EPJ Web of Conferences **173**, 06007 (2018) https://doi.org/10.1051/epjconf/201817306007
Mathematical Modeling and Computational Physics 2017

Nonlinear Wave Simulation on the Xeon Phi Knights Landing Processor

Ivan Hristov[1,2,⋆], *Goran* Goranov[3,⋆⋆], and *Radoslava* Hristova[1,2,⋆⋆⋆]

[1] *JINR, LIT, Dubna, Russia*
[2] *Sofia University, Bulgaria*
[3] *Technical University of Gabrovo, Bulgaria*

Abstract. We consider an interesting from computational point of view standing wave simulation by solving coupled 2D perturbed Sine-Gordon equations. We make an OpenMP realization which explores both thread and SIMD levels of parallelism. We test the OpenMP program on two different energy equivalent Intel architectures: 2× Xeon E5-2695 v2 processors, (code-named "Ivy Bridge-EP") in the Hybrilit cluster, and Xeon Phi 7250 processor (code-named "Knights Landing" (KNL). The results show 2 times better performance on KNL processor.

1 Introduction

The second generation *Intel*®*XeonPhi*^{TM} processors code-named Knights Landing (KNL) are expected to deliver better performance than general purpose CPUs like *Intel*®*Xeon*® processors for applications with both high degree of parallelism and well behaved communications with memory [1]. Compute-bound applications run better on KNL due to its larger (512-bit) vector registers. Bandwidth-bound applications also run better on KNL due to its high bandwidth memory (HBM).

In this work we consider an interesting from computational point of view example of numerical solution of coupled 2D perturbed Sine-Gordon equations. We really need serious computational resources because in some cases the computational domain size may be very large – 10^6–10^8 mesh points and very long time integration is also needed – 10^8–10^9 time steps. Usually applications with stencil operations (like those in the presented work) are bandwidth-bound. The calculation of the transcendental *sine* function however makes our application to be closer to the compute-bound case and hence it benefits both from using HBM and from vectorization.

The considered systems of coupled 2D perturbed Sine-Gordon equations are of practical interest because it is well known that they model the dynamics of the so called Intrinsic Josephson Junctions (IJJ) [2].

The goals of the work are:

- To make an OpenMP realization of a finite difference scheme for solving systems of 2D perturbed Sine-Gordon equations. We want this realization to explore both thread and SIMD levels of parallelism.

⋆e-mail: christov_ivan@abv.bg
⋆⋆e-mail: ph.d.g.goranov@gmail.com
⋆⋆⋆e-mail: radoslava@fmi.uni-sofia.bg

- To test the OpenMP program on two different energy equivalent Intel architectures: 2× Xeon E5-2695 v2 processors with 24 cores, 48 threads, (code-named "Ivy Bridge-EP") in the Hybrilit cluster, and Xeon Phi 7250 processor with 68 cores, 272 threads, (code-named "Knights Landing" (KNL)).

2 Mathematical model and numerical scheme

We consider the following systems of 2D perturbed Sine-Gordon equations :

$$S\,(\varphi_{tt} + \alpha\varphi_t + \sin\varphi - \gamma) = \Delta\varphi, \qquad (x, y) \in \Omega \subset R^2. \tag{1}$$

Here Δ is the 2D Laplace operator. Ω is a given domain in R^2. S is the $N_{\text{eq}} \times N_{\text{eq}}$ cyclic tridiagonal matrix:

$$S = \begin{pmatrix} 1 & s & 0 & . & 0 & s \\ s & 1 & s & 0 & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & . & 0 & s & 1 & s \\ s & . & 0 & 0 & s & 1 \end{pmatrix},$$

where $-0.5 < s \leq 0$ and N_{eq} is the number of equations. The unknown is the column vector $\varphi(x, y, t) = (\varphi_1, \ldots, \varphi_{N_{\text{eq}}})^T$. Neumann boundary conditions are considered:

$$\left.\frac{\partial\varphi}{\partial\vec{n}}\right|_{\partial\Omega} = 0. \tag{2}$$

In (2) $\vec{n}$ denotes the exterior normal to the boundary $\partial\Omega$. To close the problem (1)–(2) appropriate initial conditions are posed. The model (1)–(2) describes very well the dynamics of N_{eq} periodically stacked IJJs [2]. The parameter s represents the inductive coupling between adjacent Josephson junctions, α is the dissipation parameter, γ is the external current. All the units are normalized as in [2].

We follow the approach of construction of the numerical scheme from [3]. We solve the problem numerically in rectangular domains by using second order central finite differences with respect to all of the derivatives. As a result at every mesh point of the domain we have to solve a system with the tridiagonal matrix S. Because of the specific tridiagonal structure of S we need only $(9N_{\text{eq}} - 12)$ floating point operations for solving one system. So the algorithm complexity is $O(N_{\text{eq}} \cdot N_x \cdot N_y \cdot N_{\text{time\_steps}})$, where N_{eq} is the number of equations, N_x is the number of mesh points in x-direction, N_y is the number of mesh points in y-direction and $N_{\text{time\_steps}}$ is the number of steps in time.

3 Parallelization strategy and performance scalability results

OpenMP Fortran realization of the above numerical scheme is made. We store the unknown solution of two consecutive time levels in two multidimensional arrays $U1(N_{\text{eq}}, N_x, N_y)$, $U2(N_{\text{eq}}, N_x, N_y)$. To ensure a good data locality, the main loop over indexes of $U1$ and $U2$ follows the column major order of multidimensional arrays in Fortran language.

To utilise the computational capabilities of the considered processors we explore two levels of parallelism: SIMD parallelism (vectorization) at the inner level and thread parallelism at the outer level. The smallest piece of work which we consider consists of solving 8 successive linear systems with the cyclic tridiagonal matrix S. Such pieces of work are distributed between OpenMP threads. Both the calculation of the right-hand sides for each linear system and solving 8 linear systems at once

are vectorized. Our parallelization strategy consists of a parallelization of the main nested DO loop by using the OMP DO directive [4]:

```
!$OMP DO [ Clauses ]
    DO I=1,N_y
       DO J=1,N_x,8
          DO K=J,J+7
          ..................
          Vectorized loops for calculation of the right-hand sides
          of 8 successive linear systems
          ..................
          ENDDO
          ..................
          Vectorized loops for solving 8 systems at once
          ..................
       ENDDO
    ENDDO
!$OMP END DO
```

The use of `-O2` optimization flag for compiling ensures an automatic vectorization of the innermost loops. Therefore the indexes of the outermost loop are distributed between the OpenMP threads and all the innermost loops (all with length 8) are vectorized. Let us mention that 8 double precision words correspond to the length of one vector register in KNL processors – 512 bit and two vector registers (256 bit each) in Ivy Bridge-EP processors. As a result we achieve about 2 times better performance from vectorization on "Ivy Bridge-EP" and about 4 times better performance from vectorization on KNL processors. The achievement of 50 % effectiveness from vectorization can be explained by the fact that our application is somewhere between bandwith-bound and compute-bound cases. On the one hand the application is of stencil type which type is by rule bandwidth-bound. On the other hand a calculation of the transcendental *sine* function is needed at every mesh point of the computational domain, which makes our application closer to the compute-bound case.

In the following figure the computational domain size is: $N_{\mathrm{eq}} \times N_x \times N_y = 8 \times 4096 \times 4096$. As seen from this figure we achieve good performance scalability on both architectures and 2 times better performance on KNL processor.

4 Numerical example of a nonlinear standing wave

To check that the realized OpenMP program really works, we repeated the numerical results (in 2D case) from the classical works [5, 6]. As explained in these papers the powerful THz radiation from IJJs reported in [7] corresponds to a new type of standing wave solutions with excited so called cavity modes. For certain parameters α and γ the phase $\varphi(x, y, t)$ in a particular equation (junction) is a sum of three terms: a linear with respect to the time term, vt, a static π kink term $pi\_kink(x, y)$ and an oscillating term:

$$\varphi(x, y, t) = vt + pi\_kink(x, y) + oscillating\_term(x, y, t)$$

The oscillating term is approximately a solution of the linear wave equation:

$$\varphi_{tt} = \frac{1}{1 + 2s} \Delta\varphi, \qquad (x, y) \in \Omega \subset R^2$$

with Neumann boundary conditions.

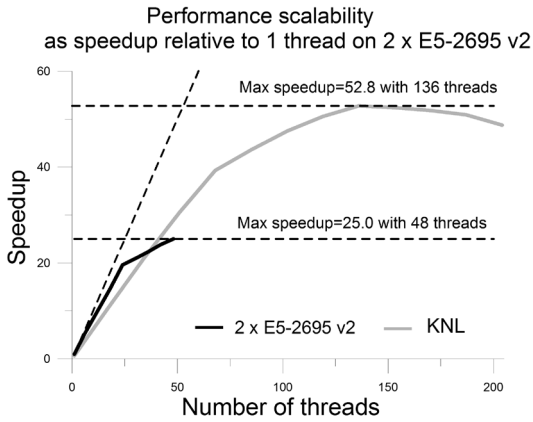

Figure 1. Performance scalability as speedup relative to one vectorized thread on 2 × "Ivy Bridge-EP" processors

As opposite to the oscillating term, which is the same for each junction (equation), the static term (in our case static π kinks) have alternative character, i.e. opposite static π kinks alternate in odd and even junctions (equations).

5 Conclusions

An OpenMP program for solving systems of 2D perturbed Sine-Gordon equations is realized and tested on two different Intel architectures: one computational node in the Hybrilit cluster consisting of two Ivy Bridge-EP processors and a KNL processor provided by RSC Group, Moscow. The results shows 2 times better performance on the KNL processor.

6 Acknowledgements

We greatly thank to the opportunity to use the computational resources of the HybriLIT cluster and to use KNL processors provided by RSC Group, Moscow. This work is supported by the National Science Fund of Bulgaria under Grant DFNI-I02/8 and by the National Science Fund of BMSE under grant I02/9/2014.

References

[1] J. Jeffers, J. Reinders, and A. Sodani, *Intel Xeon Phi Processor High Performance Programming: Knights Landing Edition* (Morgan Kaufmann, 2016)
[2] S. Sakai, P. Bodin, and N.F. Pedersen, Journal of Applied Physics **73** (5), 2411–2418 (1993)
[3] G.S. Kazacha and S.I. Serdyukova, Zhurnal Vychislitelnoi Matematiki i Matematicheskoi Fiziki **33** (3), 417–427 (1993)
[4] B. Chapman, G. Jost, and R. Van Der Pas, *Using OpenMP: portable shared memory parallel programming* (MIT press, 2008)
[5] S. Lin and X. Hu, Physical Review Letters **100** (24), p. 247006 (2008)
[6] A.E. Koshelev, Physical Review B **78** (17), p. 174509 (2008)
[7] L. Ozyuzer, A.E. Koshelev, et al., Science **318** (5854), 1291–1293 (2007)