PARALLEL AND DISTRIBUTED PROGRAMMING

ASSIGNMENT 1

Group 50

Jiahao LU, You WU, Zonghao LU

13 April 2019

## Problem Formulation

The task is to compute in parallel the two-dimensional integral

The exact answer is .

## Solution Method

The two-dimensional integral can be approximated by the midpoint rule as follows:

where is the number of intervals and .

The intervals are divided into p processors. A partial sum in x-dimension is computed on each processor, and then reduced to all processors to get the inner integral by using MPI\_Allreduce function. The outer integral in y-dimension is reduced to the master processor by MPI\_Reduce function. The master processor also measures the computation time and output the number of intervals , the result and the time.

## Experiments

1. **Testing environment**

Host: vitsippa.it.uu.se

CPU: AMD Opteron (Bulldozer) 6282SE, 2.6 GHz, 16-core, dual socket

OS: Scientific Linux release 6.10 (Carbon)

GCC: gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-23)

MPI: mpirun (Open MPI) 1.8.1

1. **Testing method**

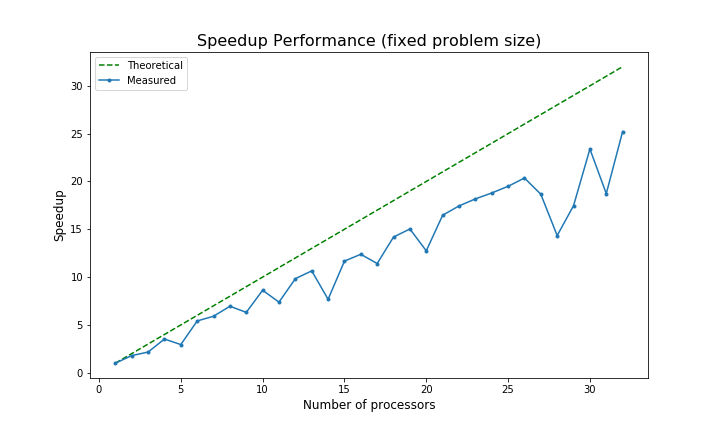
Each time is tested 4 times. Except for the first result, the measured time of the rest 3 testing is averaged to get the mean value.

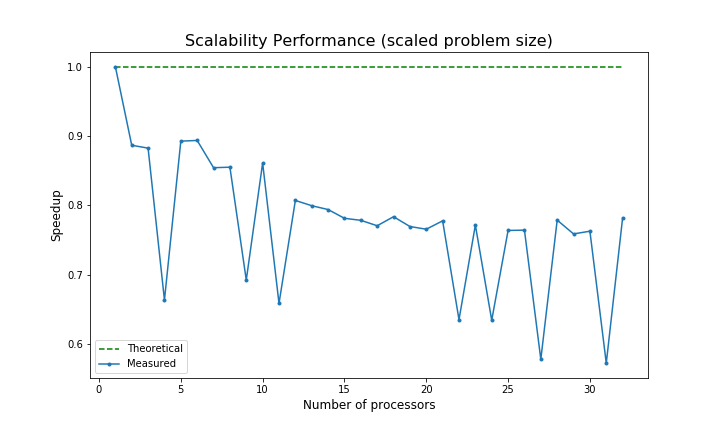
The speedup performance is tested for  intervals, running from 1 to 32 processors. The output time is measured by MPI\_Wtime() function on the master processor. Only the parallelised computation time is measured.

The scalability is tested when number of processors are increased from 1 to 32, while number of intervals are increased simultaneously in a way that the interval load per processors is kept to .

## Results and Discussion

The tested parallelisation performance and the scalability for both fixed problem size and scaled problem size are shown as follows.





Since only the parallelised computation time is measured, the theoretical speedup for fixed problem size should equal to the number of processors. The measured speedup for fixed problem size curve is approximately linear but lower than the theoretical line. The ideal speedup for scaled problem size should be 1 since the load per individual processor is kept large enough and approximately constant.

The measured curves are not ideal because of the overheads in communication between processors. The fluctuation might be due to the cache usage as number of intervals increasing, as well as the disturbance on the host server.