Masterpraktikum Scientific Computing

High Performance Computing Tutorial 3

Session 3: MPI, CG

Authors

Mantosh Kumar (Matriculation number : 03662915)
João Trindade (Matriculation number : 03673251)

Course of Study: Master of Science Informatics

Question 1 - Amdahl’s Law

$$Sp = \frac{1}{s+\frac{1-s}{p}} \qquad Eff = \frac{Sp}{p}$$

$$s = 10\% \quad Sp = \frac{1}{0.1+\frac{0.9}{p}}$$

$$Eff = 70\%$$

$$0.7 = \frac{1}{0.1p+0.9}$$

$$0.07p + 0.63 = 1$$

$$p = 5.28$$

Therefore, to achieve an efficiency of 70% or higher, only 5 processors, or less, can be used.

Question 1 - Alternatives to Amdahl’s Law

The measure scaled speedup was proposed by Gustafson,to obtain a more realistic speedup measure.

This measure is based on the fact that in some cases the sequential part does not increase with the problem size.

If T’P is the parallel processing time for P processors of the parallel component, then the time that a sequential computer would take to execute the program is TS+P.T’P.

Therefore, we obtain a new Speedup Formula.

$$ScaledSpeedup = \frac{T_S + P.T_P^{'}}{T_S + T_P^{'}}$$

Question 2 - Infiniband

Infiniband is a network communication standard commonly used in clusters. This technology offers high-bandwidth and low latency communications.

The FDR14 standard specifies a 14Gb/s speed per comunication lane. Since most Infiniband ports are 4-Lanes, this would result in a 54Gb/s speed.
[FDR Infiniband Fact Sheet]

Question 2 - MPI Benchmarks

Three types of benchmarks:

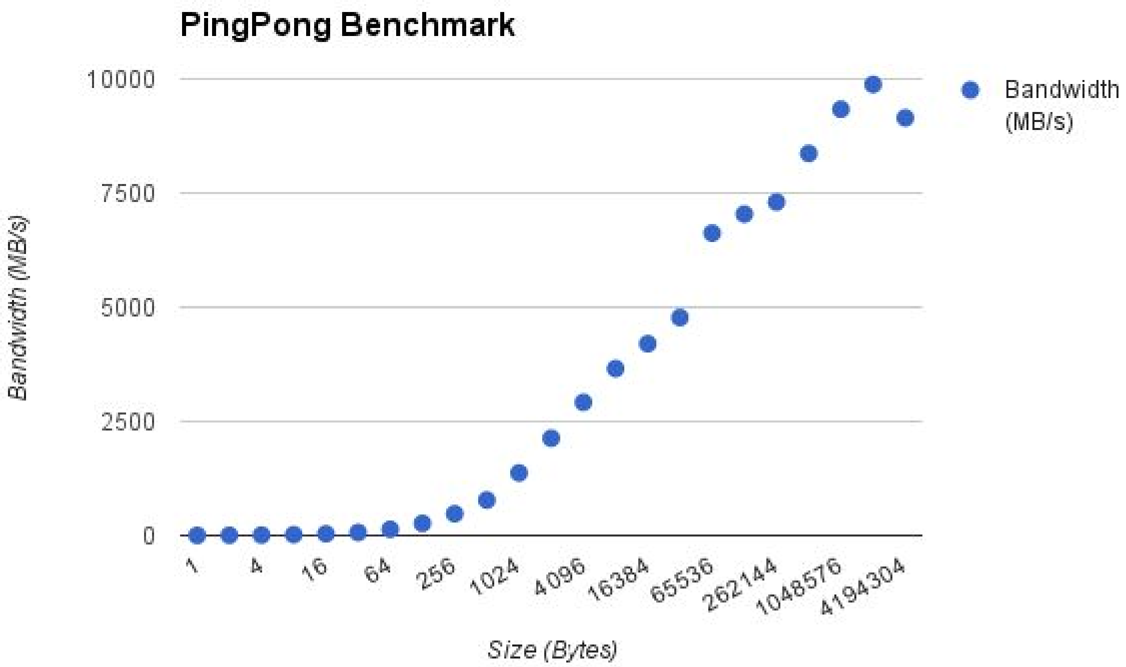
Single Transfer (eg. Ping Pong) : Only two processes communicate with each other.

Parallel Transfer (eg. sendrecv): More than 2 processes are involved in a communication. 1 to Many for example.

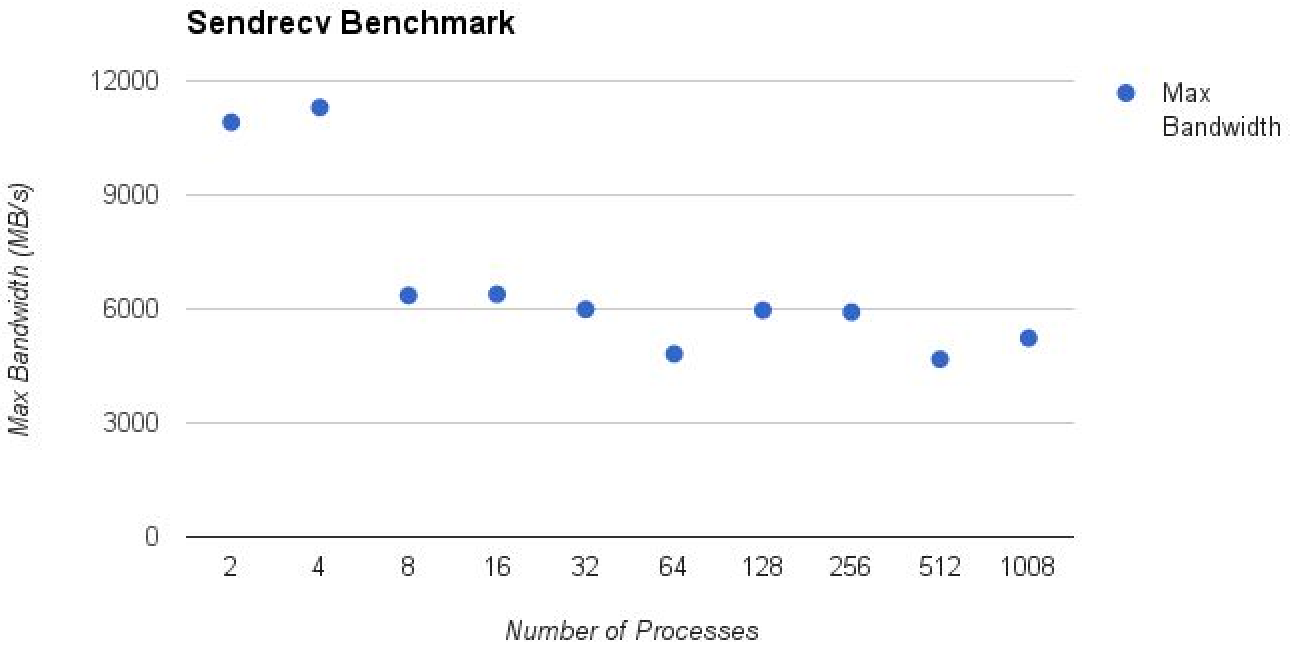
Collective Benchmark (eg. Reduce): Benchmark of a MPI Collective Operation

Benchmarks are also organized by the MPI Version when that feature was added. For instance, Non-Blocking Collective Operations Benchmarks are available under the MPI-3 Benchmarks

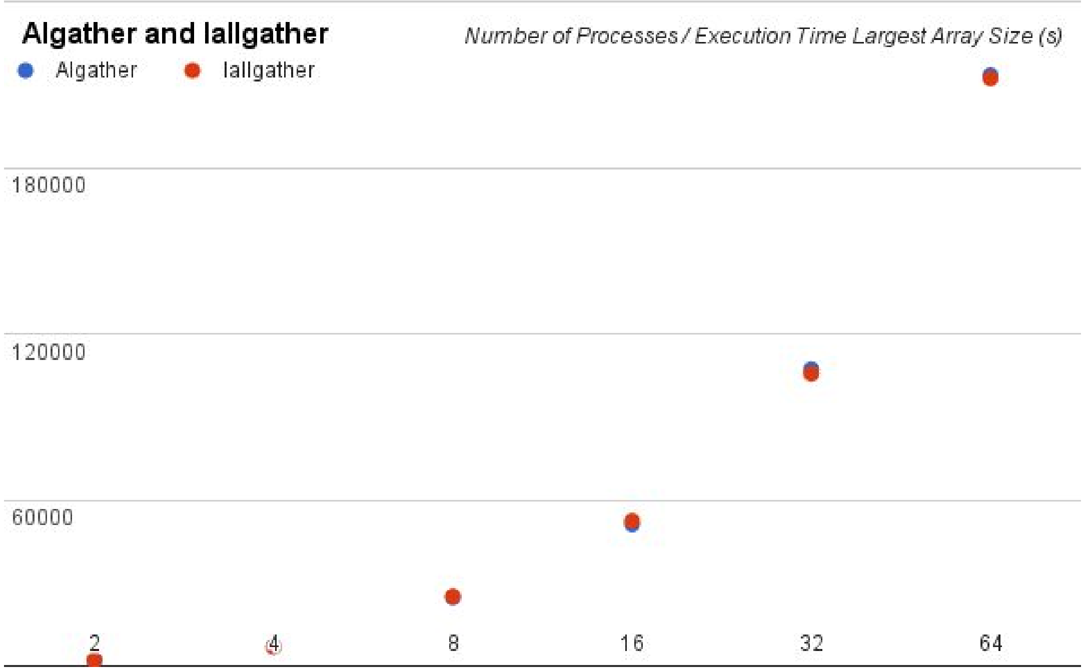
Question 2 - Benchmark Results - Single Transfer



Question 2 - Benchmark Results - Parallel Transfer



Question 2 - Benchmark Results - Collective Operation



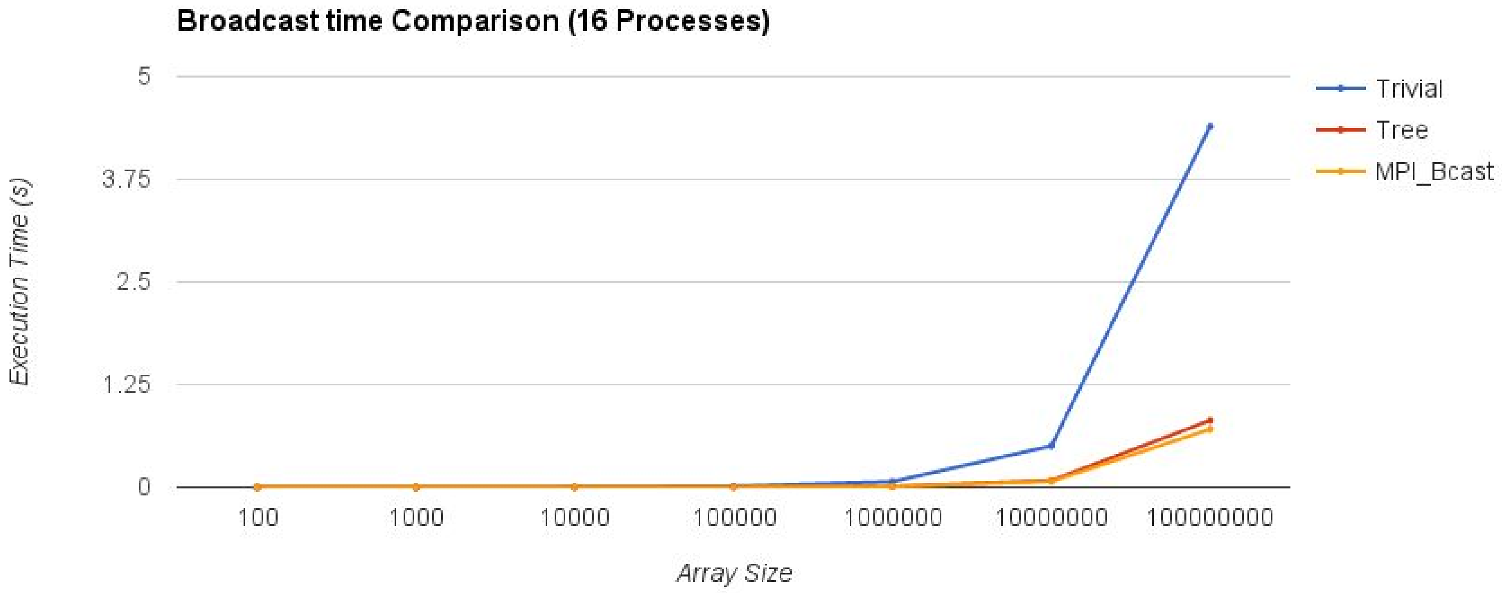
Question 3

Both Trivial and Tree implementations require a total of P mpi-messages.

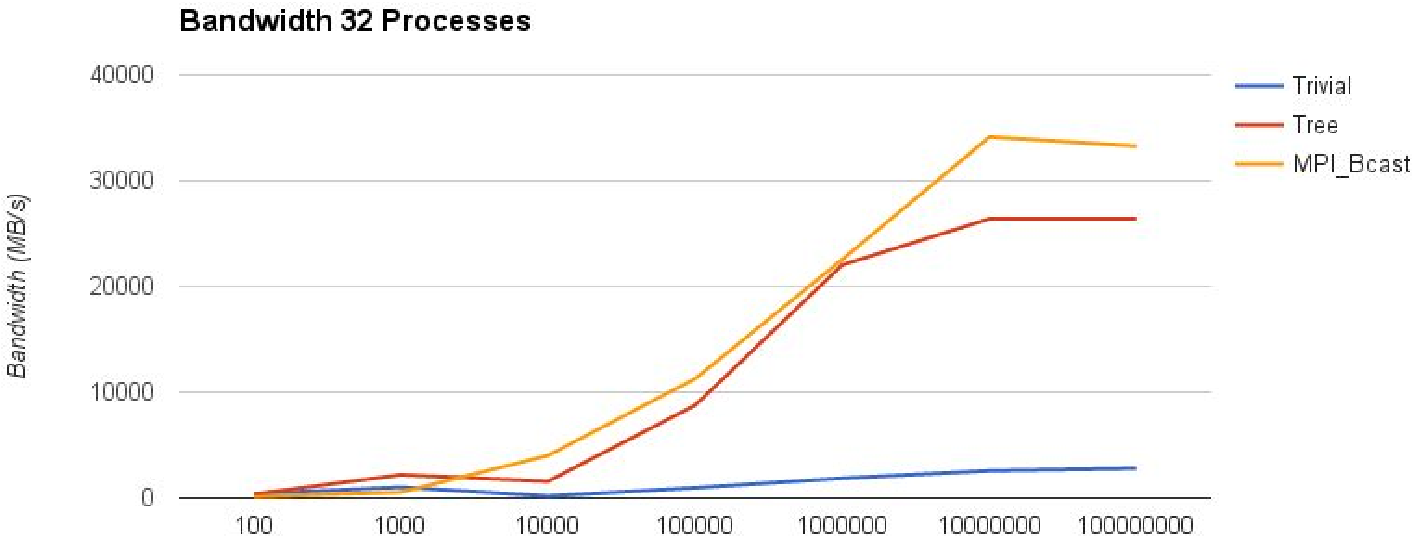
In the Trivial implementations, every message is part of the “critical path” whereas in the Tree implementations, the critical part is composed of only log(P) messages.

We concluded that the MPI BCast implementation is around 20% better than our implementation of the Tree-Broadcast

Question 3


Broadcast time Comparison (16 Processes)
Execution Time (s)
5
3.75
2.5
1.25
0
100
1000
10000
100000
1000000
10000000
100000000
Array Size
Trivial
Tree
MPI_Bcast

Question 3 - Results



Question 4 (CG method)

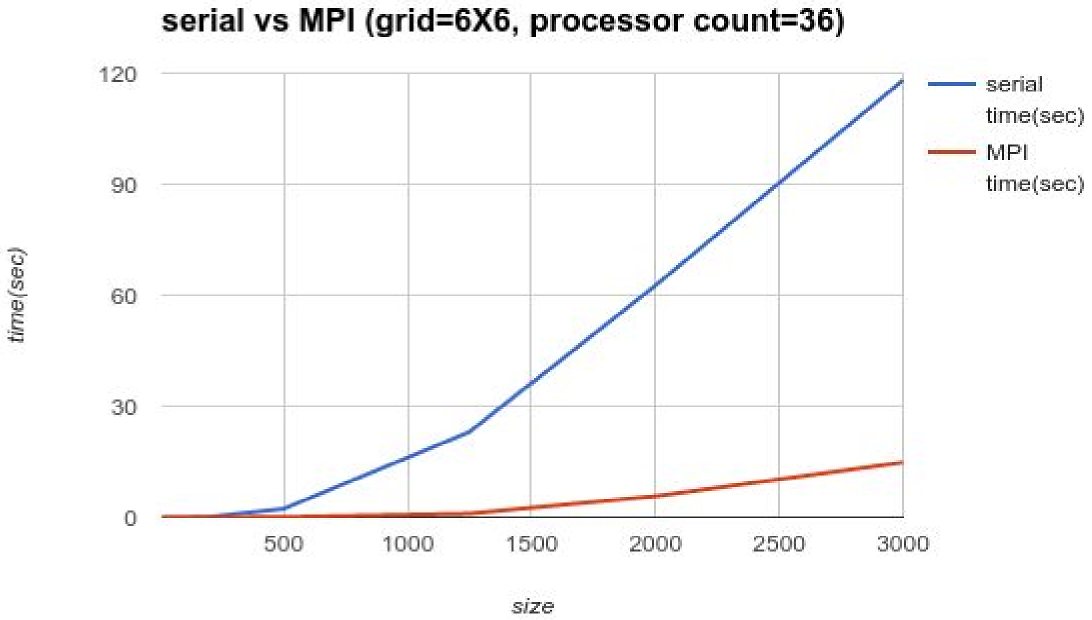
Split up the domain of all inner points into parts of equal size.

Changed the parameters of the program: Asking number of grid points per dimension instead of meshwidth.

Used a constant maximum number of iteration = 1000

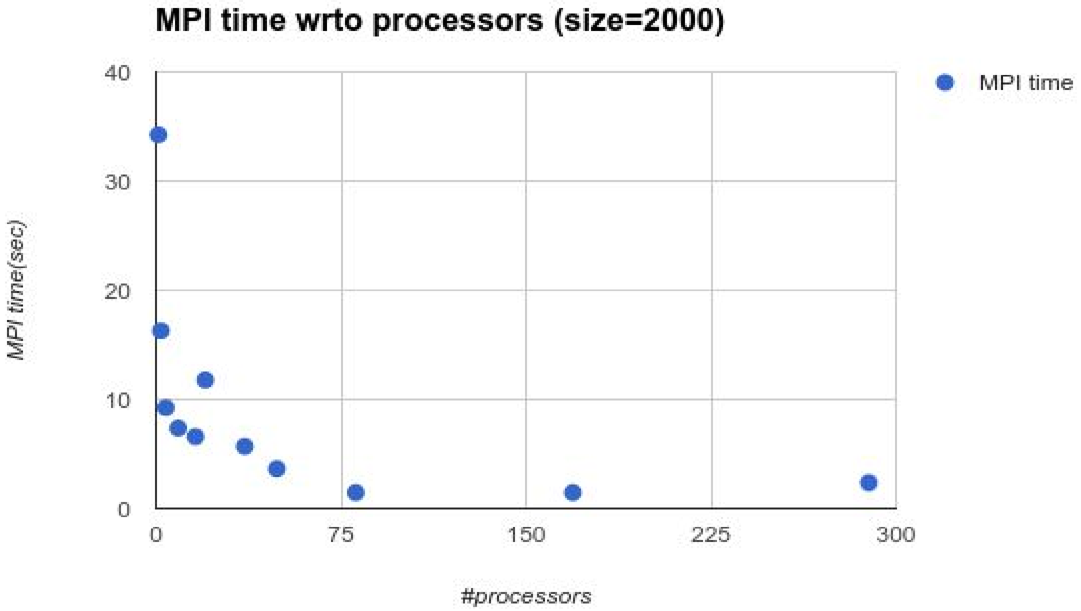
Question 4: Result

Measuring runtime depending on size

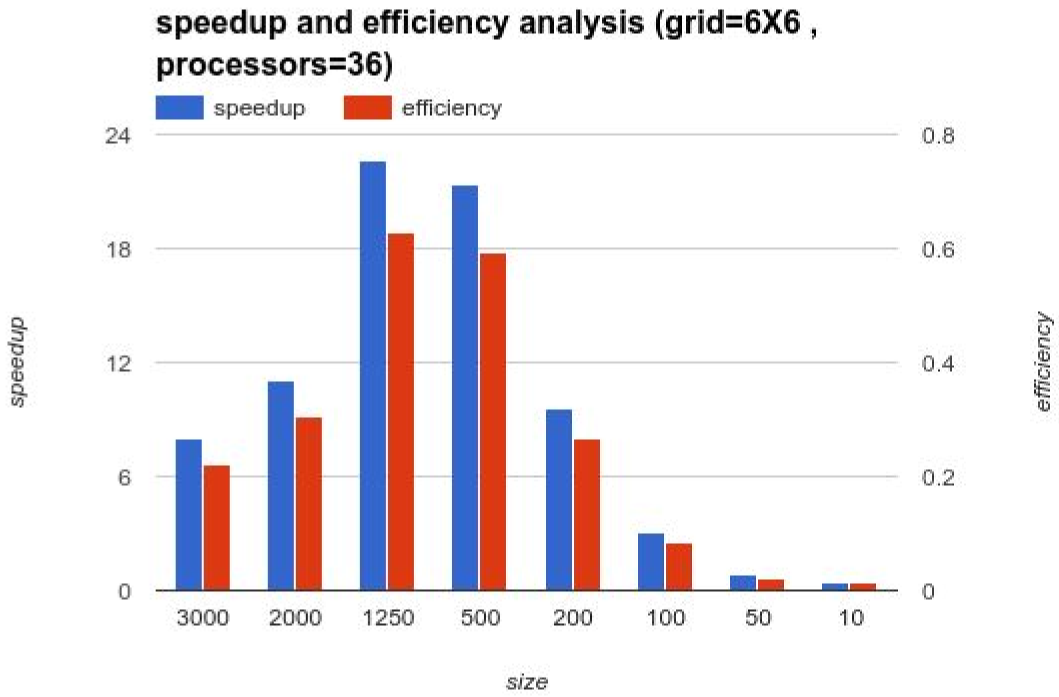


Question 4: Result

Measuring scaling behaviour with constant size and different number of processors



Question 4: Result



Thank you!