|  |  |
| --- | --- |
|  |  |

The Ministry of Education and Science of the Russian Federation

Lobachevsky State University of Nizhni Novgorod

Computing Mathematics and Cybernetics faculty

The competitiveness enhancement program   
of the Lobachevsky State University of Nizhni Novgorod   
among the world's research and education centers

Strategic initiative  
“Achieving leading positions in the field of supercomputer technology   
and high-performance computing”

**PROGRAMMING AND OPTIMIZATION   
FOR INTEL XEON PHI**

*Test questions*

Nizhni Novgorod

2014

# 01\_LECTURE. INTEL XEON PHI ARCHITECTURE

S,N,1,2.0.0.

How many cores does Intel Xeon Phi have?

-3,N

60

9,Y

61

-3,N

240

-3,N

244

---

S,N,1,2.0.0.

How many cores is it recommended to use for computations on Intel Xeon Phi?

9,Y

60

-3,N

61

-3,N

240

-3,N

244

---

S,N,1,2.0.0.

How many threads can be concurrently executed on Intel Xeon Phi coprocesssor?

-3,N

60

-3,N

61

-3,N

240

9,Y

244

---

S,N,1,2.0.0.

How many pipelines does one core of Intel Xeon Phi contain?

-3,N

1

9,Y

2

-3,N

4

-3,N

8

---

S,N,1,2.0.0.

How many threads can be concurrently executed on one core of Intel Xeon Phi?

-3,N

1

-3,N

2

9,Y

4

-3,N

8

---

S,N,1,2.0.0.

How many memory controllers does Intel Xeon Phi contain?

-3,N

1

-3,N

2

-3,N

4

9,Y

8

---

S,N,1,2.0.0.

How many memory access channels does each memory controller of Intel Xeon Phi contain?

-3,N

1

9,Y

2

-3,N

3

-3,N

4

---

S,N,1,2.0.0.

Does Xeon Phi core support out-of-order execution?

-10,N

Yes

10,Y

No

---

M,N,1,2.0.0.

Which of the following vector extensions does Intel Xeon Phi support?

-3,N

MultiMedia Extension (MMX)

-3,N

Streaming SIMD Extensions (SSE)

-3,N

Advanced Vector Extensions (AVX)

10,Y

None of the above.

---

S,N,1,2.0.0.

Does Xeon Phi core support hardware branch prediction and speculative execution?

10,Y

Yes

-10,N

No

---

S,N,1,2.0.0.

What is the size of vector registers of Intel Xeon Phi?

-3,N

128 bit

-3,N

256 bit

9,Y

512 bit

-3,N

1024 bit

---

S,N,1,2.0.0.

Vector unit is:

-10,N

not pipelined.

10,Y

pipelined.

---

S,N,1,2.0.0.

Which pipe receives the first decoded instruction of a pair?

10,Y

U-pipe

-5,N

V-pipe

-5,N

W-pipe

---

S,N,1,2.0.0.

What latency do the majority of integer arithmetic and mask instructions have?

9,Y

1 cycle

-3,N

2 cycles

-3,N

3 cycles

-3,N

4 cycles

---

S,N,1,2.0.0.

What latency do the majority of vector instructions have?

-3,N

1 cycle

-3,N

2 cycles

-3,N

3 cycles

9,Y

4 cycles

---

S,N,1,2.0.0.

What is the theoretical peak performance of Xeon Phi for single precision floating point numbers if 60 cores are used?

-3,N

528 GFLOPS

-3,N

1056 GFLOPS

9,Y

2112 GFLOPS

-3,N

4224 GFLOPS

---

S,N,1,2.0.0.

What is the theoretical peak performance of Xeon Phi for double precision floating point numbers if 60 cores are used?

-3,N

528 GFLOPS

9,Y

1056 GFLOPS

-3,N

2112 GFLOPS

-3,N

4224 GFLOPS

---

S,N,1,2.0.0.

How many stages does Intel Xeon Phi core pipeline have?

-3,N

6

9,Y

7

-3,N

8

-3,N

9

---

M,N,1,2.0.0.

Choose the correct statements.

5,Y

Core pipeline cannot fetch instructions of the same thread on two sequential cycles.

-5,N

On each cycle a core can execute 4 instructions simultaneously.

-5,N

To achieve full workload of a core it is required to execute 4 threads simultaneously.

5,Y

To achieve full workload of a core it is required to execute at least 2 threads simultaneously.

---

S,N,1,2.0.0.

Choose the correct statements.

10,Y

Each core of Intel Xeon Phi contains its own L1 and L2 caches.

-5,N

Each core of Intel Xeon Phi contains its own L1 cache and shared L2 cache.

-5,N

Each core of Intel Xeon Phi contains its own L1 and L2 caches and shared L3 cache.

---

S,N,1,2.0.0.

Which cache architecture does Intel Xeon Phi employ?

-10,N

Exclusive.

10,Y

Inclusive.

---

S,N,1,2.0.0.

Which characteristics do L1 caches (instruction L1 I-Cache and data L1 D-Cache) of Intel Xeon Phi have?

-3,N

Size is 32 Kb, line size is 32 bytes, 4-way associative.

-3,N

Size is 32 Kb, line size is 64 bytes, 4-way associative.

9,Y

Size is 32 Kb, line size is 64 bytes, 8-way associative.

-3,N

Size is 64 Kb, line size is 64 bytes, 8-way associative.

---

S,N,1,2.0.0.

What is the L2 cache size of Intel Xeon Phi core?

-3,N

128 Kb

-3,N

256 Kb

9,Y

512 Kb

-3,N

1024 Kb

---

M,N,1,2.0.0.

Which schemes of interaction between cache and main memory are implemented at Intel Xeon Phi?

3,Y

Uncacheable (UC)

-2,Y

Write-through (WT), write-combining (WC), write-protect (WP)

3,Y

Write-back (WB)

-2,Y

Write-combining (WC)

-2,Y

Write-protect (WP)

---

M,N,1,2.0.0.

Which modes and address sizes does Intel Xeon Phi support?

3,Y

32-bit mode, 32-bit address

3,Y

32-bit mode, 36-bit address

3,Y

64-bit mode, 40-bit address

-9,N

64-bit mode, 64-bit address

---

M,N,1,2.0.0.

Which page sizes does Intel Xeon Phi support?

-3,N

2 KB

5,Y

4 KB

5,Y

2 MB

-3,N

4 MB

-4,N

1 ГБ

---

M,N,1,2.0.0.

Choose the correct statements.

5,Y

Intel Xeon Phi uses separate page descriptor caches TLB L1 for data and instructions.

-5,N

Intel Xeon Phi uses shared page descriptor cache TLB L1 for data and instructions.

-5,N

Intel Xeon Phi uses separate page descriptor caches TLB L2 for data and instructions.

5,Y

Intel Xeon Phi uses shared page descriptor cache TLB L2 for data and instructions.

---

S,N,1,2.0.0.

What is the overall throughput of all memory controllers of Intel Xeon Phi?

-3,N

59,7 GB/s

-3,N

244 GB/s

9,Y

352 GB/s

-3,N

392 GB/s

---

S,N,1,2.0.0.

What is memory latency of Intel Xeon Phi?

-3,N

About 100 cycles.

-3,N

About 200 cycles.

9,Y

About 300 cycles.

-3,N

About 400 cycles.

---

S,N,1,2.0.0.

Which type of data transfers can cores of Intel Xeon Phi perform?

-3,N

From GDDR5 coprocessor memory to host memory

-3,N

From host memory to GDDR5 coprocessor memory

-3,N

From GDDR5 coprocessor memory to GDDR5 memory of the same coprocessor

9,Y

All of the above

---

S,N,1,2.0.0.

Choose the correct statement.

10,Y

Each core contains only local L1 cache directory.

-10,N

Each core contains a part of distributed L1 cache.

---

S,N,1,2.0.0.

Choose the correct statement.

-10,N

Each core contains only local L2 cache directory.

10,Y

Each core contains a part of distributed L2 cache.

---

M,N,1,2.0.0.

Which additional features are implemented at Intel Xeon Phi compared to Xeon CPUs?

2,Y

Larger vector register size.

-6,N

Extended AVX instruction set.

2,Y

Special mask register for vectorized conditional operations.

2,Y

Instructions for vector packing of non-unit stride data.

---

# 02\_Lecture. Program execution on INTEL XEON PHI. Computational Models on Intel Xeon Phi

S,N,1,2.0.0.

Which basic operating systems allow to use Intel Xeon Phi coprocessors?

-5,Y

Linux OS.

-5,Y

Windows OS.

10,Y

Linux and Windows OS.

---

S,N,1,2.0.0.

OS running on Intel Xeon Phi coprocessor is loaded:

-5,Y

from coprocessor ROM.

-5,Y

from coprocessor flash memory.

10,Y

by a request from host OS.

---

S,N,1,2.0.0.

What is Symmetric Communication Interface API (SCIF API)?

-5,N

A standard API for high performance computing.

-5,N

A high-level API that provides data transfers between host and coprocessor.

10,Y

An API that provides interconnection between host system and coprocessor.

---

S,N,1,2.0.0.

Which OS is running on Intel Xeon Phi by default?

10,Y

Specialized version of Linux based on the standard kernel.

-5,N

Unix OS with a specialized kernel.

-5,N

Windows OS with a specialized kernel.

---

M,N,1,2.0.0.

Which ways can be used for data transfer between host and coprocessor memory?

5,Y

Copying via PCI Express bus.

5,Y

DMA transfers.

-10,N

Address mapping of coprosessor memory to host memory.

---

M,N,1,2.0.0.

In which cases DMA transfers are preferable over copying (choose all appropriate cases)?

5,Y

One aims to reduce host processor load

5,Y

One operates with huge chunks of data.

-10,N

One needs parallel handling of multiple requests.

---

S,N,1,2.0.0.

In which ways could computational resources of a system with Xeon Phi coprocessors be utilized? Choose all correct ways.

5,Y

Use only host processors.

5,Y

Use only coprocessors.

-10,N

Processors and coprocessors must be used simultaneously.

---

M,N,1,2.0.0.

Which execution modes are supported? Choose all correct modes.

5,Y

Offload

5,Y

MPI

-10,N

Dynamic

---

S,N,1,2.0.0.

In Symmetric model

10,Y

MPI processes are executed on both host processors and Xeon Phi coprocessors.

-5,N

MPI processes are executed only on host processors.

-5,N

MPI processes are executed only on Xeon Phi coprocessors.

---

S,N,1,2.0.0.

In Offload mode:

-5,N

MPI processes are executed on both host processors and Xeon Phi coprocessors.

10,Y

MPI processes are executed only on host processors.

-5,N

MPI processes are executed only on Xeon Phi coprocessors.

---

S,N,1,2.0.0.

In Coprocessor-only model

-5,N

MPI processes are executed on both host processors and Xeon Phi coprocessors.

-5,N

MPI processes are executed only on host processors.

10,Y

MPI processes are executed only on Xeon Phi coprocessors.

---

M,N,1,2.0.0.

Which programming technologies, libraries and API can be used in programming for Intel Xeon Phi?

2,Y

OpenMP

2,Y

Intel TBB

-10,N

Intel ArBB

2,Y

Intel Cilk Plus

2,Y

pthreads

2,Y

MPI

---

S,N,1,2.0.0.

In Symmetric model for interactions between host processors, inside coprocessors and between host processors and coprocessors is based by default on

-10,N

tcp protocol.

10,Y

Shared memory mechanism.

---

S,N,1,2.0.0.

What happened if Xeon Phi coprocessor is busy when offload-code is invoked?

-5,N

Application is terminated.

-5,N

Application is waiting for a vacant Xeon Phi coprocessor.

10,Y

Offload-code is executed on host.

---

S,N,1,2.0.0.

Double buffering technique allows to

-5,N

Reduce data transfer time in Offload mode.

-5,N

Reduce memory consumption in Offload mode.

10,Y

Reduce or fully hide data transfer latency in Offload mode.

---

S,N,1,2.0.0.

Is it possible to transfer complex data structures (e.g. using pointers) between host and coprocessor in Offload mode?

-10,N

No.

10,Y

Yes, there is a special mechanism for this purpose.

---

Is it allowed to call MPI routines from offload-code in Offload mode?

10,Y

No.

-10,N

Yes, there is a special mechanism for this purpose.

---

# 04\_lecture. Vector extensions Of Intel Xeon Phi

S,N,1,2.0.0.

Which icc compiler flag is used for offload mode:

-3,N

-mmic

-3,N

-mic

9,Y

No flag is required

-3,N

-offload

---

S,N,1,2.0.0.

Which icc compiler flag is used for coprocessor-only mode:

9,Y

-mmic

-3,N

-mic

-3,N

No flag is required

-3,N

-offload

---

S,N,1,2.0.0.

#!/bin/sh

mpicc –O2 –openmp main.cpp –o ./program\_name

mpicc –O2 –openmp –mmic main.cpp –o ./program\_name.mic

Which mode is this script used for?

-5,N

offload

-5,N

coprocessor-only

10,Y

symmetric

---

S,N,1,2.0.0.

Which of the following scripts is used for building a program for offload mode?

10,Y

#!/bin/sh

mpicc -02 -openmp main.cpp –o program\_name

-5,N

#!/bin/sh

mpicc -02 -openmp -mmic main.cpp –o program\_name.mic

-5,N

#!/bin/sh

mpicc –O2 –openmp main.cpp –o ./program\_name

mpicc –O2 –openmp –mmic main.cpp –o ./program\_name.mic

---

S,N,1,2.0.0.

Which of the following scripts is used for building a program for coprocessor-only mode?

-5,N

#!/bin/sh

mpicc -02 -openmp main.cpp –o program\_name

10,Y

#!/bin/sh

mpicc -02 -openmp -mmic main.cpp –o program\_name.mic

-5,N

#!/bin/sh

mpicc –O2 –openmp main.cpp –o ./program\_name

mpicc –O2 –openmp –mmic main.cpp –o ./program\_name.mic

---

S,N,1,2.0.0.

Which of the following scripts is used for building a program for symmetric mode?

-5,N

#!/bin/sh

mpicc -02 -openmp main.cpp –o program\_name

-5,N

#!/bin/sh

mpicc -02 -openmp -mmic main.cpp –o program\_name.mic

10,Y

#!/bin/sh

mpicc –O2 –openmp main.cpp –o ./program\_name

mpicc –O2 –openmp –mmic main.cpp –o ./program\_name.mic

---

S,N,1,2.0.0.

Which of the following scripts is used for launching a program in offload mode?

10,Y

#!/bin/sh

mpiexec.hydra –perhost 1 ./program\_name

-5,N

#!/bin/sh

mpiexec.hydra –host mic0 –n 1 –perhost 1 ./program\_name.mic

-5,N

#!/bin/sh

mpiexec.hydra –hosts 2 node0 node1 –n 2 –perhost 1 ./program\_name: \

–hosts 4 mic0 mic1 mic2 mic3 –n 4 –perhost 1 ./program\_name.mic

---

S,N,1,2.0.0.

Which of the following scripts is used for launching a program in coprocessor-only mode?

-5,N

#!/bin/sh

mpiexec.hydra –perhost 1 ./program\_name

10,Y

#!/bin/sh

mpiexec.hydra –host mic0 –n 1 –perhost 1 ./program\_name.mic

-5,N

#!/bin/sh

mpiexec.hydra –hosts 2 node0 node1 –n 2 –perhost 1 ./program\_name: \

–hosts 4 mic0 mic1 mic2 mic3 –n 4 –perhost 1 ./program\_name.mic

---

S,N,1,2.0.0.

Which of the following scripts is used for launching a program in symmetric mode?

-5,N

#!/bin/sh

mpiexec.hydra –perhost 1 ./program\_name

-5,N

#!/bin/sh

mpiexec.hydra –host mic0 –n 1 –perhost 1 ./program\_name.mic

10,Y

#!/bin/sh

mpiexec.hydra –hosts 2 node0 node1 –n 2 –perhost 1 ./program\_name: \

–hosts 4 mic0 mic1 mic2 mic3 –n 4 –perhost 1 ./program\_name.mic

---

S,N,1,2.0.0.

#!/bin/sh

mpiexec.hydra –perhost 1 ./program\_name

Which mode is this script supposed for?

10,Y

offload

-5,N

coprocessor only

-5,N

symmetric

---

S,N,1,2.0.0.

#!/bin/sh

mpiexec.hydra –host mic0 –n 1 –perhost 1 ./program\_name.mic

Which mode is this script supposed for?

-5,N

offload

10,Y

coprocessor only

-5,N

symmetric

---

S,N,1,2.0.0.

#!/bin/sh

mpiexec.hydra –hosts 2 node0 node1 –n 2 –perhost 1 ./program\_name: \

–hosts 4 mic0 mic1 mic2 mic3 –n 4 –perhost 1 ./program\_name.mic

Which mode is this script supposed for?

-5,N

offload

-5,N

coprocessor only

10,Y

symmetric

---

S,N,1,2.0.0.

#!/bin/sh

export MICperNODE=1

sbatch –N 4 –-gres=mic:2 native\_run.sh ./program\_name

This script is for SLURM, coprocessor-only mode. How many processes per node will be created?

9,Y

1

-3,N

2

-3,N

4

-3,N

8

---

S,N,1,2.0.0.

#!/bin/sh

export MICperNODE=2

sbatch –N 2 –-gres=mic:4 native\_run.sh ./program\_name

This script is for SLURM, coprocessor-only mode. How many processes will be created in total?

-3,N

1

-3,N

2

9,Y

4

-3,N

8

---

S,N,1,2.0.0.

#!/bin/sh

export PPN=2

export MICperNODE=2

sbatch –N 2 –-gres=mic:2 symmetric\_run.sh ./program\_name

This script is for SLURM, symmetric mode. How many processes will be created in total?

-3,N

1

-3,N

2

-3,N

4

9,Y

8

---

S,N,1,2.0.0.

Which of the following scripts for SLURM provides exclusive access to two cluster nodes each with at least 2 coprocessors?

-3,N

#!/bin/sh

sbatch –N 2 –-gres=mic:1

-3,N

#!/bin/sh

salloc –N 1 --gres=mic:2

9,Y

#!/bin/sh

salloc –N 2 --gres=mic:2

-3,N

#!/bin/sh

sbatch –N 2 –-gres=mic:2

---

S,N,1,2.0.0.

A program has the following piece of code:

#pragma offload\_attribute(push, target(mic))

void f1(){…}

#pragma offload\_attribute(pop)

Is –mmic compiler flag required for code execution on host?

10,Y

No

-10,N

Yes

---

S,N,1,2.0.0.

Is it possible to execute program built with –mmic on host?

10,Y

No

-10,N

Yes

---

S,N,1,2.0.0.

Which compiler flag is used to build for native mode?

10,Y

-mmic

-5,N

-mphi

-5,N

-mic

---

S,N,1,2.0.0.

A program has the following piece of code:

#pragma offload target(mic:0) signal(s1)

{

F1(p1, p2);

}

F2();

In which order will F1 and F2 be executed?

-5,N

First F1, then F2

-5,N

First F2, then F1

10,Y

Asynchronously F1 and F2

---

S,N,1,2.0.0.

A program has the following piece of code:

#pragma offload target(mic:0)

{

F1(p1, p2);

}

F2();

In which order will F1 and F2 be executed?

10,Y

First F1, then F2

-5,N

First F2, then F1

-5,N

Asynchronously F1 and F2

---

S,N,1,2.0.0.

A program has the following piece of code:

#pragma offload target(mic:0) wait (s1)

{

F1(p1, p2);

}

F2();

In which order will F1 and F2 be executed?

10,Y

First F1, then F2

-5,N

First F2, then F1

-5,N

Asynchronously F1 and F2

---

S,N,1,2.0.0.

A program has the following piece of code:

F1();

#pragma offload target(mic:0) signal(s1)

{

F2(p1, p2);

}

In which order will F1 and F2 be executed?

10,Y

First F1, then F2

-5,N

First F2, then F1

-5,N

Asynchronously F1 and F2

---

S,N,1,2.0.0.

A program has the following piece of code:

F1();

#pragma offload target(mic:0) wait(s1)

{

F2(p1, p2);

}

In which order will F1 and F2 be executed?

10,Y

First F1, then F2

-5,N

First F2, then F1

-5,N

Asynchronously F1 and F2

---

S,N,1,2.0.0.

A program has the following piece of code:

int r = 0;

int main()

{

#pragma offload target(mic:0)

{

setR();

}

printf("%d", r);

}

...

setR()

{

r = 1;

}

What will be printed on the screen?

10,Y

0

-5,N

1

-5,N

Result is not defined

---

# Questions

S,N,1,2.0.0.

The approach to organize computations using SIMD instructions (SSE, SSE2, etc.) is called:

-3,N

Interpolation

-3,N

Modulation

10,Y

Vectorization

-3,N

Combination

---

S,N,1,2.0.0.

The essence of SIMD paradigm is:

10,Y

Performing an operation with several sets of data in parallel

-3,N

Performing several operations on the same data in parallel

-3,N

Performing several different operations on several different sets of data in parallel

-3,N

Performing an operation on the same data several times

---

S,N,1,2.0.0.

Vector register size for floating-point data in SSE is:

-3,N

64 bytes

-3,N

32 bytes

10,Y

16 bytes

-3,N

8 bytes

---

S,N,1,2.0.0.

Vector register size for floating-point data in AVX is:

-3,N

64 bytes

10,Y

32 bytes

-3,N

16 bytes

-3,N

8 bytes

---

S,N,1,2.0.0.

Vector register size for floating-point data in Xeon Phi vector extensions:

10,Y

64 bytes

-3,N

32 bytes

-3,N

16 bytes

-3,N

8 bytes

---

S,N,1,2.0.0.

Which of the following statements is correct?

10,Y

A fully scalar code cannot achieve good performance (as percentage of peak performance) on Xeon Phi.

-10,N

Vectorized code cannot achieve good performance (as percentage of peak performance) on Xeon Phi.

---

S,N,1,2.0.0.

FMA instruction allows to perform the following operation:

-3,N

a = a + b

10,Y

a = a + b \* c

-3,N

a = a + b / c;

-3,N

a = a \* b \* c;

---

S,N,1,2.0.0.

Which of the following statements is correct?

-3,N

FMA instruction allows to perform 3 arithmetic operations per clock.

-3,N

FMA instruction allows to perform 2 arithmetic operations without any loss of precision.

10,Y

FMA allows to perform 2 arithmetic operations with a single round-off in the end.

-3,N

FMA allows to perform 3 arithmetic operations with a single round-off in the end.

---

S,N,1,2.0.0.

How is extended support for mathematical functions implemented in Xeon Phi?

-3,N

The main mathematical functions are implemented in hardware.

-3,N

The main mathematical functions are computed without any loss of precision.

10,Y

Four of mathematical functions are implemented in hardware for single precision.

-3,N

All mathematical functions are implemented in hardware for single precision.

---

S,N,1,2.0.0.

Vectorization is a parallelism on level of:

-3,N

Processes.

-3,N

Threads.

-3,N

Cores.

10,Y

Operations inside a core.

---

S,N,1,2.0.0.

Which of the following compiler directives are intended for vectorization?

-5,N

#pragma omp parallel for

5,Y

#pragma ivdep

5,Y

#pragma simd

-5,N

#pragma cilk\_for

---

S,N,1,2.0.0.

Which of the following ways can be used for vectorization?

-3,N

Only compiler vectorization of loops.

-3,N

Only manual usage of intrinsic functions.

10,Y

Compiler and manual vectorization.

-3,N

Only programming using special vectorized languages.

---

S,N,1,2.0.0.

Which of the following keywords in C/C++ is used to notify a compiler that a certain region of memory is pointed by the only pointer (there is no pointer collision)?

-3,N

register

9,Y

restrict

-3,N

auto

-3,N

extern

---

S,N,1,2.0.0.

Which compiler directive is used to notify that there are no dependencies in a loop?

9,Y

#pragma ivdep

-3,N

#pragma simd

-3,N

#pragma omp

-3,N

#pragma vector aligned

---

S,N,1,2.0.0.

Which compiler directive is used to force unconditional loop vectorization?

-3,N

#pragma ivdep

9,Y

#pragma simd

-3,N

#pragma omp

-3,N

#pragma vector aligned

---

S,N,1,2.0.0.

Which compiler directive is used to notify a compiler that all arrays used in a loop are aligned?

-3,N

#pragma ivdep

-3,N

#pragma simd

-3,N

#pragma omp

9,Y

#pragma vector aligned

---

S,N,1,2.0.0.

Array Notation in Intel Cilk Plus is used for

-5,N

Code parallelization

10,Y

Code vectorization

-5,N

Code simplification

---

S,N,1,2.0.0.

Elemental Functions in Intel Cilk Plus are used for

10,Y

Code vectorization

-5,N

Code parallelization

-5,N

Code simplification

---

# 05\_Practice. Optimization of APPLICATIONS FOR Intel Xeon Phi USING Intel C/C++ Compiler. vECTORIZATION

S,N,1,2.0.0.

Are iterations of the following loop dependent? If so, select proper characterizations of the dependence.

for (int i = 0; i < n – 1; i++)

a[i + 1] = a[i] \* 2 – 5;

-5,N

Iterations are independent.

5,Y

Each iteration depends on the previous one: a[i + 1] cannot be computed before a[i].

-5,N

Each iteration depends on the next one: a[i] cannot be computed before a[i + 1].

5,Y

Each iteration depends on the previous one: a[i] cannot be computed before a[i - 1].

---

S,N,1,2.0.0.

Are iterations of the following loop dependent? If so, select proper characterizations of the dependence.

for (int i = 0; i < n – 2; i++)

a[i + 2] = a[i] \* 3 + 1;

-3,N

Iterations are independent.

-3,N

Each iteration depends on the previous one: a[i + 1] cannot be computed before a[i].

10,Y

All even iterations are dependent between each other, all odd iterations are dependent between each other, but there is no dependency between even and odd iterations.

-3,N

Each iteration depends on the previous one: a[i + 2] cannot be computed before a[i].

---

S,N,1,2.0.0.

Are iterations of the following loop dependent? If so, select proper characterizations of the dependence. Arrays a, b are guaranteed to not overlap.

for (int i = 0; i < n; i++)

a[i] = b[i] \* 2 + 8;

10,Y.

Iterations are independent.

-3,N

Each iteration depends on the previous one: a[i + 1] cannot be computed before a[i].

-3,N

Each iteration depends on the previous one: a[i] cannot be computed before b[i].

-3,N

Each iteration depends on the previous one: a[i + 1] cannot be computed before b[i].

---

S,N,1,2.0.0.

Are iterations of the following loop dependent? If so, select proper characterizations of the dependence. Arrays a, b are guaranteed to not overlap.

for (int i = 0; i < n - 2; i++)

a[i] = b[i + 2] \* 4 - 5;

10,Y.

Iterations are independent.

-3,N

Each iteration depends on the previous one: a[i + 1] cannot be computed before a[i].

-3,N

Each iteration depends on the previous one: a[i] cannot be computed before b[i + 2].

-3,N

All even iterations are dependent between each other, all odd iterations are dependent between each other, but there is no dependency between even and odd iterations.

---

S,N,1,2.0.0.

Are iterations of the following loop dependent? If so, select proper characterizations of the dependence. Arrays a, b, c are guaranteed to not overlap.

for (int i = 1; i < n - 2; i++)

a[i] = b[i + 2] \* c[i – 1];

10,Y.

Iterations are independent.

-3,N

Each iteration depends on the previous one: a[i + 1] cannot be computed before a[i].

-3,N

Each iteration depends on the previous one: a[i + 1] cannot be computed before b[i + 2].

-3,N

Each iteration depends on the previous one: a[i + 1] cannot be computed before c[i - 1].

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

for (int j = 2; j < 1000; j++)

{

if (number == 1) break;

int r;

r = number % j;

if (r == 0)

{

number /= j;

divisors[idx].push\_back(j);

j--;

}

}

10,Y

No, there is dependence

-5,N

Yes

-5,N

No, number of iterations is not multiple of 8

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 128

…

int rr[LOOP\_SIZE];

for(int k = 0; k < LOOP\_SIZE; k++)

{

rr[k] = number % k;

}

-5,N

No, there is dependence

10,Y

Yes

-5,N

No, number of iterations is too small

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 130

…

int rr[LOOP\_SIZE];

for(int k = 0; k < LOOP\_SIZE; k++)

{

rr[k] = number % k;

}

-3,N

No, there is dependence

10,Y

Yes, the loop will be split into two and the main part will be vectorized

-3,N

No, number of iterations is too small

-3,N

No, number of iterations is not multiple of 8

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 16

…

int rr[LOOP\_SIZE];

for(int k = 0; k < LOOP\_SIZE; k++)

{

rr[k] = number % k;

}

-5,N

No, there is dependence

10,Y

Yes

-5,N

No, number of iterations is too small

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 16

…

int rr[LOOP\_SIZE];

…

p = 1;

for(int k = 0; k < LOOP\_SIZE; k++)

{

p \*= rr[k];

}

-5,N

No, there is dependence

10,Y

Yes

-5,N

No, number of iterations is too small

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 30

…

int rr[LOOP\_SIZE];

…

p = 1;

for(int k = 0; k < LOOP\_SIZE; k++)

{

p \*= rr[k];

}

-5,N

No, there is dependence

10,Y

Yes, the loop will be split into two and the main part will be vectorized

-5,N

No, number of iterations is too small

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 128

…

int rr[LOOP\_SIZE];

…

p = 1;

for(int k = 0; k < LOOP\_SIZE; k++)

{

p \*= rr[k];

}

-5,N

No, there is dependence

10,Y

Yes

-5,N

No, number of iterations is too small

---

S,N,1,2.0.0.

What speedup is expected due to vectorization of the following code on Xeon Phi?

#define LOOP\_SIZE 16

…

int rr[LOOP\_SIZE];

…

p = 1;

for(int k = 0; k < LOOP\_SIZE; k++)

{

p \*= rr[k];

}

10,Y

<16

-3,N

16

-3,N

>16

-3,N

8

---

S,N,1,2.0.0.

What speedup is expected due to vectorization of the following code on Xeon Phi?

#define LOOP\_SIZE 16

…

int rr[LOOP\_SIZE];

for(int k = 0; k < LOOP\_SIZE; k++)

{

rr[k] = number % k;

}

10,Y

<16

-3,N

16

-3,N

>16

-3,N

8

---

S,N,1,2.0.0.

What speedup is expected due to vectorization of the following code on Xeon Phi?

#define LOOP\_SIZE 24

…

int rr[LOOP\_SIZE];

…

p = 1;

for(int k = 0; k < LOOP\_SIZE; k++)

{

p \*= rr[k];

}

10,Y

<12

-3,N

16

-3,N

>12

-3,N

8

---

S,N,1,2.0.0.

What speedup is expected due to vectorization of the following code on Xeon Phi?

#define LOOP\_SIZE 24

…

int rr[LOOP\_SIZE];

for(int k = 0; k < LOOP\_SIZE; k++)

{

rr[k] = number % k;

}

10,Y

<12

-3,N

16

-3,N

>12

-3,N

8

---

# 06\_Lecture. Optimization of applications for Intel Xeon Phi. Intel C/C++ Compiler

M,N,1,2.0.0.

Which of the following language constructions provide data exchange between host and coprocessor memory?

-3,N

#pragma offload\_attribute

3,Y

#pragma offload\_transfer

3,Y

#pragma offload

-3,N

\_\_attribute\_\_((target(mic)))

---

S,N,1,2.0.0.

A system has 3 coprocessors. Directive #pragma offload target(mic:4) results in:

-3,N

Execution of the following code block on coprocessor 0.

9,Y

Execution of the following code block on coprocessor 1.

-3,N

Execution of the following code block on coprocessor 2.

-3,N

Runtime error.

---

S,N,1,2.0.0.

Which parameter of #pragma offload directive controls data copy from host to coprocessor?

-3,N

out

10,Y

in

-3,N

target

-3,N

signal

---

S,N,1,2.0.0.

Which parameter of #pragma offload directive controls data copy from coprocessor to host?

10,Y

out

-3,N

in

-3,N

target

-3,N

signal

---

M,N,1,2.0.0.

Which data types can be copied using #pragma offload with parameter “in”?

5,Y

Scalar variables of all standard types

5,Y

Structures with no pointers (POD)

-10,N

Static arrays

---

S,N,1,2.0.0.

Which parameters should be placed instead of [PARAMETERS] in the following piece of code for correctness and efficiency?

\_\_attribute\_\_((target(mic))) void func(float\* a,

float\* b, int count, float c, float d)

{

#pragma omp parallel for

for (int i = 0; i < count; ++i)

{

a[i] = b[i]\*c + d;

}

}

int main()

{

const int count = 100;

float a[count], b[count], c, d;

…

#pragma offload target(mic) [PARAMETERS]

func(a, b, count, c, d);

…

}

-3,N

in(a) inout(b)

9,Y

in(a) out(b)

-3,N

nocopy(a, b)

-3,N

inout(a, b)

---

M,N,1,2.0.0.

Which of the following Intel Cilk Plus keywords allow to execute a function on a coprocessor?

3,Y

\_Cilk\_offload

3,Y

\_Cilk\_offload\_to

-3,N

\_Cilk\_spawn

-3,N

\_Cilk\_shared

---

S,N,1,2.0.0.

Which of the following Intel Cilk Plus keywords is used for non-blocking execution on a coprocessor (CPU is not waiting for the coprocessor)?

-3,N

\_Cilk\_offload

-3,N

\_Cilk\_offload\_to

9,Y

\_Cilk\_spawn

-3,N

\_Cilk\_shared

---

S,N,1,2.0.0.

Which of the following Intel Cilk Plus keywords is used to declare a variable available from both CPU and coprocessor?

-3,N

\_Cilk\_offload

-3,N

\_Cilk\_offload\_to

-3,N

\_Cilk\_spawn

9,Y

\_Cilk\_shared

---

M,N,1,2.0.0.

Which of the following programming languages support explicit memory copying in offload mode?

3,Y

C

3,Y

C++

3,Y

Fortran

-9,N

C#

---

M,N,1,2.0.0.

Which of the following programming languages support implicit memory copying in offload mode?

3,Y

C

3,Y

C++

-3,N

Fortran

-3,N

C#

---

M,N,1,2.0.0.

Array A is of size 100. How to access all elements using Array Notation?

3,Y

A[:]

-9,N

A[100]

3,Y

A[0:100]

3,Y

A[0:100:1]

---

M,N,1,2.0.0.

Array A is of size 100. How to access elements 10, 11, 12, 13, 14 using Array notation?

-3,N

A[10:14]

-3,N

A[10:13]

9,Y

A[10:5]

-3,N

A[10:4]

---

M,N,1,2.0.0.

Array A is of size 100. How to access elements 2, 4, 6, 8, 10 using Array notation?

-3,N

A[2:10]

-3,N

A[2:2:10]

9,Y

A[2:5:2]

-3,N

A[2:2:5]

---

M,N,1,2.0.0.

Which of the following expressions are valid in Array Notation?

-3,N

a[0:5] = b[0:6];

3,Y

a[0:4] = 5;

3,Y

a[0:4] = b[i];

-3,N

b[i] = a[0:4];

---

M,N,1,2.0.0.

Choose the correct statements concerning Elemental Functions:

3,Y

Implicit calls are forbidden

-9,N

passing parameters by reference is forbidden

3,Y

passing structures by value is forbidden

3,Y

synchronization is forbidden

---

S,N,1,2.0.0.

Which compiler option gives the most detailed vectorization report?

-3,N

–vec-report3

9,Y

–vec-report6

-3,N

–vec-report5

-3,N

–vec-report1

---

S,N,1,2.0.0.

When using -vec-report3, messages of inability to vectorize a loop look like:

-5,N

loop was not vectorized

10,Y

loop was not vectorized: unsupported data type

-5,N

vectorization support: type TTT is not supported for operation OOO

---

S,N,1,2.0.0.

When using -vec-report6, messages of inability to vectorize a loop look like:

-5,N

loop was not vectorized

-5,N

loop was not vectorized: unsupported data type

10,Y

vectorization support: type TTT is not supported for operation OOO

---

S,N,1,2.0.0.

Which of the following language constructs allow to align a static array?

12,Y

\_\_declspec(align(64)) or \_\_attribute\_\_((aligned(64)))

-3,N

\_mm\_malloc(bufsize, 64)

-3,N

\_\_assume\_aligned(A, 64)

-3,N

\_\_assume(n1%16==0)

-3,N

#pragma vector aligned

---

S,N,1,2.0.0.

Which of the following language constructs allow to align a dynamic array?

-3,N

\_\_declspec(align(64)) or \_\_attribute\_\_((aligned(64)))

12,Y

\_mm\_malloc(bufsize, 64)

-3,N

\_\_assume\_aligned(A, 64)

-3,N

\_\_assume(n1%16==0)

-3,N

#pragma vector aligned

---

S,N,1,2.0.0.

Which of the following language constructs notified the compiler that array is aligned by 64 bytes?

-3,N

\_\_declspec(align(64)) or \_\_attribute\_\_((aligned(64)))

-3,N

\_mm\_malloc(bufsize, 64)

12,Y

\_\_assume\_aligned(A, 64)

-3,N

\_\_assume(n1%16==0)

-3,N

#pragma vector aligned

---

S,N,1,2.0.0.

Which of the following language constructs notifies the compiler that a variable is a multiple of 16?

-3,N

\_\_declspec(align(64)) or \_\_attribute\_\_((aligned(64)))

-3,N

\_mm\_malloc(bufsize, 64)

-3,N

\_\_assume\_aligned(A, 64)

12,Y

\_\_assume(n1%16==0)

-3,N

#pragma vector aligned

---

S,N,1,2.0.0.

Which of the following language constructs notifies the compiler that all arrays used in a loop are aligned?

-3,N

\_\_declspec(align(64)) or \_\_attribute\_\_((aligned(64)))

-3,N

\_mm\_malloc(bufsize, 64)

-3,N

\_\_assume\_aligned(A, 64)

-3,N

\_\_assume(n1%16==0)

12,Y

#pragma vector aligned

---

S,N,1,2.0.0.

Compiler option -vec-report allows to retrieve:

9,Y

Vectorization report: which code was vectorized, which was not, reasons.

-3,N

Vectorization guide: how to modify the code for better vectorization.

-3,N

Optimization guide: how to modify the code for optimization.

-3,N

Optimization report: how the code was modified and optimized by the compiler.

.

---

S,N,1,2.0.0.

Опция компилятора Intel -guide-vec позволяет получить:

-3,N

Vectorization report: which code was vectorized, which was not, reasons.

9,Y

Vectorization guide: how to modify the code for better vectorization.

-3,N

Optimization guide: how to modify the code for optimization.

-3,N

Optimization report: how the code was modified and optimized by the compiler.

---

S,N,1,2.0.0.

Опция компилятора Intel -guide позволяет получить:

-3,N

Vectorization report: which code was vectorized, which was not, reasons.

-3,N

Vectorization guide: how to modify the code for better vectorization.

9,Y

Optimization guide: how to modify the code for optimization.

-3,N

Optimization report: how the code was modified and optimized by the compiler.

---

S,N,1,2.0.0.

Опция компилятора Intel -opt-report позволяет получить:

-3,N

Vectorization report: which code was vectorized, which was not, reasons.

-3,N

Vectorization guide: how to modify the code for better vectorization.

-3,N

Optimization guide: how to modify the code for optimization.

9,Y

Optimization report: how the code was modified and optimized by the compiler.

---

S,N,1,2.0.0.

Mapping of thread to cores that provides maximum number of threads per core (some cores might be free) is set by:

-3,N

KMP\_AFFINITY=”balanced”

-3,N

KMP\_AFFINITY=”scatter”

9,Y

KMP\_AFFINITY=”compact”

---

S,N,1,2.0.0.

What is the effect of poor load balancing?

10,Y

Inefficiency

-5,N

Computational errors

-5,N

Lack of memory

---

M,N,1,2.0.0.

What are the symptoms of poor load balancing?

-10,N

Incorrect results

5,Y

A large share of sequential code

5,Y

Small speedup compare to sequential code.

---

M,N,1,2.0.0.

Efficiency of a parallel application is significantly influenced by:

2,Y

Choice of synchronization primitives

2,Y

Frequency of synchronization between threads

2,Y

Fraction of sequential execution in total run time

-10,N

Source code size

2,Y

Load balancing

2,Y

Overheads of thread management

---

M,N,1,2.0.0.

What are the goals of load balancing?

-10,N

Provide correct results of a parallel program

5,Y

Increase performance of a parallel program

5,Y

Provide efficient utilization of hardware resources

---

# 07\_Practice. Prime factorization. Vectorization and load balancing

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

for (int j = 2; j < 1000; j++)

{

if (number == 1) break;

int r;

r = number % j;

if (r == 0)

{

number /= j;

divisors[idx].push\_back(j);

j--;

}

}

10,Y

No, because of data dependencies

-5,N

Yes

-5,N

No, because iteration number is odd

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 128

…

int rr[LOOP\_SIZE];

for(int k = 0; k < LOOP\_SIZE; k++)

{

rr[k] = number % k;

}

-5,N

No, because of data dependencies

10,Y

Yes

-5,N

No, because of small iteration number

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 130

…

int rr[LOOP\_SIZE];

for(int k = 0; k < LOOP\_SIZE; k++)

{

rr[k] = number % k;

}

-3,N

No, because of data dependencies

10,Y

The main part of the loop can be vectorized

-3,N

No, because of small iteration number

-3,N

No, because number of iteration is odd

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 16

…

int rr[LOOP\_SIZE];

for(int k = 0; k < LOOP\_SIZE; k++)

{

rr[k] = number % k;

}

-5,N

No, because of data dependencies

10,Y

Yes

-5,N

No, because of small iteration number

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 16

…

int rr[LOOP\_SIZE];

…

p = 1;

for(int k = 0; k < LOOP\_SIZE; k++)

{

p \*= rr[k];

}

-5,N

No, because of data dependencies

10,Y

Yes

-5,N

No, because of small iteration number

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 30

…

int rr[LOOP\_SIZE];

…

p = 1;

for(int k = 0; k < LOOP\_SIZE; k++)

{

p \*= rr[k];

}

-5,N

No, because of data dependencies

10,Y

The main part of the loop can be vectorized

-5,N

No, because of small iteration number

---

S,N,1,2.0.0.

Can the following loop be vectorized by the compiler on Xeon Phi?

#define LOOP\_SIZE 128

…

int rr[LOOP\_SIZE];

…

p = 1;

for(int k = 0; k < LOOP\_SIZE; k++)

{

p \*= rr[k];

}

-5,N

No, because of data dependencies

10,Y

Yes

-5,N

No, because of small iteration number

---

S,N,1,2.0.0.

What speedup from vectorization of the following loop is expected on Xeon Phi?

#define LOOP\_SIZE 16

…

int rr[LOOP\_SIZE];

…

p = 1;

for(int k = 0; k < LOOP\_SIZE; k++)

{

p \*= rr[k];

}

10,Y

<16

-3,N

16

-3,N

>16

-3,N

8

---

S,N,1,2.0.0.

What speedup from vectorization of the following loop is expected on Xeon Phi?

#define LOOP\_SIZE 16

…

int rr[LOOP\_SIZE];

for(int k = 0; k < LOOP\_SIZE; k++)

{

rr[k] = number % k;

}

10,Y

<16

-3,N

16

-3,N

>16

-3,N

8

---

S,N,1,2.0.0.

What speedup from vectorization of the following loop is expected on Xeon Phi?

#define LOOP\_SIZE 24

…

int rr[LOOP\_SIZE];

…

p = 1;

for(int k = 0; k < LOOP\_SIZE; k++)

{

p \*= rr[k];

}

10,Y

<12

-3,N

16

-3,N

>12

-3,N

8

---

S,N,1,2.0.0.

What speedup from vectorization of the following loop is expected on Xeon Phi?

#define LOOP\_SIZE 24

…

int rr[LOOP\_SIZE];

for(int k = 0; k < LOOP\_SIZE; k++)

{

rr[k] = number % k;

}

10,Y

<12

-3,N

16

-3,N

>12

-3,N

8

---

# 08\_Practice. Optimization of European Option Pricing

S,N,1,2.0.0.

Which keyword in C/C++ guarantees that a region of memory is accessed by a single pointer (there is no pointer overlapping)?

-3,N

register

9,Y

restrict

-3,N

auto

-3,N

extern

---

S,N,1,2.0.0.

Which compiler directive is used to guarantee there are no potential vector dependencies?

9,Y

#pragma ivdep

-3,N

#pragma simd

-3,N

#pragma omp

-3,N

#pragma vector aligned

---

S,N,1,2.0.0.

Which compiler directive is used for explicit loop vectorization?

-3,N

#pragma ivdep

9,Y

#pragma simd

-3,N

#pragma omp

-3,N

#pragma vector aligned

---

S,N,1,2.0.0.

Which compiler directive is used to guarantee that all arrays used in a loop are aligned?

-3,N

#pragma ivdep

-3,N

#pragma simd

-3,N

#pragma omp

9,Y

#pragma vector aligned

---

---

S,N,1,2.0.0.

Which compiler options must be used to compile the following function?

void GetOptionPrices(

float \* restrict pT, float \* restrict pK,

float \* restrict pS0, float \* restrict pC)

{

int i;

float d1, d2, erf1, erf2;

for (i = 0; i < N; i++)

{

d1 = (logf(pS0[i] / pK[i]) + (r + sig \* sig \* 0.5f) \*

pT[i]) / (sig \* sqrtf(pT[i]));

d2 = (logf(pS0[i] / pK[i]) + (r - sig \* sig \* 0.5f) \*

pT[i]) / (sig \* sqrtf(pT[i]));

erf1 = 0.5f + 0.5f \* erff(d1 / sqrtf(2.0f));

erf2 = 0.5f + 0.5f \* erff(d2 / sqrtf(2.0f));

pC[i] = pS0[i] \* erf1 - pK[i] \* expf((-1.0f) \* r \*

pT[i]) \* erf2;

}

}

-3,N

No special options

10,Y

-restrict

-3,N

-O2

-3,N

-debug

---

S,N,1,2.0.0.

Which compiler options must be used to compile the following function using AVX?

void GetOptionPrices(float \* restrict pT, float \* restrict pK, float \* restrict pS0, float \* restrict pC)

{

int i;

float d1, d2, erf1, erf2;

for (i = 0; i < N; i++)

{

d1 = (logf(pS0[i] / pK[i]) + (r + sig \* sig \* 0.5f) \*

pT[i]) / (sig \* sqrtf(pT[i]));

d2 = (logf(pS0[i] / pK[i]) + (r - sig \* sig \* 0.5f) \*

pT[i]) / (sig \* sqrtf(pT[i]));

erf1 = 0.5f + 0.5f \* erff(d1 / sqrtf(2.0f));

erf2 = 0.5f + 0.5f \* erff(d2 / sqrtf(2.0f));

pC[i] = pS0[i] \* erf1 - pK[i] \* expf((-1.0f) \* r \*

pT[i]) \* erf2;

}

}

-5,N

No special options

10,Y

-restrict -mavx

-5,N

-mavx

---

S,N,1,2.0.0.

Which compiler options must be used to compile the following function using AVX?

void GetOptionPricesV4(float \*pT, float \*pK, float \*pS0, float \*pC)

{

int i;

float d1, d2, erf1, erf2;

#pragma simd

for (i = 0; i < N; i++)

{

d1 = (logf(pS0[i] / pK[i]) + (r + sig \* sig \* 0.5f) \*

pT[i]) / (sig \* sqrtf(pT[i]));

d2 = (logf(pS0[i] / pK[i]) + (r - sig \* sig \* 0.5f) \*

pT[i]) / (sig \* sqrtf(pT[i]));

erf1 = 0.5f + 0.5f \* erff(d1 / sqrtf(2.0f));

erf2 = 0.5f + 0.5f \* erff(d2 / sqrtf(2.0f));

pC[i] = pS0[i] \* erf1 - pK[i] \* expf((-1.0f) \* r \*

pT[i]) \* erf2;

}

}

-5,N

No special options, #pragma simd is enough

10,Y

-mavx

-5,N

-O3

---

S,N,1,2.0.0.

The following code need to be built with AVX. Is any modification of the code required?

void GetOptionPricesV7(float \*pT, float \*pK, float \*pS0, float \*pC)

{

int i;

float d1, d2, erf1, erf2, invf;

float sig2 = sig \* sig;

#pragma omp parallel for private(invf, d1, d2, erf1, erf2)

for (i = 0; i < N; i++)

{

invf = invsqrtf(sig2 \* pT[i]);

d1 = (logf(pS0[i] / pK[i]) + (r + sig2 \* 0.5f) \*

pT[i]) \* invf;

d2 = (logf(pS0[i] / pK[i]) + (r - sig2 \* 0.5f) \*

pT[i]) \* invf;

erf1 = 0.5f + 0.5f \* erff(d1 \* invsqrt2);

erf2 = 0.5f + 0.5f \* erff(d2 \* invsqrt2);

pC[i] = pS0[i] \* erf1 - pK[i] \* expf((-1.0f) \* r \*

pT[i]) \* erf2;

}

}

10,Y

No

-5,N

Restrict keyword must be added at declaration of function parameters

-5,N

#pragma simd must be added before the loop

---

S,N,1,2.0.0.

#pragma offload target(mic) compiler directive is used for:

-3,N

Execution of the following code block on coprocessor 0.

-3,N

Execution of the following code block on any available coprocessor, or runtime error in case all coprocessors are busy or there are no coprocessors.

10,Y

Execution of the following code block on any available coprocessor, or on CPU in case all coprocessors are busy or there are no coprocessor.

-3,N

Execution of the following code block on CPU.

---

# 09\_Lecture. Optimization of applications for Intel Xeon Phi: Intel MKL, Intel VTune Amplifier XE

M,N,1,2.0.0.

Choose correct ways of enabling Automatic Offload for Intel MKL:

5,Y

Calling function mkl\_mic\_enable();

5,Y

Setting environment variable MKL\_MIC\_ENABLE=1

-5,N

Calling function mkl\_mic\_set\_Workdivision(MKL\_TARGET\_MIC, 0, 0.5);

-5,N

Setting environment variable MKL\_HOST\_WORKDIVISION=50

---

M,N,1,2.0.0.

Choose correct ways of disabling Automatic Offload for Intel MKL:

3,Y

Calling function mkl\_mic\_disable();

-9,N

Setting environment variable MKL\_MIC\_DISABLE=1

3,Y

Calling function mkl\_mic\_set\_workdivision(MIC\_TARGET\_HOST, 0, 1.0);

3,Y

Setting environment variable MKL\_HOST\_WORKDIVISION=100

---

M,N,1,2.0.0.

Which of the following ways can be used to set 30% work load of MKL on a coprocessor 0 in Automatic Offload mode?

3,Y

Calling function mkl\_mic\_set\_Workdivision(MKL\_TARGET\_MIC, 0, 0.3);

-3,N

Calling function mkl\_mic\_set\_Workdivision(MKL\_TARGET\_MIC, 0, 0.7);

3,Y

Setting environment variable MKL\_MIC\_0\_WORKDIVISION=0.3

-3,N

Setting environment variable MKL\_MIC\_0\_WORKDIVISION=0.7

---

S,N,1,2.0.0.

Which of the following analysis types of Amplifier detects bottlenecks and pieces of code that take most computational time?

10,Y

Hotspots

-5,N

Concurrency

-5,N

Locks and Waits

---

S,N,1,2.0.0.

Which of the following analysis types of Amplifier shows efficiency of utilization of cores, demonstrates quality of parallelization and pieces of code that need parallelization?

-5,N

Hotspots

10,Y

Concurrency

-5,N

Locks and Waits

---

S,N,1,2.0.0.

Which of the following analysis types of Amplifier shows thread synchronization points and evaluates efficiency of synchronization?

-5,N

Hotspots

-5,N

Concurrency

10,Y

Locks and Waits

---

S,N,1,2.0.0.

Choose an appropriate description of Hotspot analysis of Amplifier:

10,Y

Detects computational bottlenecks, most time consuming functions and pieces of code. Is mostly used in the first stage of optimization.

-5,N

Shows efficiency of utilization of cores, demonstrates quality of parallelization and pieces of code that need parallelization.

-5,N

Shows thread synchronization points and evaluates efficiency of synchronization.

---

S,N,1,2.0.0.

Choose an appropriate description of Concurrency analysis of Amplifier:

-5,N

Detects computational bottlenecks, most time consuming functions and pieces of code. Is mostly used in the first stage of optimization.

10,Y

Shows efficiency of utilization of cores, demonstrates quality of parallelization and pieces of code that need parallelization.

-5,N

Shows thread synchronization points and evaluates efficiency of synchronization.

---

S,N,1,2.0.0.

Choose an appropriate description of Locks and Waits analysis of Amplifier:

-5,N

Detects computational bottlenecks, most time consuming functions and pieces of code. Is mostly used in the first stage of optimization.

-5,N

Shows efficiency of utilization of cores, demonstrates quality of parallelization and pieces of code that need parallelization.

10,Y

Shows thread synchronization points and evaluates efficiency of synchronization.

---

S,N,1,2.0.0.

Which of the following analysis types of Amplifier detects most time consuming functions and pieces of code on Xeon Phi?

10,Y

Lightweight Hotspots

-5,N

General Exploration

-5,N

Bandwidth

---

S,N,1,2.0.0.

Which of the following analysis types of Amplifier detects microarchitectural reasons of performance degradation on Xeon Phi?

-5,N

Lightweight Hotspots

10,Y

General Exploration

-5,N

Bandwidth

---

S,N,1,2.0.0.

Which of the following analysis types of Amplifier is used to investigate memory bandwidth on Xeon Phi?

-5,N

Lightweight Hotspots

-5,N

General Exploration

10,Y

Bandwidth

---

S,N,1,2.0.0.

Choose an appropriate description of Lightweight Hotspots analysis of Amplifier on Xeon Phi:

10,Y

Detects most time consuming functions and pieces of code on Xeon Phi

-5,N

Detects microarchitectural reasons of performance degradation on Xeon Phi

-5,N

Investigates memory bandwidth on Xeon Phi

---

S,N,1,2.0.0.

Choose an appropriate description of General Exploration analysis of Amplifier on Xeon Phi:

-5,N

Detects most time consuming functions and pieces of code on Xeon Phi

10,Y

Detects microarchitectural reasons of performance degradation on Xeon Phi

-5,N

Investigates memory bandwidth on Xeon Phi

---

S,N,1,2.0.0.

Choose an appropriate description of Bandwidth analysis of Amplifier on Xeon Phi:

-5,N

Detects most time consuming functions and pieces of code on Xeon Phi

-5,N

Detects microarchitectural reasons of performance degradation on Xeon Phi

10,Y

Investigates memory bandwidth on Xeon Phi

---

S,N,1,2.0.0.

Choose an appropriate description of cycles per instruction (CPI) metric?

9,Y

Indicates how memory latency influences performance

-3,N

Shows average ratio of vector operations per cache access

-3,N

Shows ratio of vector operations to vector instructions executed by a thread

-3,N

Shows memory bandwidth

---

S,N,1,2.0.0.

Choose an appropriate description of compute to data access ratio metric

-3,N

Indicates how memory latency influences performance

9,Y

Shows average ratio of vector operations per cache access

-3,N

Shows ratio of vector operations to vector instructions executed by a thread

-3,N

Shows memory bandwidth

---

S,N,1,2.0.0.

Choose an appropriate description of vectorization intensity metric

-3,N

Indicates how memory latency influences performance

-3,N

Shows average ratio of vector operations per cache access

9,Y

Shows ratio of vector operations to vector instructions executed by a thread

-3,N

Shows memory bandwidth

---

S,N,1,2.0.0.

Choose an appropriate description of memory bandwidth metric?

-3,N

Indicates how memory latency influences performance

-3,N

Shows average ratio of vector operations per cache access

-3,N

Shows ratio of vector operations to vector instructions executed by a thread

9,Y

Shows memory bandwidth

---

S,N,1,2.0.0.

Which of the following performance metrics indicates how memory latency influences performance?

9,Y

cycles per instruction, CPI

-3,N

compute to data access ratio

-3,N

vectorization intensity

-3,N

memory bandwidth

---

S,N,1,2.0.0.

Which of the following performance metrics shows average ratio of vector operations per cache access?

-3,N

cycles per instruction, CPI

9,Y

compute to data access ratio

-3,N

vectorization intensity

-3,N

memory bandwidth

---

S,N,1,2.0.0.

Which of the following performance metrics shows ratio of vector operations to vector instructions executed by a thread?

-3,N

cycles per instruction, CPI

-3,N

compute to data access ratio

9,Y

vectorization intensity

-3,N

memory bandwidth

---

S,N,1,2.0.0.

Which of the following performance metrics shows memory bandwidth?

-3,N

cycles per instruction, CPI

-3,N

compute to data access ratio

-3,N

vectorization intensity

9,Y

memory bandwidth

---

M,N,1,2.0.0.

Intel MKL does not support routines of the following areas:

-5,N

Linear algebra

-5,N

FFT

5,Y

Computer vision

---

M,N,1,2.0.0.

What kind of information about program execution can be retrieved by using compiler options "-profile-functions -profile-loops=all -profile-loops-report=2"?

3,Y

Which functions take most time

3,Y

Which loops take most time

-9,N

Average, minimum and maximum run time of each function

3,Y

Average, minimum and maximum number of loop iterations

---

S,N,1,2.0.0.

Which of the following tools should be used to detect problems related to memory and multithreading?

10,Y

Intel Inspector XE

-5,N

Intel VTune Amplifier XE

-5,N

Intel Compiler with options "-profile-functions -profile-loops=all -profile-loops-report=2" and Loop Profile Viewer

---

S,N,1,2.0.0.

Which of the following tools should be used to retrieve information about functions and loops taking most computational time as well as average, minimum and maximum number of iterations?

-5,N

Intel Inspector XE

-5,N

Intel VTune Amplifier XE

10,Y

Intel Compiler with options "-profile-functions -profile-loops=all -profile-loops-report=2" and Loop Profile Viewer

---

S,N,1,2.0.0.

Which of the following tools should be used to retrieve information about efficiency of multithreaded execution of an application?

-5,N

Intel Inspector XE

10,Y

Intel VTune Amplifier XE

-5,N

Intel Compiler with options "-profile-functions -profile-loops=all -profile-loops-report=2" and Loop Profile Viewer

---

S,N,1,2.0.0.

Inspector can be used for Xeon Phi applications to:

10,Y

Detect data races and deadlocks when launched on CPU (with disabled offload).

-5,N

Detect problems of parallel efficiency on CPU (with disabled offload).

-5,N

Retrieve information about most time consuming functions and loops as well as average, minimum and maximum number of iterations.

---

S,N,1,2.0.0.

Amplifier can be used for Xeon Phi applications to:

-5,N

Detect data races and deadlocks when launched on CPU (with disabled offload).

10,Y

Detect problems of parallel efficiency on CPU (with disabled offload).

-5,N

Retrieve information about most time consuming functions and loops as well as average, minimum and maximum number of iterations.

---

S,N,1,2.0.0.

Loop Profile Viewer (combined with appropriate compiler options) can be used for Xeon Phi applications to:

-5,N

Detect data races and deadlocks when launched on CPU (with disabled offload).

-5,N

Detect problems of parallel efficiency on CPU (with disabled offload).

10,Y

Retrieve information about most time consuming functions and loops as well as average, minimum and maximum number of iterations.

---

M,N,1,2.0.0.

Amplifier is capable of:

3,Y

Detecting functions and pieces of code that take most computational time. Analyzing call stack and source code

3,Y

Measuring processor events that influence performance, e.g. cache misses, branch mispredictions.

-9,N

Giving recommendations on efficient parallelization of pieces of code

3,Y

Measuring waiting time in synchronization and CPU load

---

S,N,1,2.0.0.

How to launch application "./program\_name" in offload mode with parameter 10 from Amplifier GUI?

9,Y

Set "./program\_name" in Application and "10" in Application parameters

-3,N

Set "./program\_name 10" in Application and leave Application parameters empty

-3,N

Set "ssh" in Application and "mic0 ./program\_name 10" in Application parameters

-3,N

Set "ssh mic0 ./program\_name" in Application and "10" in Application parameters

---

S,N,1,2.0.0.

How to launch application "./program\_name" in native mode with parameter 10 from Amplifier GUI?

-3,N

Set "./program\_name" in Application and "10" in Application parameters

-3,N

Set "./program\_name 10" in Application and leave Application parameters empty

9,Y

Set "ssh" in Application and "mic0 ./program\_name 10" in Application parameters

-3,N

Set "ssh mic0 ./program\_name" in Application and "10" in Application parameters

---

S,N,1,2.0.0.

How to launch Lightweight Hotspots analysis in Amplifier for offload mode?

10,Y

amplxe-cl –collect knc-lightweight-hotspots –knob target-cards=0,1 –result-dir ./program\_cmd -- ./program.out

-5,N

amplxe-cl –collect knc-lightweight-hotspots –reslut-dir ./program-cmd -- ssh mic0 “export LD\_LIBRARY\_PATH=~/;

export OMP\_NUM\_THREADS=244; export KMP\_AFFINITY=balanced;

./program.out”

-5,N

amplxe-cl –collect-with runsa-knc –knob event-config=CPU\_CLK\_UNHALTED,L2\_DATA\_READ\_MISS\_MEM\_FILL:sa=1000,

L2\_DATA\_WRITE\_MISS\_MEM\_FILL,L2\_VICTIM\_REQ\_WITH\_DATA,SNP\_HINT\_L2,HWP\_L2MISS –knob target-cards=0,1 –result-dir ./program-cmd -- ./program.out

---

S,N,1,2.0.0.

How to launch Lightweight Hotspots analysis in Amplifier for native mode?

-5,N

amplxe-cl –collect knc-lightweight-hotspots –knob target-cards=0,1 –result-dir ./program\_cmd -- ./program.out

10,Y

amplxe-cl –collect knc-lightweight-hotspots –reslut-dir ./program-cmd -- ssh mic0 “export LD\_LIBRARY\_PATH=~/;

export OMP\_NUM\_THREADS=244; export KMP\_AFFINITY=balanced;

./program.out”

-5,N

amplxe-cl –collect-with runsa-knc –knob event-config=CPU\_CLK\_UNHALTED,L2\_DATA\_READ\_MISS\_MEM\_FILL:sa=1000,

L2\_DATA\_WRITE\_MISS\_MEM\_FILL,L2\_VICTIM\_REQ\_WITH\_DATA,SNP\_HINT\_L2,HWP\_L2MISS –knob target-cards=0,1 –result-dir ./program-cmd -- ./program.out

---

S,N,1,2.0.0.

How to launch analysis with specified performance metrics in Amplifier for offload mode?

-5,N

amplxe-cl –collect knc-lightweight-hotspots –knob target-cards=0,1 –result-dir ./program\_cmd -- ./program.out

-5,N

amplxe-cl –collect knc-lightweight-hotspots –reslut-dir ./program-cmd -- ssh mic0 “export LD\_LIBRARY\_PATH=~/;

export OMP\_NUM\_THREADS=244; export KMP\_AFFINITY=balanced;

./program.out”

10,Y

amplxe-cl –collect-with runsa-knc –knob event-config=CPU\_CLK\_UNHALTED,L2\_DATA\_READ\_MISS\_MEM\_FILL:sa=1000,

L2\_DATA\_WRITE\_MISS\_MEM\_FILL,L2\_VICTIM\_REQ\_WITH\_DATA,SNP\_HINT\_L2,HWP\_L2MISS –knob target-cards=0,1 –result-dir ./program-cmd -- ./program.out

---

S,N,1,2.0.0.

How to access general statistics of a specific launch using Amplifier?

10,Y

amplxe-cl –report summary –r ./offload\_cmd/

-5,N

amplxe-cl –report hotspots –r ./offload\_cmd/

-5,N

amplxe-cl –report hw-events –r ./offload\_cmd/

---

S,N,1,2.0.0.

How to retrieve a list of slowest functions of a specific launch using Amplifier?

-5,N

amplxe-cl –report summary –r ./offload\_cmd/

10,Y

amplxe-cl –report hotspots –r ./offload\_cmd/

-5,N

amplxe-cl –report hw-events –r ./offload\_cmd/

---

S,N,1,2.0.0.

How to retrieve information of hardware events of a specific launch using Amplifier?

-5,N

amplxe-cl –report summary –r ./offload\_cmd/

-5,N

amplxe-cl –report hotspots –r ./offload\_cmd/

10,Y

amplxe-cl –report hw-events –r ./offload\_cmd/

---

# 10\_Practice. Optimization of matrix multiplication. Improving memory efficiency

S,N,1,2.0.0.

Which of the following execution modes does Intel MKL support?

-5,N

Only on multicore CPUs.

-5,N

On multicore CPUs or Xeon Phi in coprocessor-only mode.

10,Y

On multicore CPUs or Xeon Phi in coprocessor-only and offload modes.

---

S,N,1,2.0.0.

Which mode of MKL is best suited for automatic execution of its routines on Xeon Phi and requires minimal code modification (given the main program runs on CPU)?

-5,N

Native Execution

-5,N

Compiler Assisted Offload, CAO

10,Y

Automatic Offload, AO

---

S,N,1,2.0.0.

Which mode of MKL provides maximum control of data transfers between host and Xeon Phi (given the main program runs on CPU)?

-5,N

Native Execution

10,Y

Compiler Assisted Offload, CAO

-5,N

Automatic Offload, AO

---

S,N,1,2.0.0.

Which mode of MKL allows to run its routines on coprocessor (given CPU is not used, the main program runs on Xeon Phi)?

10,Y

Native Execution

-5,N

Compiler Assisted Offload, CAO

-5,N

Automatic Offload, AO

---

S,N,1,2.0.0.

Which of the following types of BLAS routines can work on Xeon Phi in Automatic Offload mode?

-3,N

\*GEMM

-3,N

\*TRSM and \*TRMM

9,Y

\*GEMM, \*TRSM and \*TRMM

-3,N

All BLAS routines

---

S,N,1,2.0.0.

Which of the following types of BLAS routines can work on Xeon Phi in Compiler Assisted Offload mode?

-3,N

\*GEMM

-3,N

\*TRSM and \*TRMM

-3,N

\*GEMM, \*TRSM and \*TRMM

9,Y

All BLAS routines

---

S,N,1,2.0.0.

Which of the following types of BLAS routines can work on Xeon Phi in native mode?

-3,N

\*GEMM

-3,N

\*TRSM and \*TRMM

-3,N

\*GEMM, \*TRSM and \*TRMM

9,Y

All BLAS routines

---

S,N,1,2.0.0.

Which parameter of #pragma offload is used to explicitly specify the device to execute the code on?

-3,N

out

-3,N

in

10,Y

target

-3,N

signal

---

S,N,1,2.0.0.

Which parameter of #pragma offload is used for asynchronous execution on coprocessor?

-3,N

out

-3,N

in

-3,N

target

10,Y

signal

---