Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

CoreNEURON: Performance and Energy Efficiency Evaluation on Intel and Arm CPUs

EAHPC 2020

Joel Criado[1], Marta Garcia[1],
Pramod Kumbhar[2], Omar Awile[2],
Ioannis Magkanaris[2], Filippo Mantovani[1]

[1] Barcelona Supercomputing Center, Spain
[2] Ecole Polytechnique Fédérale de Lausanne, Switzerland

14th September, 2020

Contents

- Introduction
- Environment
  - CoreNEURON
  - NMODL Framework
  - Hardware
- Methodology
- Evaluation
  - Performance
  - Instruction Mix
  - Energy Efficiency
- Conclusions and Future Work

Introduction

- The neuroscience community is in the race of understanding the human brain functioning.
  - Nowadays, models are becoming more complex and larger.
  - The community has developed applications to cover this demand.
- We analyzed CoreNEURON to understand how it works in the complex world of HPC.
  - Different architectures and compilers
- Key points:
  - Detailed study of the instruction mix.
  - Performance and energy measurements from state-of-the art systems.
  - Focus on vectorization and SIMD units.

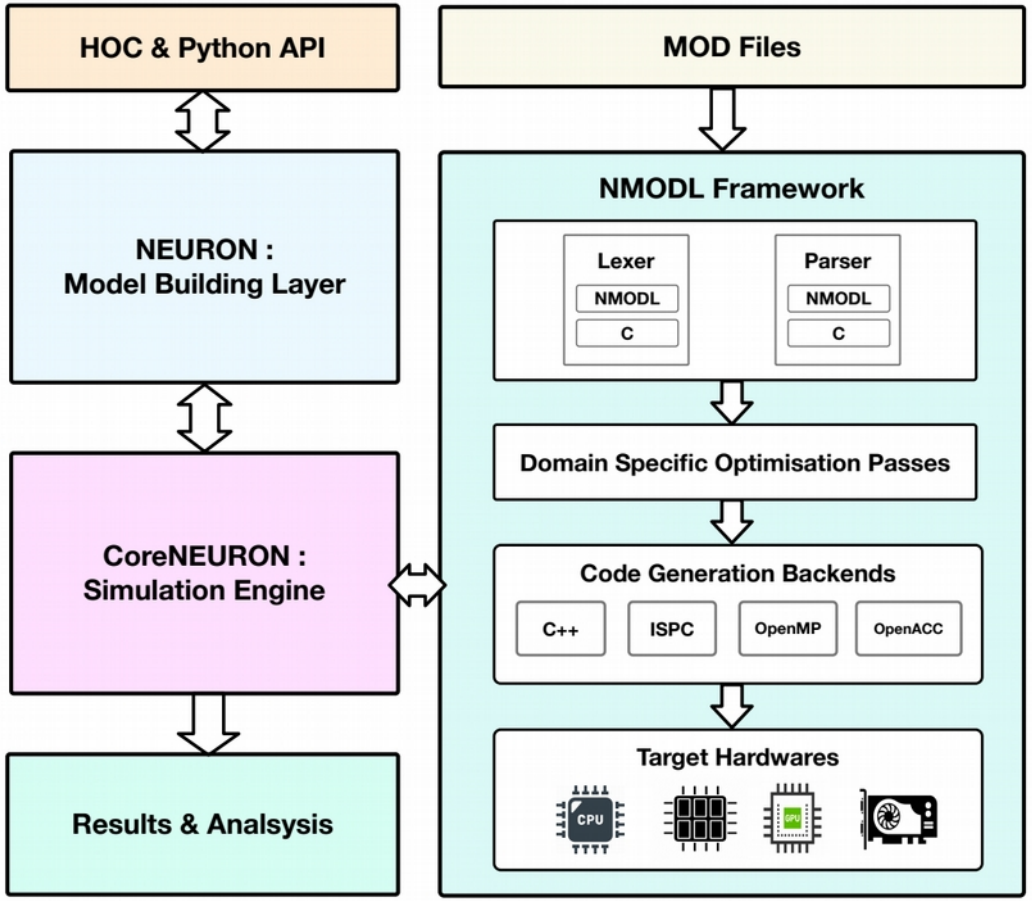
Environment: CoreNEURON

- Based on the NEURON simulator.
- Designed for morphologically detailed neuron models.
- Uses a domain-specific language (DSL), called NMODL, to describe neuron mechanisms.
  - DSL translated to C++ using the MOD2C source-to-source compiler.
  - NMODL source-to-source compiler replaces MOD2C.

P. Kumbhar et al., “An optimizing multi-platform source-to-source compiler framework for the neuron modeling language” in International Computational Science. Springer, 2020, pp. 45-58.

Environment: NMODL Framework

- Translates the DSL to different target languages/programming models
- Targets different architectures
- Focus on ISPC (Intel SPMD Program Compiler) backend for Arm and x86
  - ISPC targets the specific SIMD of each architecture



Environment: Hardware

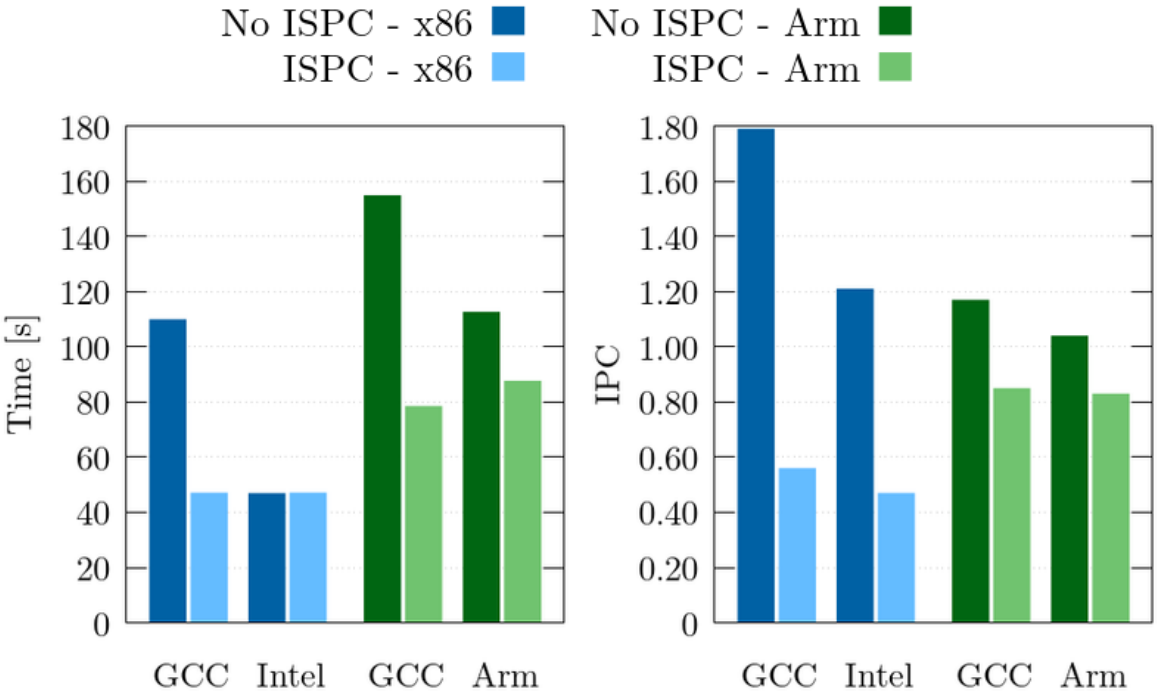
- Dibona-TX2:
  - x2 Marvell ThunderX2 CN9980 at 2.0 GHz and 32 cores
  - 256 GB DDR4 2666 MHz main memory
  - 128 bits SIMD vector width
  - 40 nodes
- MareNostrum4:
  - x2 Intel x86 Skylake 8160 at 2.1 GHz and 24 cores
  - 96 GB DDR4 3200 MHz main memory
  - 128 / 256 / 5122 bits SIMD vector width
  - 3456 nodes

Methodology

- Input used: synthetic model consisting of a multiple ring network of branching neurons.
- Different comparisons:
  - Hardware: Armv8 vs. x86
  - Compiler: Generic (GCC 8) vs. vendor-specific (Intel 2019.5, Arm HPC 20.1)
  - Application: ISPC vs. without ISPC
- Study targeting vectorization:
  - Metrics gathered for the most computing-intensive kernels (+90% of execution time).

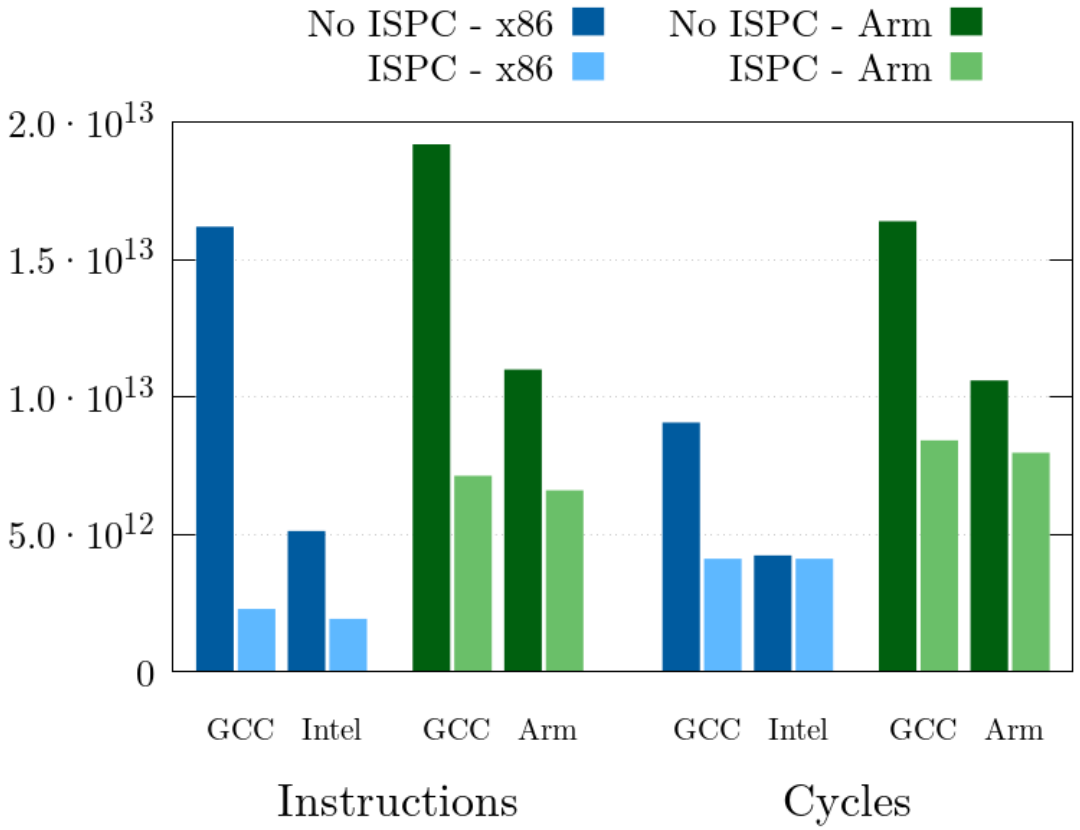
Evaluation: Performance

- In x86, ISPC improves the performance just for GCC. Both Intel and GCC with ISPC have the same execution time.
  - Although having the same execution time, IPCs are very different.
- In Armv8, both compilers benefit from ISPC.
- ISPC versions have lower IPC than no ISPC ones. This comes from a different usage of SIMD units.



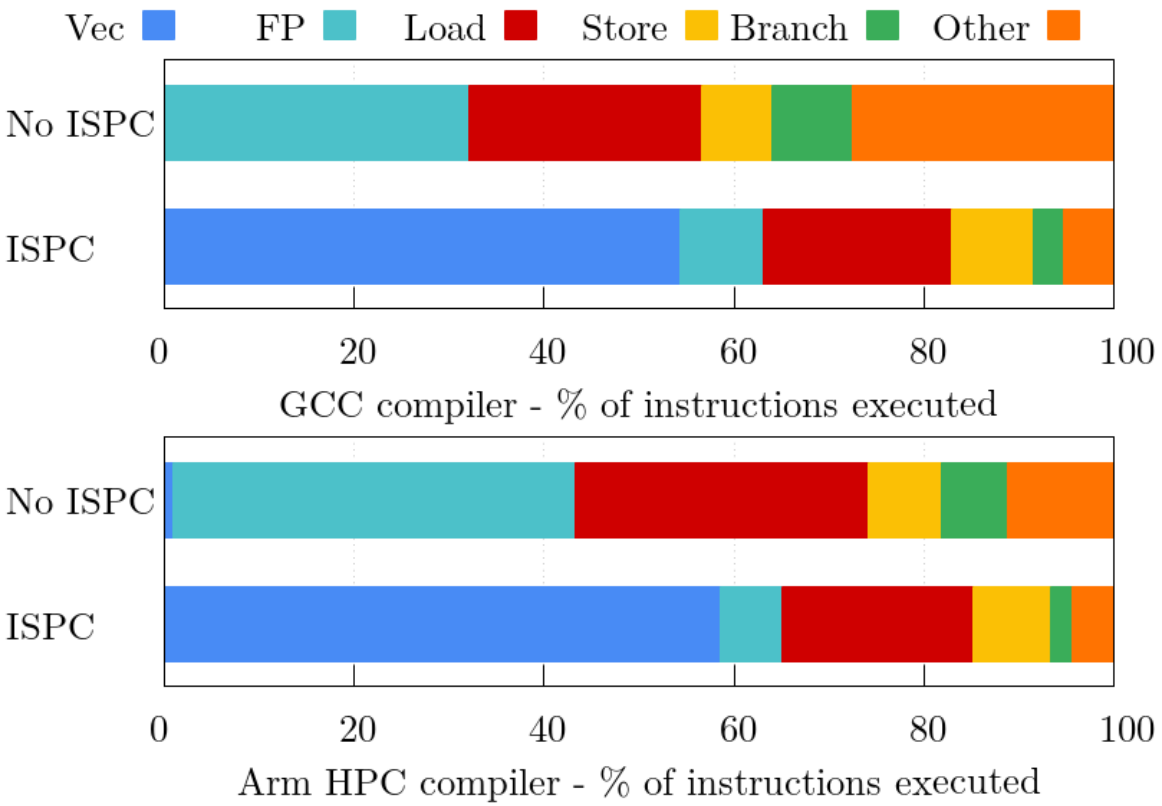
Evaluation: Performance

- Cycles show similar behaviour to the Elapsed Time. There's no variation in the frequency.
- When using ISPC, the number of instructions is similar for each architecture.
- ISPC is able to drastically reduce, especially with GCC, the number of instructions.



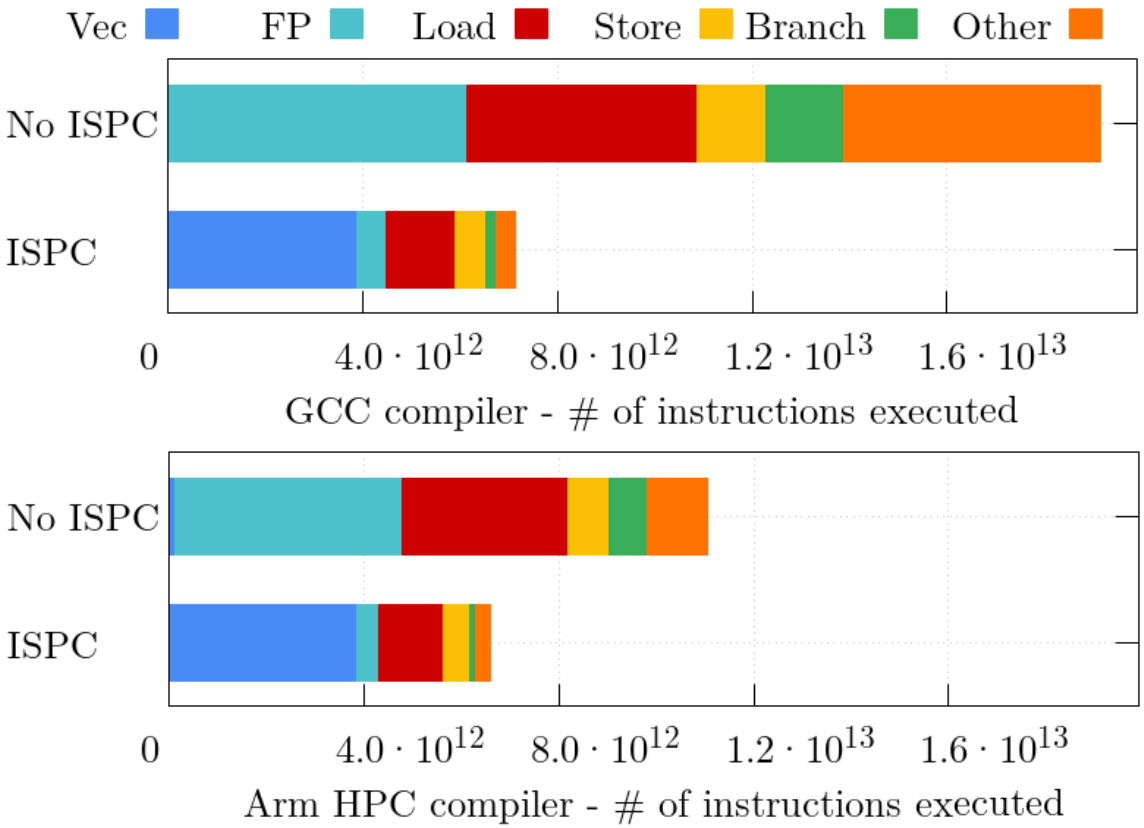
Evaluation: Instruction Mix

- Without ISPC, the program doesn't have vector instructions
  - From less than 0.1% to more than 50%
- FP instructions become vectorial with ISPC.



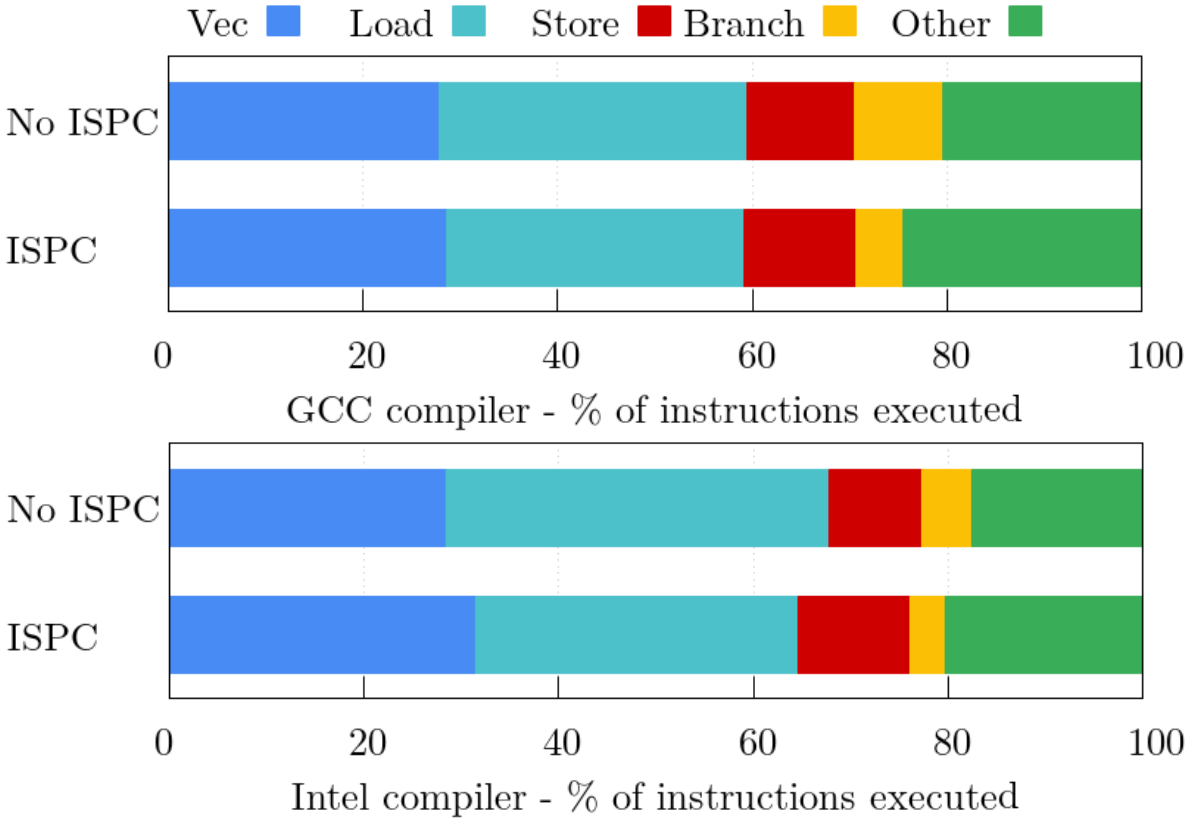
Evaluation: Instruction Mix

- ~50% reduction in the number of instructions without ISPC
- 3 times less instructions for GCC, 2 times for Arm
- ISPC code is also optimized in terms of total instructions and number of branches



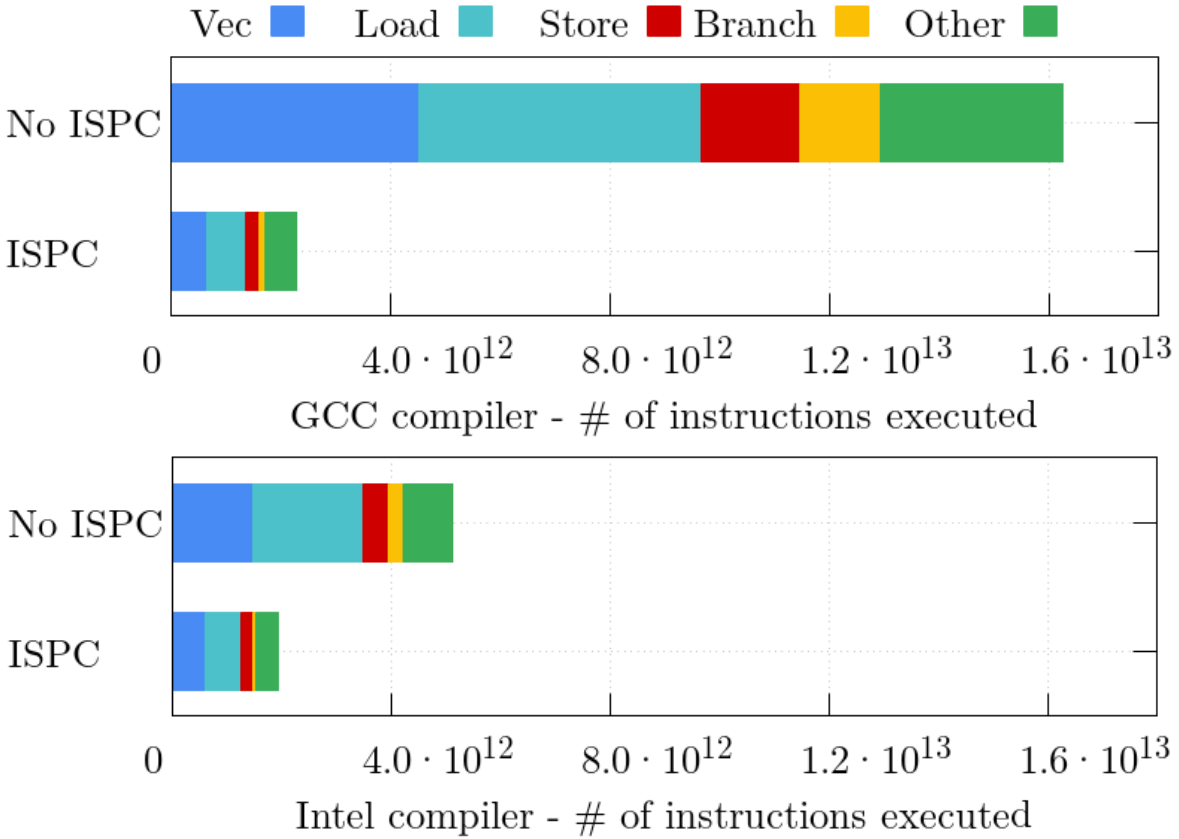
Evaluation: Instruction Mix

- Ratios are very similar with and without ISPC
- In contrast to Armv8 results, without ISPC the x86 compilers issue vector instructions.



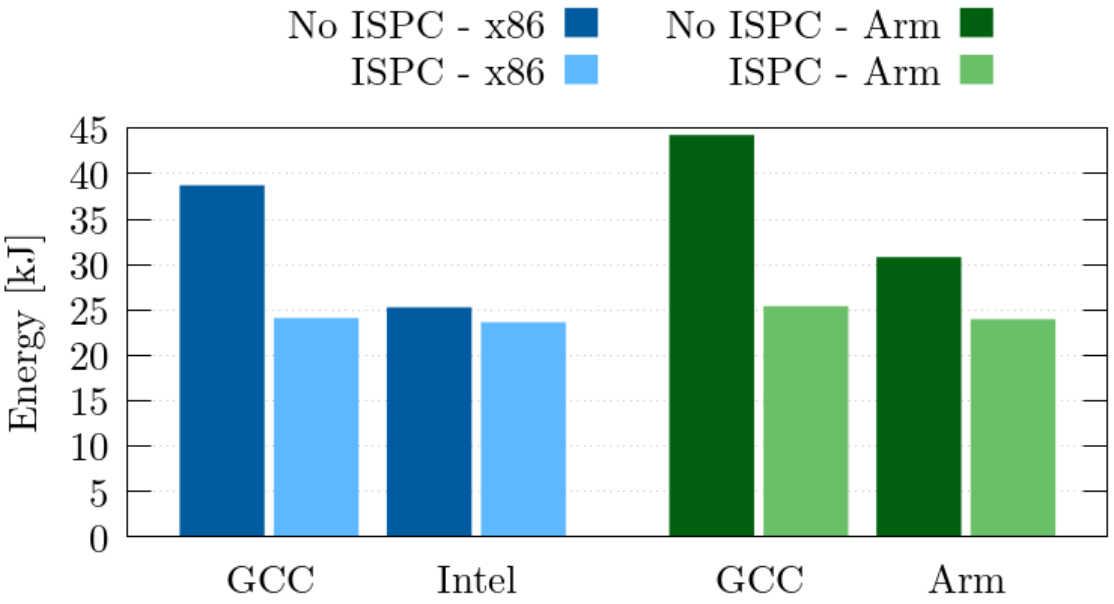
Evaluation: Instruction Mix

- ~7 times less instructions for GCC.
  - Analyzing the binary, we detect that No ISPC version uses SSE SIMD and ISPC uses AVX-512.
- ~2.7 times less instructions for Intel.
  - In this case, No ISPC version uses AVX2 extension.
- Similar to Armv8, ISPC code is optimized in terms of number of branches and total instructions.



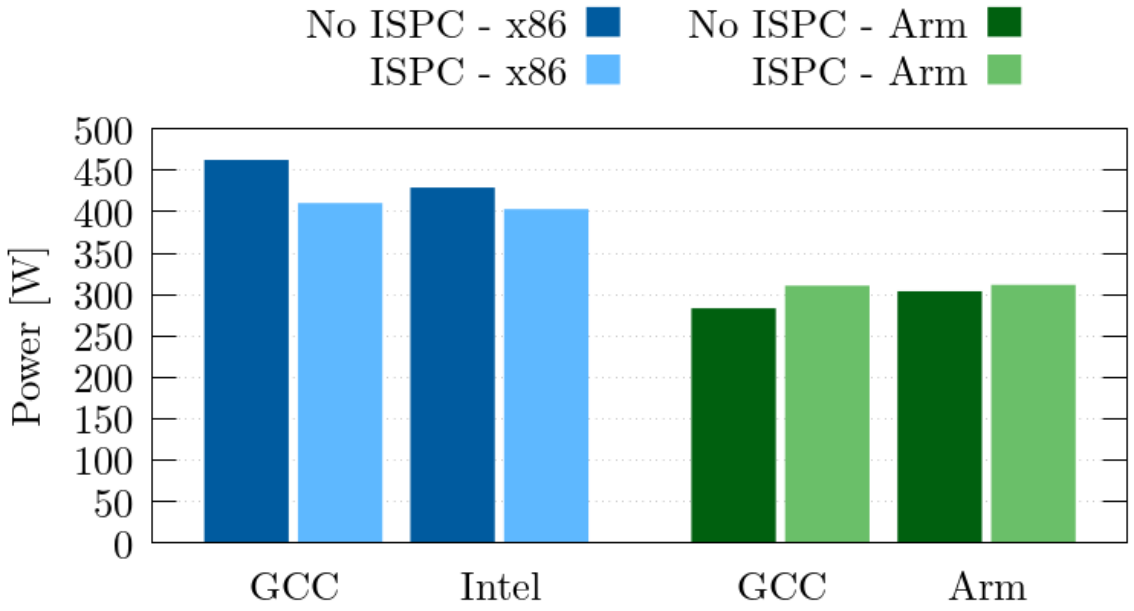
Evaluation: Energy Efficiency

- The less consuming architecture is the fastest one.
- Vendor-specific compilers allow to reach a lower energy-to-solution.
- ISPC version consumes nearly the same Energy in every scenario.



Evaluation: Energy Efficiency

- Average power on x86 is 433 W, while on Armv8 is 297 W.
- Arm executions without ISPC consume less power.
  - Power saved from not using the NEON SIMD extension.



Conclusions and Future Work

- Evaluation of the neural simulator CoreNEURON in Arm and Intel architectures, trying different compilers, and the ISPC backend for the DSL.
  - Speedups from 1.2 to 2.3 times achieved using ISPC.
- Concerning the Arm ecosystem:
  - Arm HPC compiler works fine with a complex application and delivers better performance than GCC.
  - Armv8 Marvell ThunderX2 CPUs are 1.5 times slower than Intel Skylake.
  - Nevertheless, they achieve the same energy consumption.
- Future work:
  - Analysis of memory usage.
  - Use a more realistic input case.
  - Perform tests in the newest Arm CPUs.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

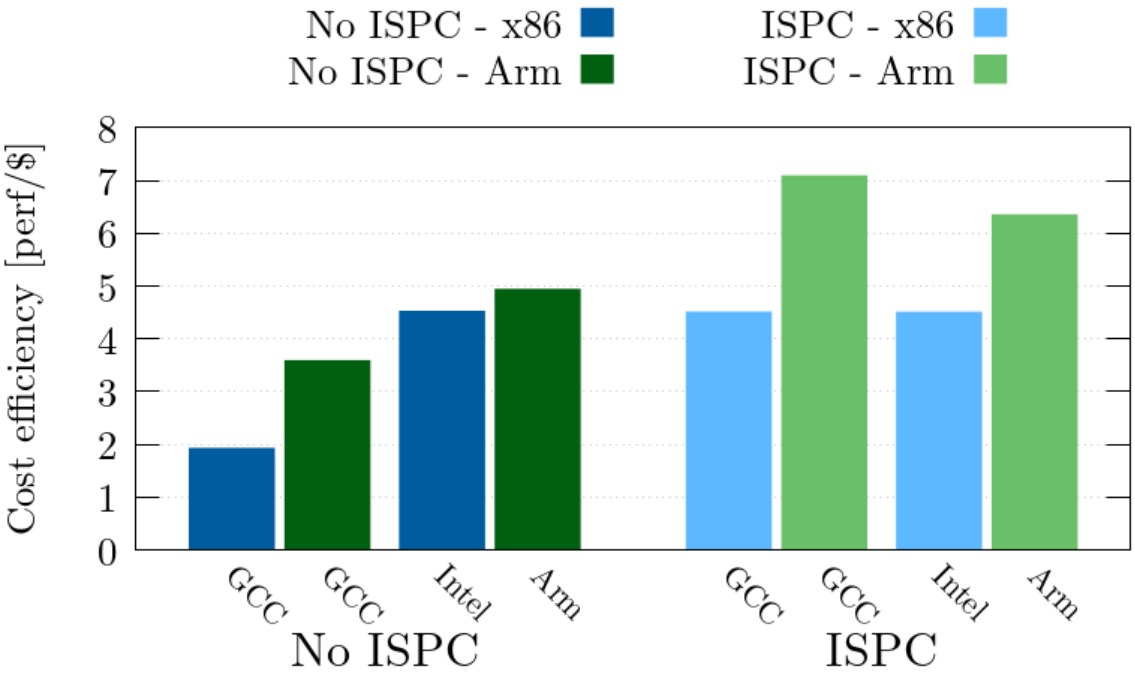
Thank you

CoreNEURON: Performance and Energy Efficiency Evaluation on Intel and Arm CPUs

joel.criado@bsc.es

Evaluation: Cost vs Performance

- Prices based on the official recommended retail price.
- Every combination of Armv8 shows a better ratio than x86.
  - Up to 57% higher when using ISPC.



Conclusions and Future Work

- Evaluation of the neural simulator CoreNEURON in Arm and Intel architectures, trying different compilers, and the ISPC backend for the DSL.
  - Speedups from 1.2 to 2.3 times achieved using ISPC.
- Concerning the Arm ecosystem:
  - Arm HPC compiler works fine with a complex application and delivers better performance than GCC.
  - Armv8 Marvell ThunderX2 CPUs are 1.5 times slower than Intel Skylake.
  - Nevertheless, they achieve the same energy consumption.
  - Armv8 Marvell ThunderX2 shows a higher cost efficiency, between 1.3 and 1.5 times better than x86.
- Future work:
  - Analysis of memory usage.
  - Use a more realistic input case.
  - Perform tests in the newest Arm CPUs.