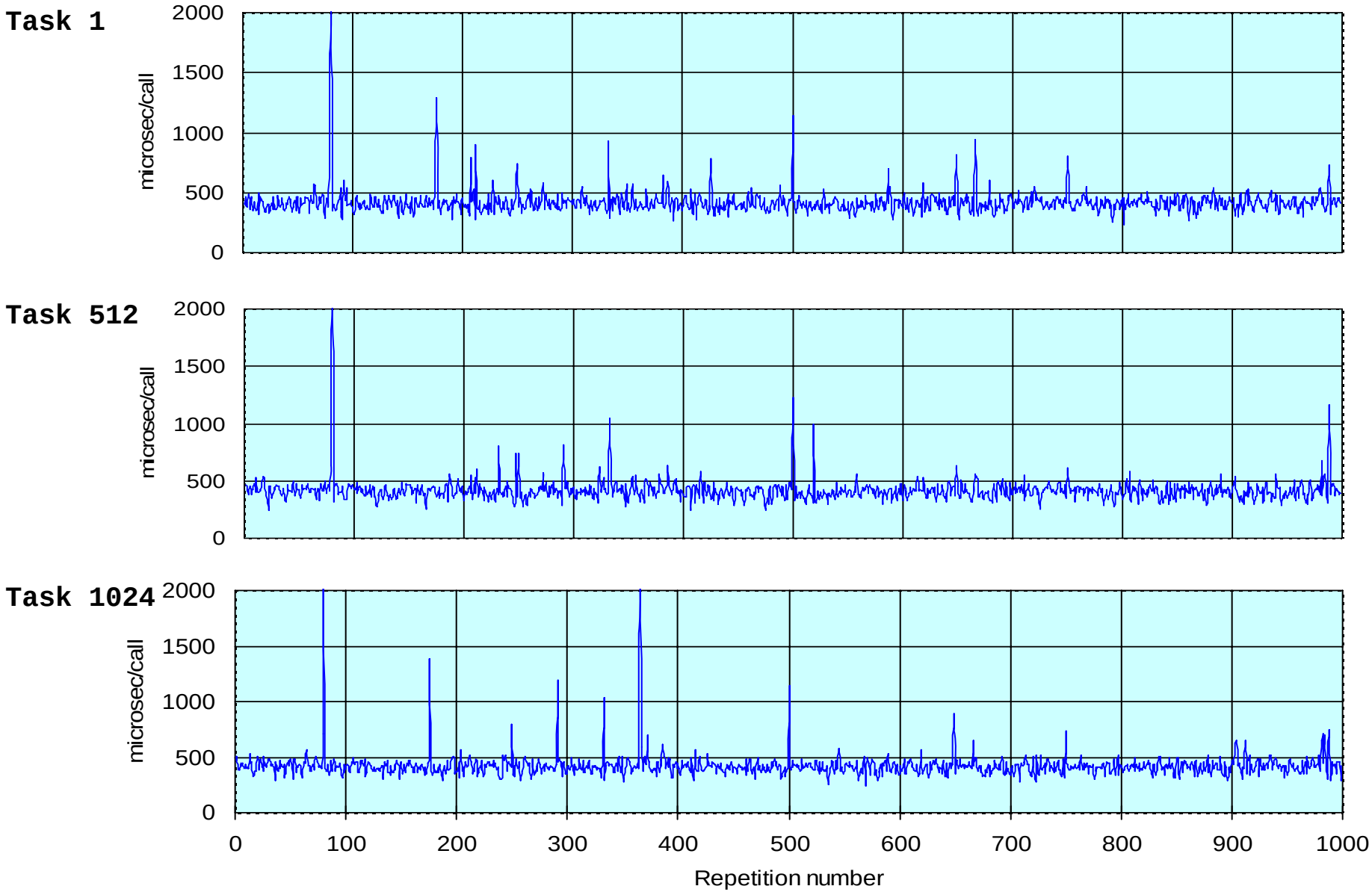
# Time for 4K exchanges on P5



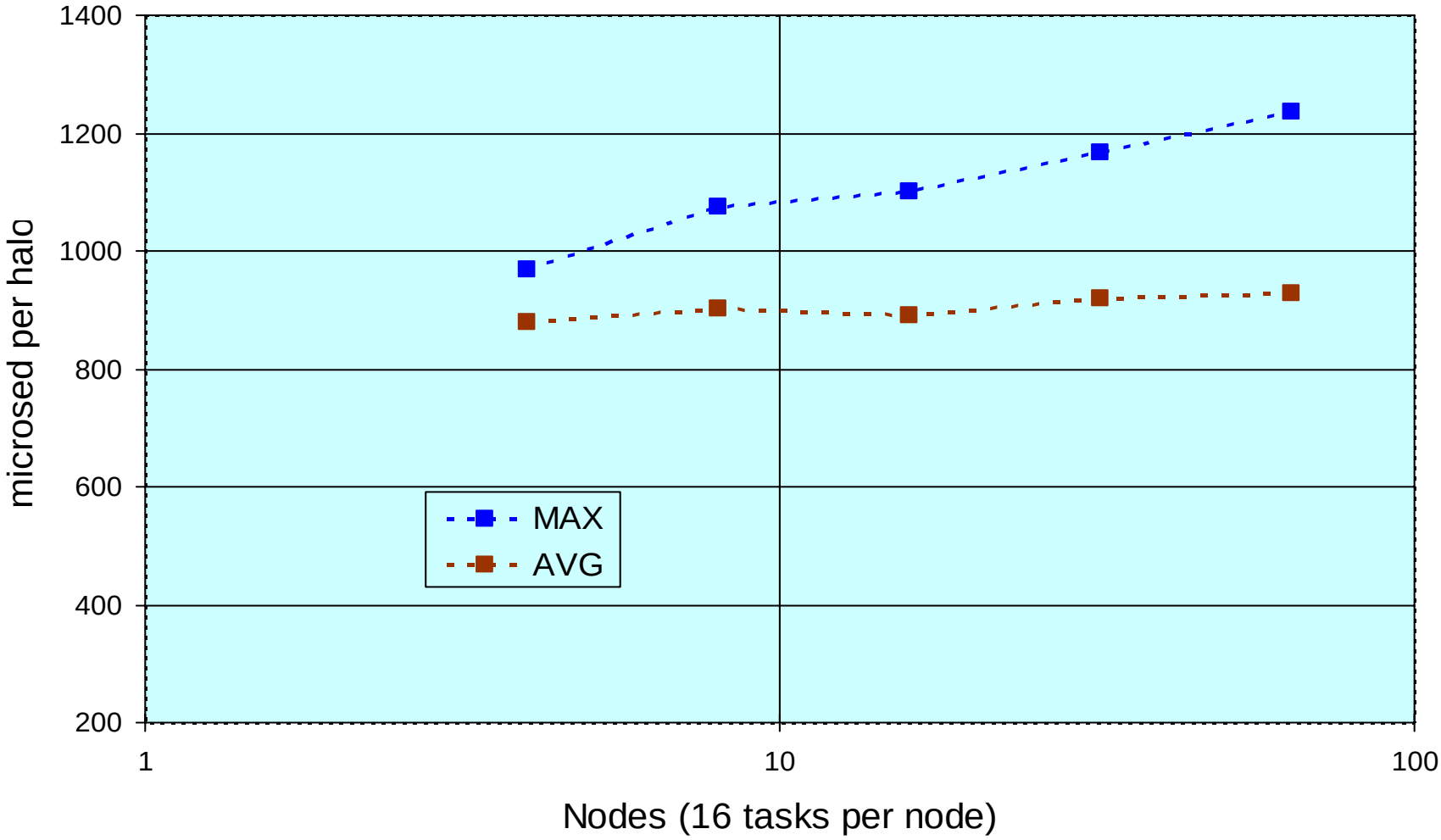

**2*(2*irecv+2*isend)**

**EAGER_LIMIT=32K**

# Time per call on P5 (64 nodes, 1024 tasks, 4K exch)



# Time for 40K exchanges on P5




**EAGER_LIMIT=32K**

**2*(2*irecv+2*isend)**

# Difference between AVG and MAX

- Could be due to
  - Queuing of packets in switch
  - Retransmits in across switch
  - Interference on switch from other programs
  - Interference with poe s/w
- Why does MAX/AVG increase with number of Nodes?
  - Each exchange takes as long as longest Task
  - If one in 200 exchanges takes double time then with 1000 tasks all exchanges will take double time

# Different Communication Patterns

1. 2*(irecv,isend)
   barrier
   irecv north, irecv south, isend north, isend south, wait
   irecv east, irecv west, isend east, isend west, wait

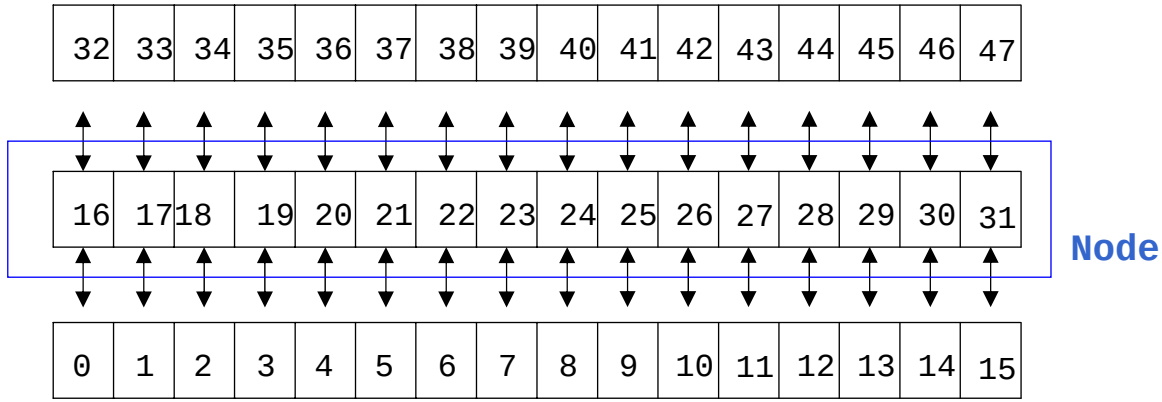
2. 4*irecv,4*isend
   barrier
   irecvs north/south/east/west irecvs
   isends north/south/east/west isends, wait

3. 4*isend, 4*recv
   barrier
   isends north/south/east/west
   recvs north/south/east/west, wait

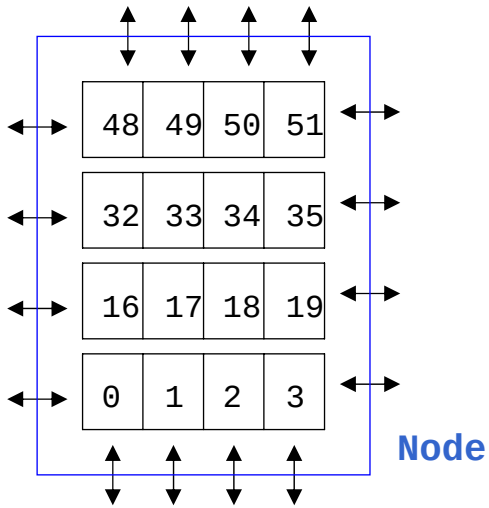
4. 4*isend, 4*recv, bunched tasks
   barrier
   isends north/south/east/west
   recvs north/south/east/west, wait

# Bunching tasks on node

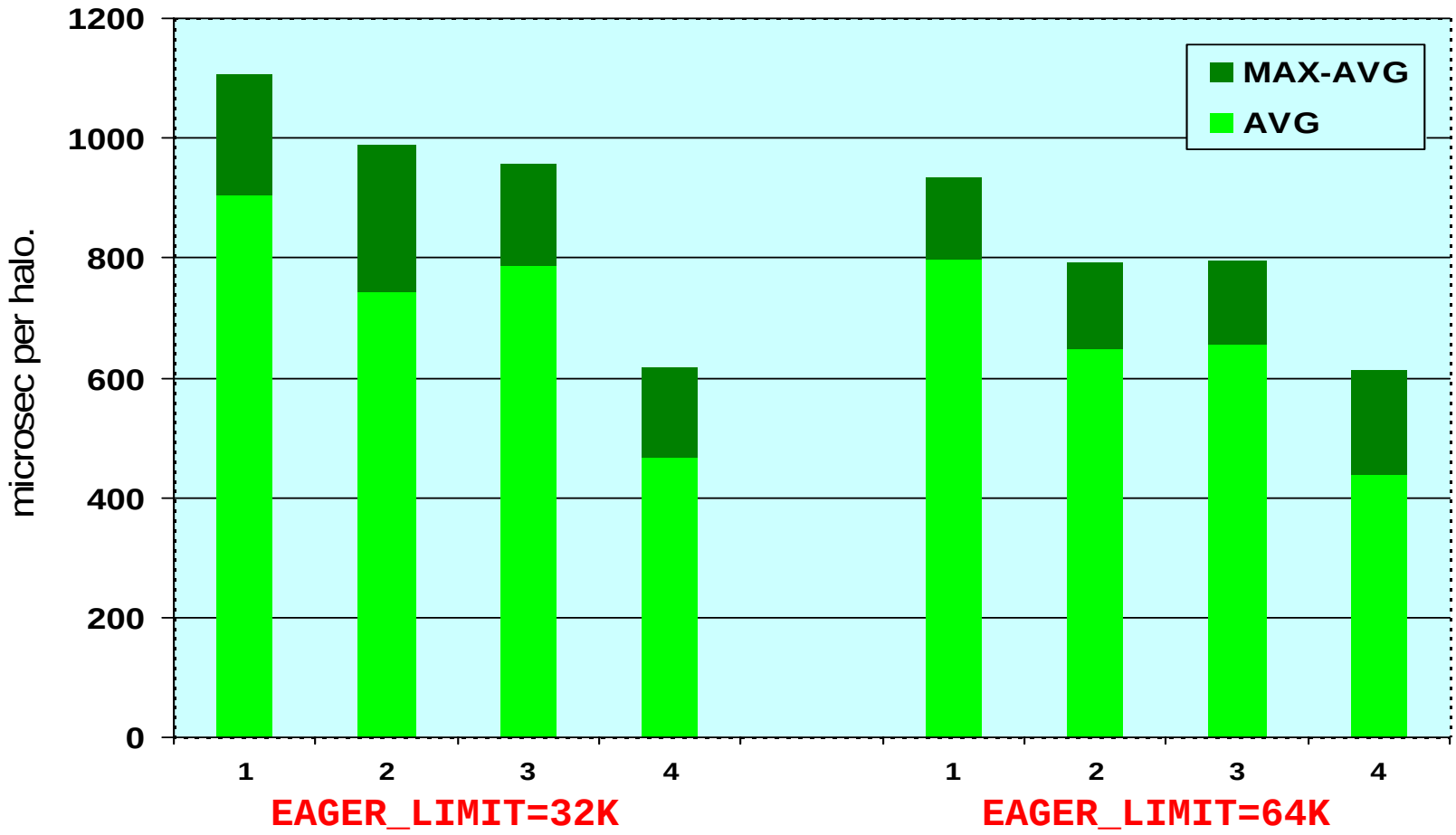
- Non bunched tasks
  - Off node communication = 32 send/recvs




- Bunched Tasks
  - Off node communication = 16 send/recvs



# Time for 40K exchanges on P5: 16 nodes, 256 tasks



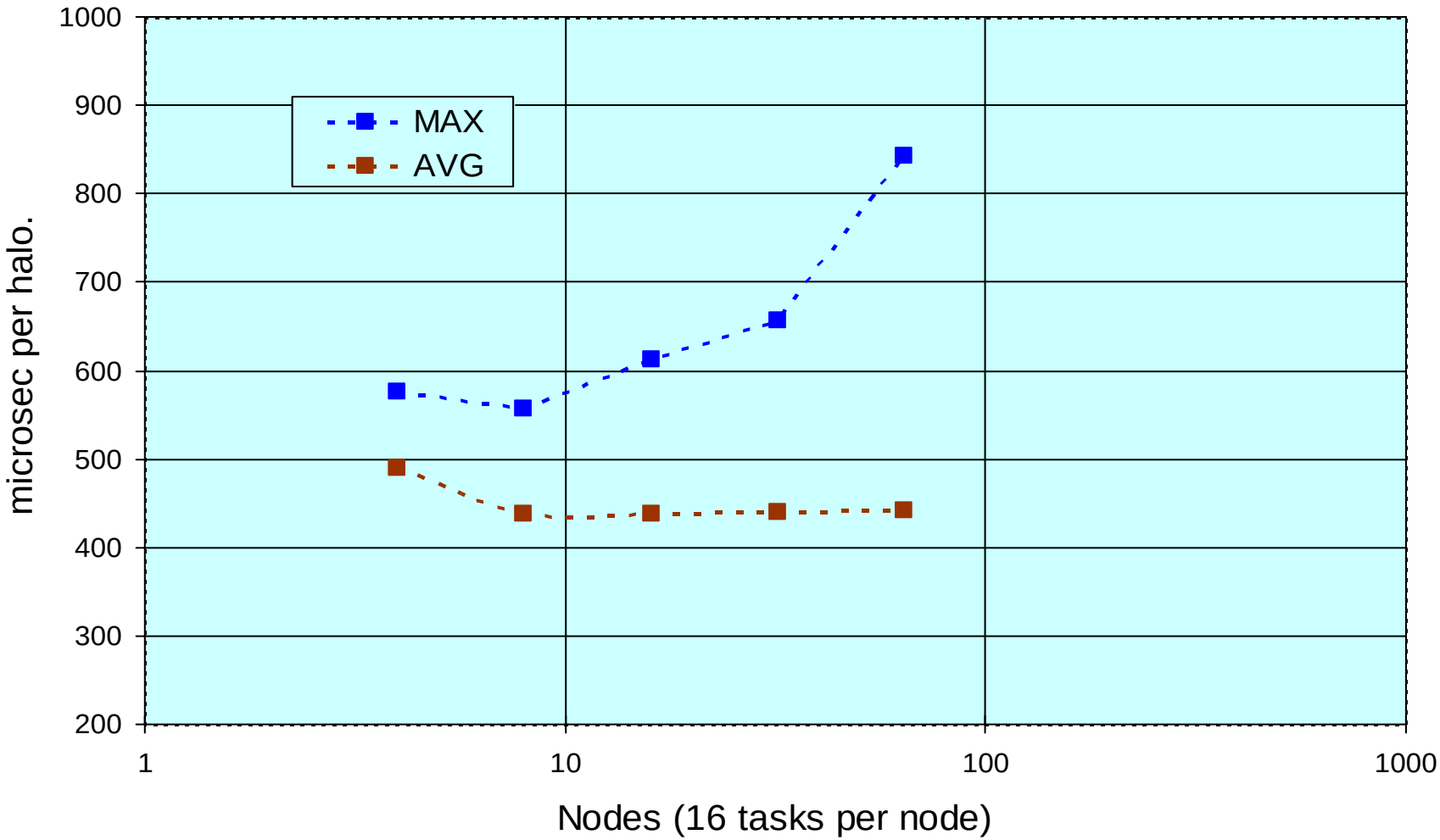

1: 2*(2*irecv,2*isend)

2: 4*irecv,4*isend

3: 4*isend,4*recv

4: 4*isend,4*recv bunched tasks

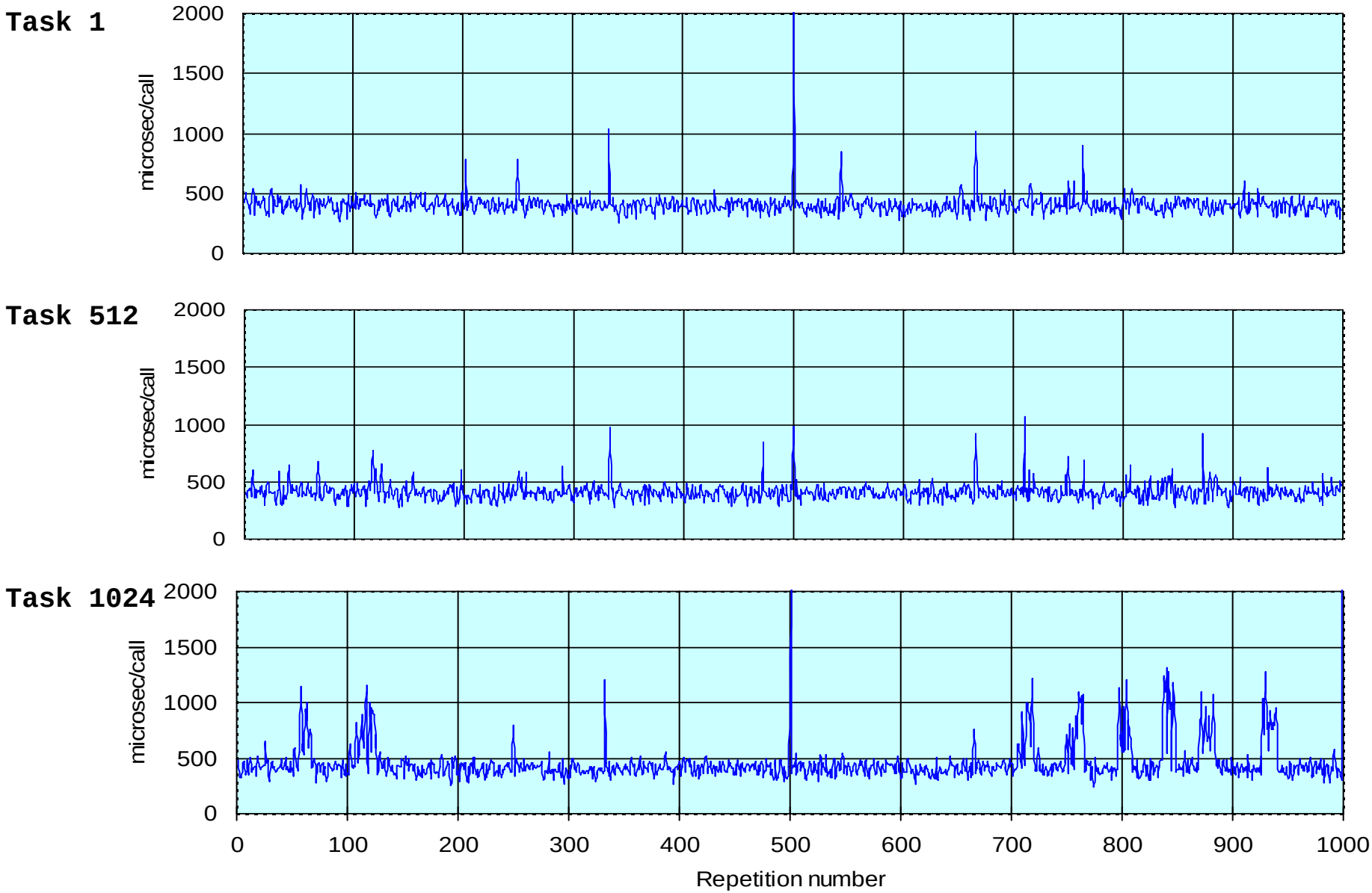
# Time for 40K exchanges on P5



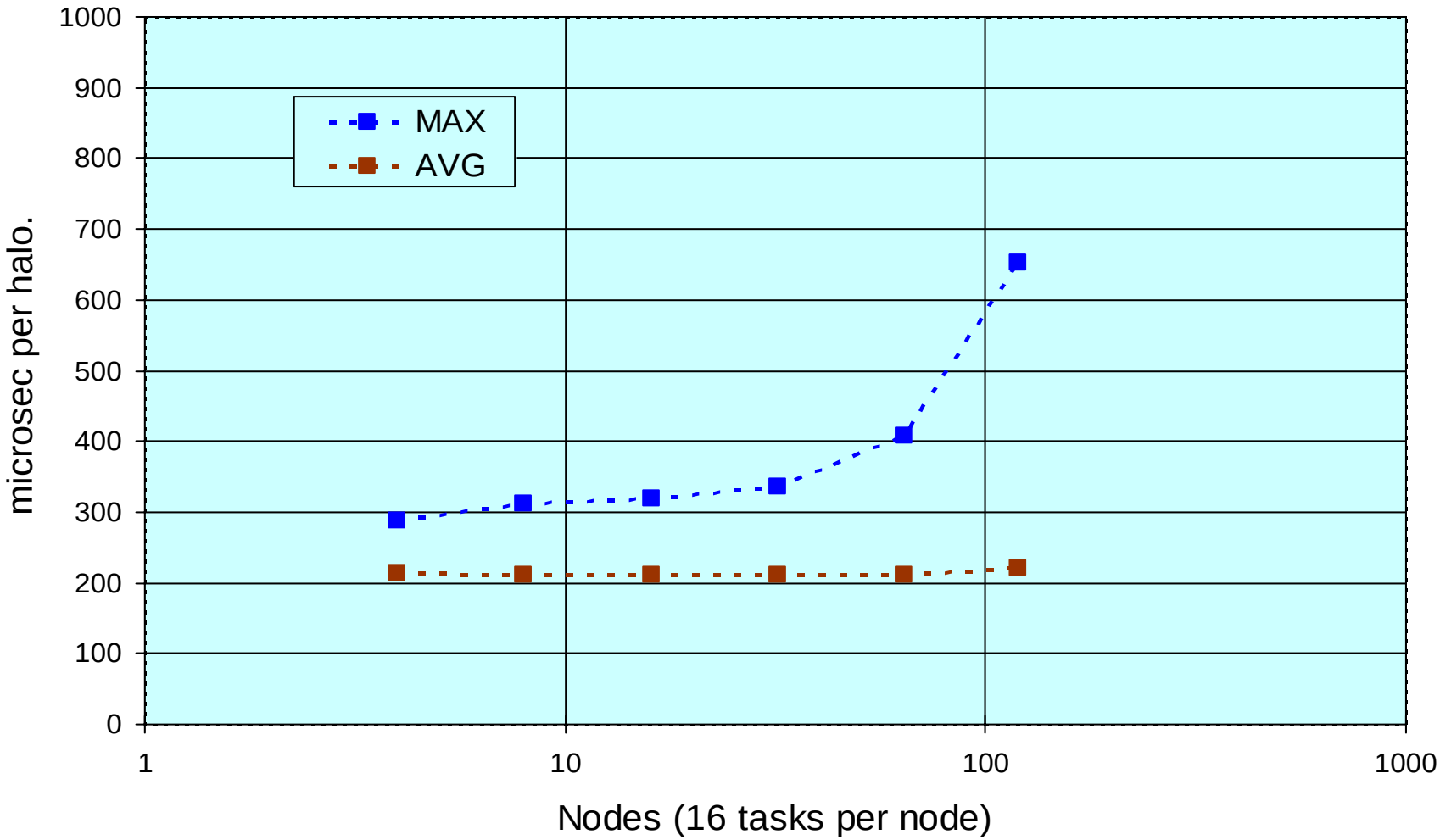

**EAGER_LIMIT=64K**

**4*isend,4*recv bunched tasks**

# Time per call on P5 (64 nodes, 1024 tasks, 40K exch)



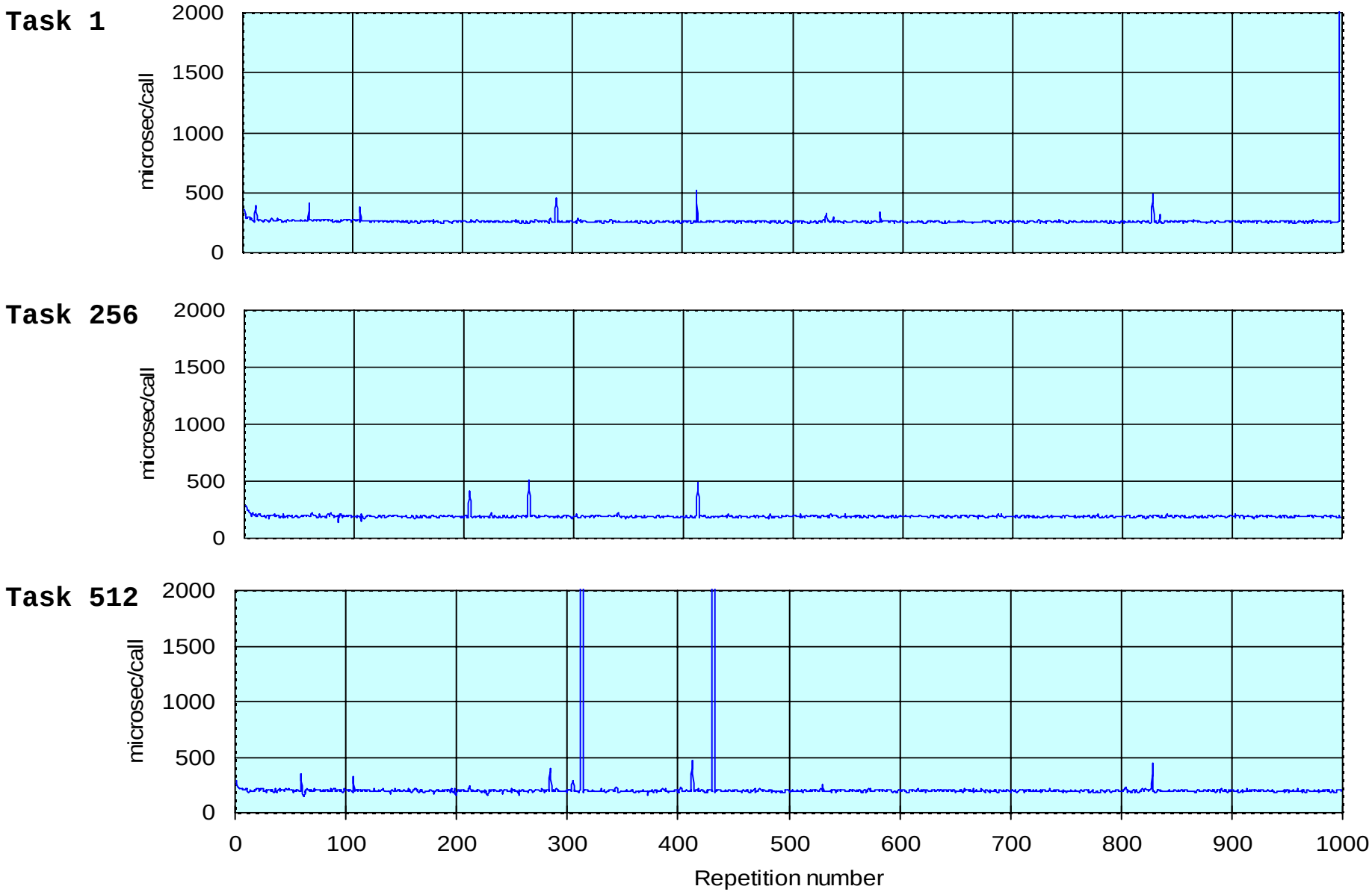
# Time for 40K halo exchanges on P6



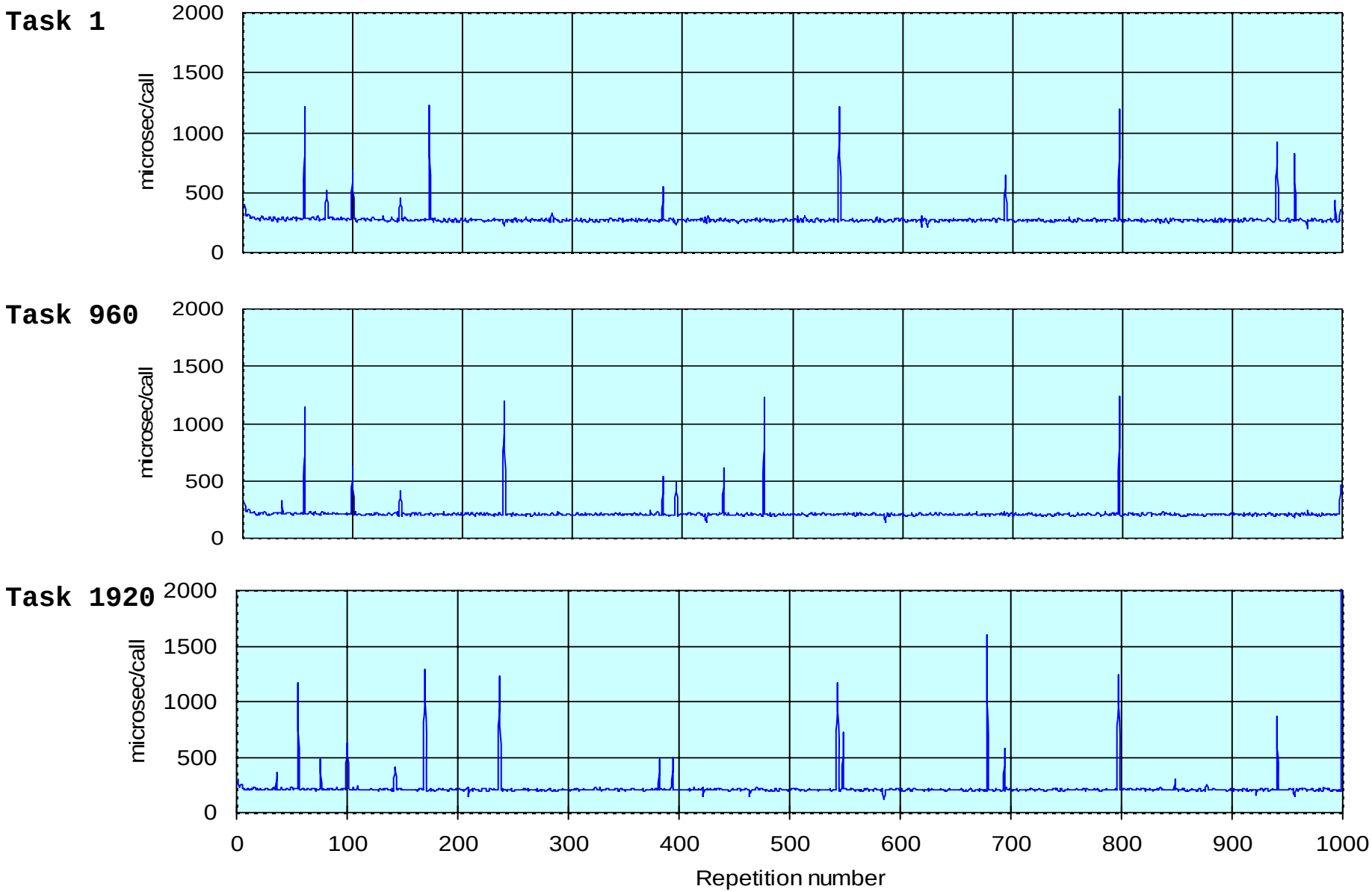

**EAGER_LIMIT=64K**

**4*isend,4*recv bunched tasks**

# Time per halo on P6 (32 nodes, 512 tasks, 40k exch)

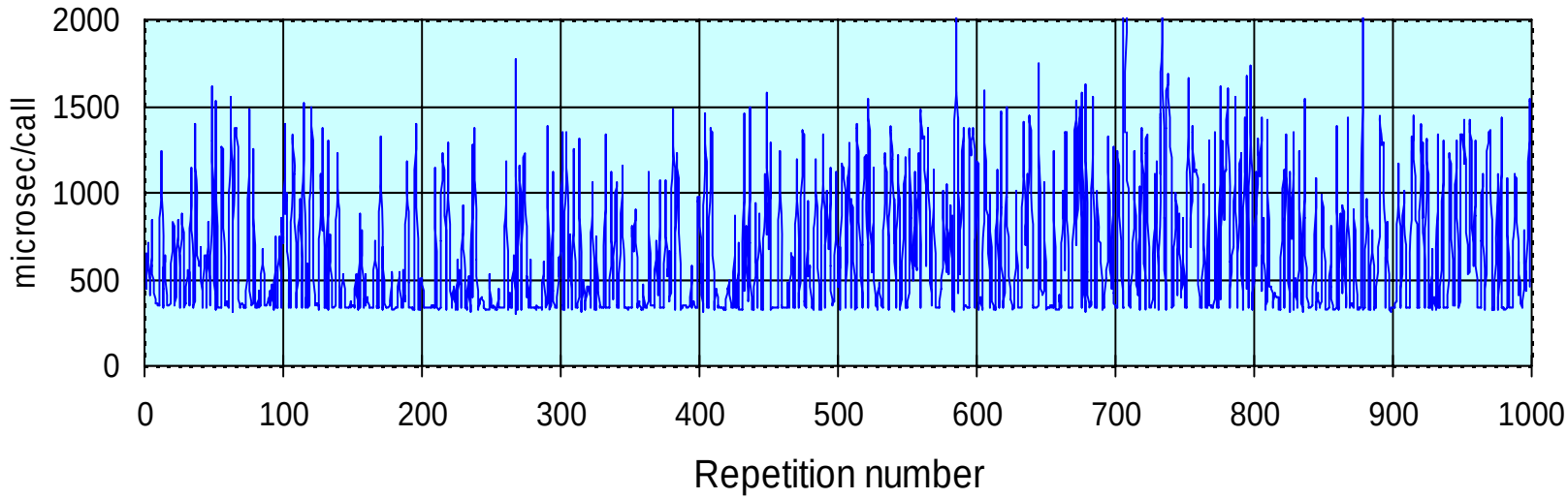


# Time per halo on P6 (120 nodes, 1920 tasks, 40k exch)



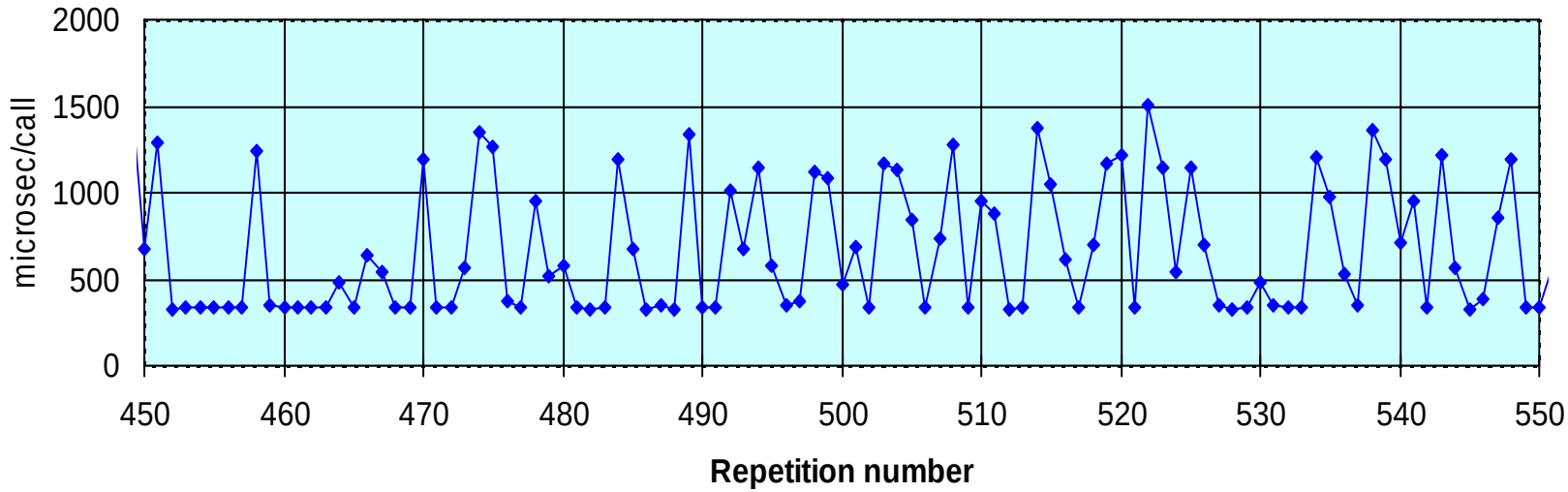
# Max time per halo on P6
# (120 nodes, 1920 tasks, 40k exch)

**All Tasks**



# Max time per halo on P6
# (120 nodes, 1920 tasks, 40k exch)

**All Tasks**



# Conclusions

- For Halo exchange:
  - Can get factor of 2 improvement by optimisation:
    - Combine exchanges
    - Set environment variables (e.g. MP_EAGER_LIMIT)
    - Communication pattern
    - Task placement (bunching)
  - What next?
    - Node allocation
    - Application changes (e.g. computation/communication overlap)
    - Reduce “jitter” – which will be done