# mpi - c - scaling analysis of all reduce

Experiment run date : March 30, 2024 post sunspot upgrade

### job script

```
NNODES=`wc -l < $PBS_NODEFILE`
RANKS_PER_NODE=12          # Number of MPI ranks per node
 CPU_BINDING=list:1-2:9-10:17-18:25-26:33-34:41-42:52-53:60-61:68-69:76-77:84-85:92-93
# STRACE_WRAPPER=/lus/gila/projects/CSC250STDM10_CNDA/kaushik/gitrepos/src-strace-analyser/strace-analyzer/strace-wrapper.sh
# LOGDIR=$PBS_O_WORKDIR/strace_1_$NNODES mpiexec   --env FI_CXI_DEFAULT_CQ_SIZE=16384  --env FI_CXI_OVFLOW_BUF_SIZE=8388608 --env FI_CXI_CQ_FILL_PERCENT=20 --np ${NRANKS} -ppn ${RANKS_PER_NODE} --cpu-bind  $CPU_BINDING  $STRACE_WRAPPER  ./test0
mpiexec   --env FI_CXI_DEFAULT_CQ_SIZE=16384  --env FI_CXI_OVFLOW_BUF_SIZE=8388608 --env FI_CXI_CQ_FILL_PERCENT=20 --np ${NRANKS} -ppn ${RANKS_PER_NODE} --cpu-bind  $CPU_BINDING   ./test0
```

## Summary

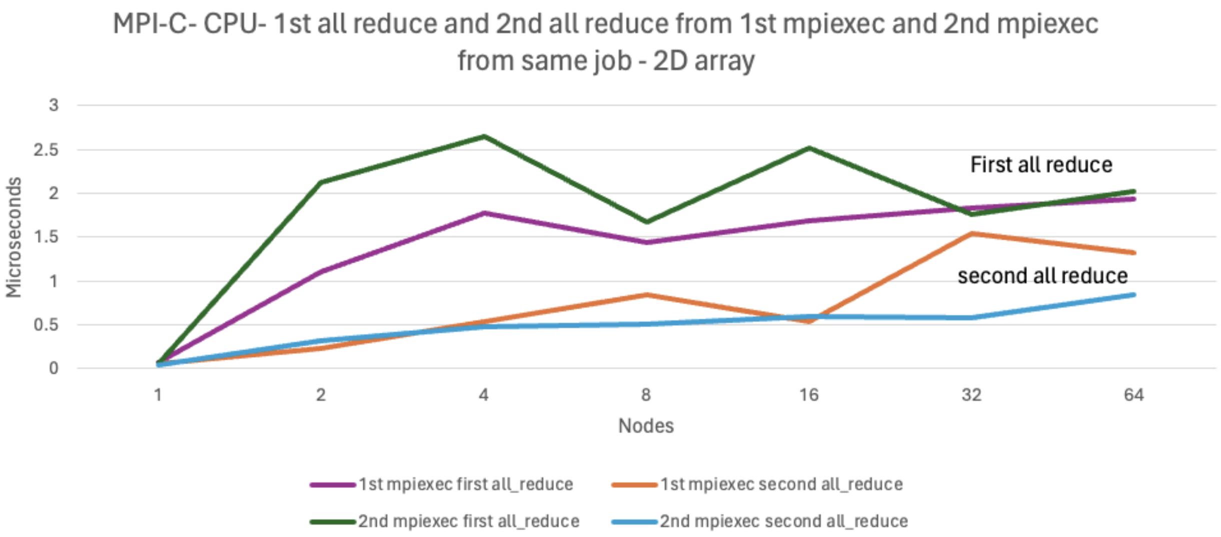
1. In this run, i used a 2d array.
2. Not a significant difference in the first and second mpiexec
3. the second all reduce was roughly 0.5x faster than the first.

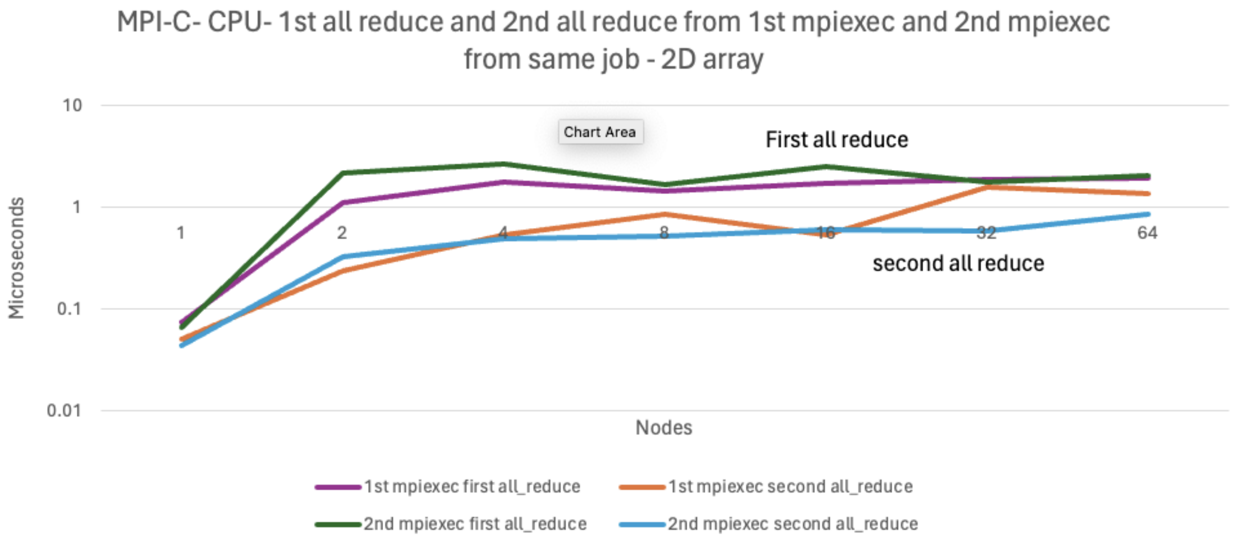
```
    int s_reduce[1024][1024];
    int r_reduce[1024][1024];

    MPI_Allreduce( s_reduce, r_reduce, 1024, MPI_INT, MPI_SUM, MPI_COMM_WORLD );

    MPI_Barrier( MPI_COMM_WORLD );
    t2 = MPI_Wtime();

       if ( rank == 0 )   printf("First all reduce time : %8.4lf \n", ( t2 - t1 ) * 1e6 / (double)1000 );


    MPI_Barrier( MPI_COMM_WORLD );
    t3 = MPI_Wtime();
```

## Results

| | | | 1st mpiexec first all_reduce | 1st mpiexec second all_reduce | 1st mpiexec diff 2 -1 | 2nd mpiexec first all_reduce | 2nd mp second all_redu |
|---|---|---|---|---|---|---|---|
| Kau_iter1 | pbs-script.o8987461 | NUM_OF_NODES=1 | 0.0742 | 0.0507 | 0.68 | 0.0647 | 0.043 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Kau_iter1 | pbs-script.o8987462 | NUM_OF_NODES=2 | 1.1103 | 0.2367 | 0.21 | 2.1246 | 0.3243 |
| Kau_iter1 | pbs-script.o8987463 | NUM_OF_NODES=4 | 1.7696 | 0.5308 | 0.30 | 2.648 | 0.4858 |
| Kau_iter1 | pbs-script.o8987464 | NUM_OF_NODES=8 | 1.4393 | 0.8404 | 0.58 | 1.6752 | 0.5079 |
| Kau_iter1 | pbs-script.o8987465 | NUM_OF_NODES=16 | 1.6868 | 0.535 | 0.32 | 2.5167 | 0.5918 |
| Kau_iter1 | pbs-script.o8987466 | NUM_OF_NODES=32 | 1.8343 | 1.5379 | 0.84 | 1.7647 | 0.584 |
| Kau_iter1 | pbs-script.o8987467 | NUM_OF_NODES=64 | 1.9295 | 1.3292 | 0.69 | 2.0279 | 0.8472 |
| | | | | | | | |





## ALCF MPI ALL reduce benchmark

## job script

```
NNODES=`wc -l < $PBS_NODEFILE`
RANKS_PER_NODE=12          # Number of MPI ranks per node
NRANKS=$(( NNODES * RANKS_PER_NODE ))
CPU_BINDING=list:1-2:9-10:17-18:25-26:33-34:41-42:52-53:60-61:68-69:76-77:84-85:92-93
# STRACE_WRAPPER=/lus/gila/projects/CSC250STDM10_CNDA/kaushik/gitrepos/src-strace-analyser/str
ace-analyzer/strace-wrapper.sh

# LOGDIR=$PBS_O_WORKDIR/strace_1_$NNODES mpiexec  --env FI_CXI_DEFAULT_CQ_SIZE=16384  --env FI
_CXI_OVFLOW_BUF_SIZE=8388608 --env FI_CXI_CQ_FILL_PERCENT=20  --np ${NRANKS} -ppn ${RANKS_PER_
NODE} --cpu-bind  $CPU_BINDING  $STRACE_WRAPPER  ./collectives


mpiexec  --env FI_CXI_DEFAULT_CQ_SIZE=16384  --env FI_CXI_OVFLOW_BUF_SIZE=8388608 --env FI_CXI
_CQ_FILL_PERCENT=20  --np ${NRANKS} -ppn ${RANKS_PER_NODE} --cpu-bind  $CPU_BINDING   ./collec
tives
date
```

```
char *s_reduce, *r_reduce, *b_bcast;
s_reduce = (char *)malloc( N_AllMax * sizeof( char ) );
r_reduce = (char *)malloc( N_AllMax * sizeof( char ) );
```

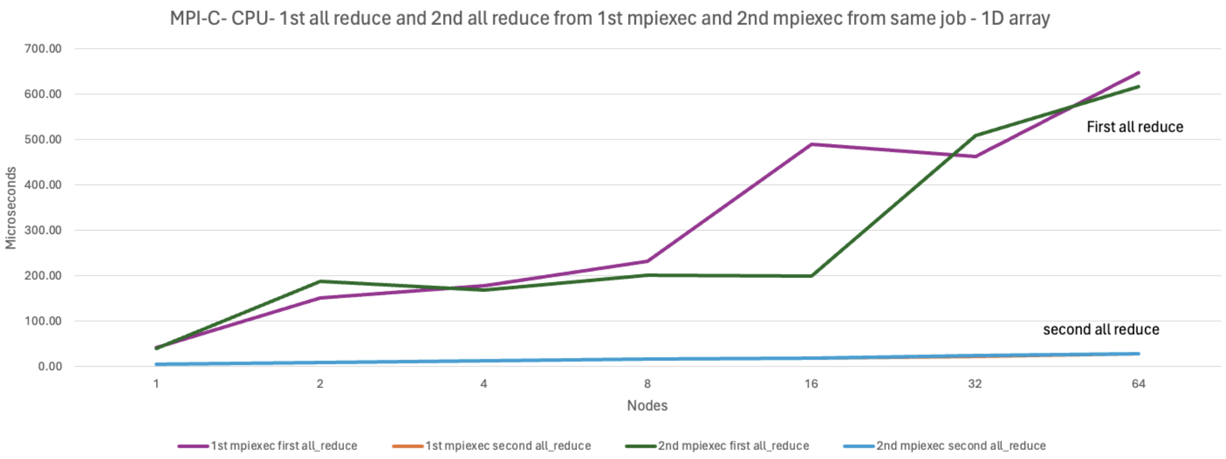
```
MPI_Barrier( MPI_COMM_WORLD );
t1 = MPI_Wtime();
MPI_Allreduce( r_reduce, s_reduce, 8, MPI_CHAR, MPI_SUM, MPI_COMM_WORLD );
MPI_Barrier( MPI_COMM_WORLD );
t2 = MPI_Wtime();
if ( rank == 0 ) printf( "First call: Allreduce %4d B COMM_WORLD, us: %8.4lf\n", 8, ( t2 - t1
) * 1e6 );
```

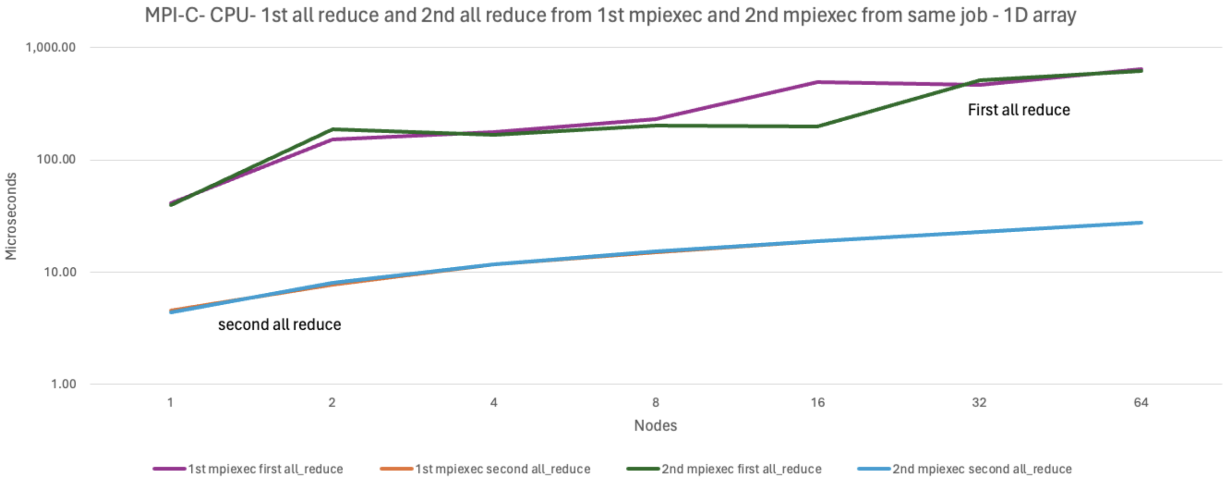
## summary

1. In the next run i used a 1d array for all reduce
2. Not a significant difference in the first and second mpiexec
3. the 2nd all reduce was ( 9x - 23x ) on scaling faster than the 1st all reduce.

## results

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | 1st mpiexec first all_reduce | 1st mpiexec second all_reduce | 1st mpiexec diff 2 -1 | 2nd mpiexec first all_reduce | 2nd mp second all_redu |
| vit_iter1 | pbs-script.o8987402 | NUM_OF_NODES=1 | 40.9298 | 4.5251 | 9.045059778 | 39.815 | 4.3828 |
| vit_iter1 | pbs-script.o8987403 | NUM_OF_NODES=2 | 150.7238 | 7.7647 | 19.41141319 | 187.2753 | 7.9606 |
| vit_iter1 | pbs-script.o8987404 | NUM_OF_NODES=4 | 177.8282 | 11.6205 | 15.30297319 | 168.312 | 11.6645 |
| vit_iter1 | pbs-script.o8987405 | NUM_OF_NODES=8 | 232.2414 | 15.0524 | 15.42886184 | 200.9109 | 15.1711 |
| vit_iter1 | pbs-script.o8987406 | NUM_OF_NODES=16 | 490.1307 | 18.6681 | 26.25498578 | 198.8156 | 18.6603 |
| vit_iter1 | pbs-script.o8987407 | NUM_OF_NODES=32 | 461.6156 | 22.5845 | 20.43948726 | 509.1253 | 22.7425 |
| vit_iter1 | pbs-script.o8987408 | NUM_OF_NODES=64 | 647.0799 | 27.4639 | 23.56110749 | 616.5575 | 27.4842 |
| | | | | | | | |


MPI-C- CPU- 1st all reduce and 2nd all reduce from 1st mpiexec and 2nd mpiexec from same job - 1D array
Microseconds
700.00
600.00
500.00
400.00
300.00
200.00
100.00
0.00
1
2
4
8
16
32
64
Nodes
First all reduce
second all reduce
1st mpiexec first all_reduce
1st mpiexec second all_reduce
2nd mpiexec first all_reduce
2nd mpiexec second all_reduce


MPI-C- CPU- 1st all reduce and 2nd all reduce from 1st mpiexec and 2nd mpiexec from same job - 1D array
Microseconds
1,000.00
100.00
10.00
1.00
1
2
4
8
16
32
64
Nodes
First all reduce
second all reduce
1st mpiexec first all_reduce
1st mpiexec second all_reduce
2nd mpiexec first all_reduce
2nd mpiexec second all_reduce