Exercise 06

HPL benchmark using MKL multithread library

Exercise 6.1:

The Linpack benchmark called the Highly Parallel Computing Benchmark (HPL) is a software package that solves a random dense linear system in double precision (64 bits) arithmetic on distributed-memory computers. This benchmark is known for being the one often used for the TOP500 report[1].

The HPL package provides a testing and timing program to quantify the accuracy of the obtained solution as well as the time needed to compute it[2]. This benchmark is widely used to validate the system performance and stability, measuring the floating point execution rate.

The first step into running the benchmark consists in tuning the input data file "HPL.dat". The challenge is to find the optimal parameters combination in order to access the performance of the machine.

As regards to the problem size (matrix dimension N) to be considered, the largest problem size fitting in memory is aimed. The amount of memory used by HPL is essentially the size of the coefficient matrix. The block size NB, on the other hand, is used by HPL for the data distribution as well as for the computational granularity. From a data distribution point of view, the smallest NB, the better the load balance. Concerning the process grid (P x Q), the choice of values depends on the physical interconnection network available. In general, P and Q should be approximately equal, with Q slightly larger than P.[3] The file used in this work (input) with the best configuration is presented in appendix A.

In order to understand how the configuration of the input file with MPI processes, number of threads, P, and Q affect the access of performance of the machine, many different inputs were tested. The results are presented in table 1.

Table 1: Output from xhpl executable for different configurations

MPI	Thread	Run 1 Gflops	Run 2 Gflops	Run 3 Gflops	Mean Gflops	Std Gflops	P	Q	N	NB
1	20	26,99	26,89	26,93	26,94	0,05	1	1	64521	256
2	10	51,26	52,00	51,24	51,50	0,43	1	2	64521	256
4	5	101,70	101,50	101,70	101,63	0,12	2	2	64521	256
5	4	123,60	124,00	123,80	123,80	0,20	1	5	64521	256
10	2	227,80	226,70	225,50	226,67	1,15	2	5	64521	256
20	1	420,40	423,80	413,80	419,33	5,08	4	5	64521	256

From table 1 one can notice that the best configuration, that gives us the higher performance, is the last one: with 20 MPI processes, 1 thread, P = 4, Q = 5, N = 64521, and NB = 256. One output is presented in appendix B.

1 www.top500.org/

2 http://www.netlib.org/benchmark/hpl/

3 http://www.netlib.org/benchmark/hpl/faqs.html

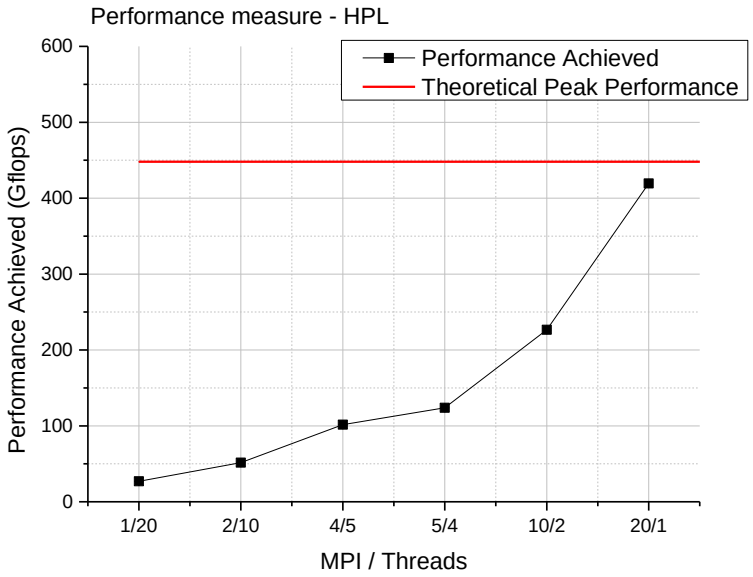
Exercise 06

It is known that the theoretical peak performance of the machine may be computed by the expression[4] (1):

Node performance in GFLOPS = (CPU speed in GHz) x (number of CPU cores) x (CPU instruction per cycle) x (number of CPUs per node). (1)

From the Ulysses characteristics, the theoretical peak performance (node) could be computed as being 448 GFLOPS.

Considering this result, graph 1 could be plotted, in which the performance achieved by different parameter configurations is compared with the theoretical peak performance of the machine.



Graph 1: Performance achieved for different configurations of MPI processes and number of threads - xhpl executable compiled with mpicc.

Exercise 6.2:

In order to analyze the influence of the compiler on the performance achieved, considering the aforementioned best configuration, it was possible to construct table 2, in which the output of different precompiled Intel executables are compared with the best previous output (xhpl compiled with mpicc).

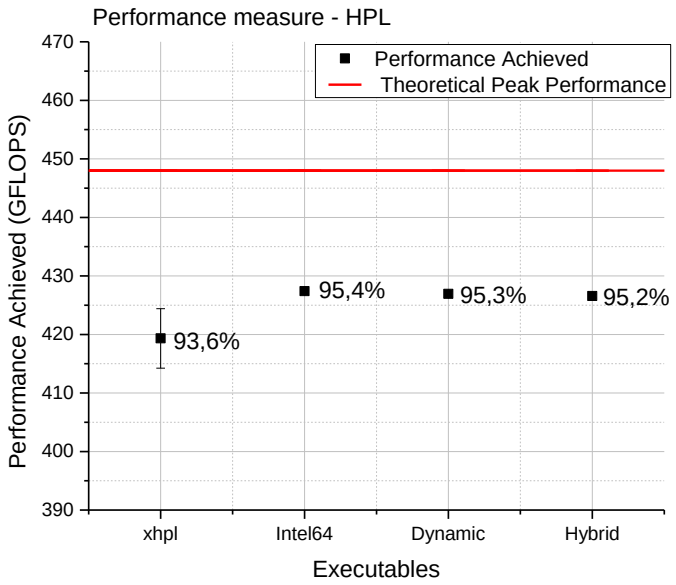
Table 2: Comparison among different executables for the best configuration

Executables	Run 1 Gflops	Run 2 Gflops	Run 3 Gflops	Mean Gflops	Std Gflops	% of T. Peak Performance
xhpl	420,40	423,80	413,80	419,33	5,08	93,6
xhpl_intel64	427,72	427,58	426,89	427,40	0,44	95,4
xhpl_intel64_dynamic	427,01	427,38	426,40	426,93	0,49	95,3
xhpl_hybrid_intel64	426,48	426,93	426,32	426,57	0,32	95,2

[4] https://saiclearning.wordpress.com/2014/04/08/how-to-calculate-peak-theoretical-performance-of-a-cpu-based-hpc-system/

Exercise 06

The results presented in table 2 are better displayed in graph 2.



Graph 2: Performance achieved for the best configuration for different executables.

As can be noticed from table 2 and graph 2, the other compilers' output are closer to the theoretical peak performance with much smaller standard deviation associated, Intel64 presenting the best result with 95,4% of the theoretical peak performance.

Exercise 06

APPENDIX A - HPL input:

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6           device out (6=stdout,7=stderr,file)
1           # of problems sizes (N)
64512        Ns
1           # of NBs
256          NBs
0           PMAP process mapping (0=Row-,1=Column-major)
1           # of process grids (P x Q)
4           Ps
5           Qs
16.0         threshold
1           # of panel fact
2           PFACTs (0=left, 1=Crout, 2=Right)
1           # of recursive stopping criterium
4           NBMINs (>= 1)
1           # of panels in recursion
2           NDIVs
1           # of recursive panel fact.
1           RFACTs (0=left, 1=Crout, 2=Right)
1           # of broadcast
1           BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1           # of lookahead depth
1           DEPTHs (>=0)
2           SWAP (0=bin-exch,1=long,2=mix)
64          swapping threshold
0           L1 in (0=transposed,1=no-transposed) form
0           U  in (0=transposed,1=no-transposed) form
1           Equilibration (0=no,1=yes)
8           memory alignment in double (> 0)
##### This line (no. 32) is ignored (it serves as a separator). ######
0                         Number of additional problem sizes for PTRANS
1200 10000 30000             values of N
0                         number of additional blocking sizes for PTRANS
40 9 8 13 13 20 16 32 64       values of NB
```

Exercise 06

APPENDIX B - HPL output:

```
================================================================================
HPLinpack 2.2  --  High-Performance Linpack benchmark  --   February 24, 2016
Written by A. Petitet and R. Clint Whaley,  Innovative Computing Laboratory, UTK
Modified by Piotr Luszczek, Innovative Computing Laboratory, UTK
Modified by Julien Langou, University of Colorado Denver
================================================================================

An explanation of the input/output parameters follows:
T/V    : Wall time / encoded variant.
N      : The order of the coefficient matrix A.
NB     : The partitioning blocking factor.
P      : The number of process rows.
Q      : The number of process columns.
Time   : Time in seconds to solve the linear system.
Gflops : Rate of execution for solving the linear system.

The following parameter values will be used:

N      :   64512
NB     :     256
PMAP   : Row-major process mapping
P      :       4
Q      :       5
PFACT  :   Right
NBMIN  :       4
NDIV   :       2
RFACT  :   Crout
BCAST  :  1ringM
DEPTH  :       1
SWAP   : Mix (threshold = 64)
L1     : transposed form
U      : transposed form
EQUIL  : yes
ALIGN  : 8 double precision words

--------------------------------------------------------------------------------

- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
      ||Ax-b||_oo / ( eps * ( || x ||_oo * || A ||_oo + || b ||_oo ) * N )
- The relative machine precision (eps) is taken to be            2.220446e-16
- Computational tests pass if scaled residuals are less than                16.0

Column=000000256 Fraction= 0.4% Gflops=1.551e+04
...
Column=000064256 Fraction=99.6% Gflops=4.207e+02
================================================================================
T/V              N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR11C2R4      64512   256     4     5             425.75              4.204e+02
HPL_pdgesv() start time Mon Nov 26 15:04:56 2018

HPL_pdgesv() end time   Mon Nov 26 15:12:01 2018

--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV-
Max aggregated wall time rfact . . . :               2.38
+ Max aggregated wall time pfact . . :               0.70
+ Max aggregated wall time mxswp . . :               0.45
Max aggregated wall time update  . . :             423.07
+ Max aggregated wall time laswp . . :              29.28
Max aggregated wall time up tr sv  . :               0.24
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0015076 ...... PASSED
================================================================================
Finished      1 tests with the following results:
              1 tests completed and passed residual checks,
              0 tests completed and failed residual checks,
              0 tests skipped because of illegal input values.
--------------------------------------------------------------------------------
End of Tests.
================================================================================
cn08-16 : flushing filesystem cache...
cn08-16 : filesystem cache flushed
```