# Benchmarking a hemodynamics application on Intel based HPC systems

Ferdinando AURICCHIO[b], Marco FEDELE[b], Marco FERRETTI[a], Adrien LEFIEUX[c], Rodrigo ROMAROWSKI[b], Luigi SANTANGELO[a], Alessandro VENEZIANI[c,d]
[a] *University of Pavia, Department of Computer Engineering, Pavia, Italy*
[b] *University of Pavia, Department of Civil Engineering and Architecture, Pavia, Italy*
[c] *Emory University, Department of Mathematics and Computer Science, Atlanta, USA*
[d] *School of Advanced Studies IUSS, Pavia, Italy*

**Abstract.** Three different INTEL based HPC systems are used to benchmark an application of the LifeV library for running simulations of patient-specific cardiovascular hemodynamics. Running times and scalability measures are collected with two real-size experiments. A third small-size test case is used to profile the code, exposing the effect of compiler vectorization, MPI efficiency and memory footprint. Results show that the efficiency of the application is heavily dependent on the performance of the underlining libraries. Also, profiling suggests that new architectural features available in INTEL Knight Landing processors will not bring benefit, unless the internal code is completely redesigned.

**Keywords.** Parallel architectures, MPI, scalability, LifeV, bottlenecks, profiler tools, tracer tools, vectorization, MPI latency, memory footprint, tightly-coupled applications

## 1. Introduction

This paper reports on the effects of porting a cardiovascular hemodynamics application on three different HPC systems that share the basic processor architecture (INTEL microarchitecture in Sandy Bridge, Hashwell and Broadwell processors) and interconnection fabric. The systems used for the experiment are two XSEDE machines (Stampede [5] and Comet [6]), and a PRACE-CINECA one (Marconi A1 partition [9]).

After a brief introduction on the LifeV library [2] used to set up the application to simulate patient-specific cardiovascular hemodynamics, we describe and compare the results obtained running LifeV on the three different architectures, with a specific focus on the application scalability. The ultimate aim is to possibly increase the application performance in order to reduce the running time for a real case problem instance.

By using profiling and tracing tools, we highlight where the application spends most time, the impact of the MPI communications, how many loops are automatically vectorized by the compiler, the relevance of memory footprint. In the end, on the basis of the profiling data collected, we also speculate on the role that the new Knights Landing architecture can play, specifically with its AVX512 instructions set.

The paper is organized as follows: Section 2 gives an overview of LifeV and its applicability to the blood flow simulation. The following Sections 3, 4 and 5 describe the results obtained running the application on Stampede, Comet and Marconi,