

Parallel Algorithm for Solving TOV Equations for Sequence of Cold and Dense Nuclear Matter Models

Alexander Ayriyan^{1,*}, Ján Buša Jr.^{1,2}, Hovik Grigorian^{1,3}, and Gevorg Poghosyan⁴

¹Laboratory of Information Technologies, JINR

²Institute of Experimental Physics, SAS, Slovakia

³Department of Theoretical Physics, YSU, Armenia

⁴Karlsruhe Institute of Technology, Germany

Abstract. We have introduced parallel algorithm simulation of neutron star configurations for set of equation of state models. The performance of the parallel algorithm has been investigated for testing set of EoS models on two computational systems. It scales when using with MPI on modern CPUs and this investigation allowed us also to compare two different types of computational nodes.

1 Introduction

Investigation of cold and dense nuclear matter is one of the key problems in modern particle physics. One of the ways of selecting and testing EoS models is comparison of the mechanical characteristics of compact stars given by models to the observational constraints [1]. To obtain mechanical characteristics of compact stars, one has to calculate the mass and radius relation of static spherical symmetry neutron stars. Such relations are obtained as a solution of the Tolman-Oppenheimer-Volkoff (TOV) equations [2, 3]:

$$\begin{cases} \frac{dP(r)}{dr} = -\frac{GM(r)\varepsilon(r)}{r^2} \left(1 + \frac{P(r)}{\varepsilon(r)}\right) \left(1 + \frac{4\pi r^3 P(r)}{M(r)}\right), \\ \frac{dM(r)}{dr} = 4\pi r^2 \varepsilon(r), \end{cases} \quad (1)$$

where r is the coordinate distance from the center, $M(r)$, $\varepsilon(r)$, and $P(r)$ are mass, energy density, and pressure profiles correspondingly, and G is the gravitational constant. Each of these profiles as a solution of TOV equations depends on central value of energy density $\varepsilon(r=0)$. The relation between the radius R where the pressure is $P(r=R)=0$ and the total mass $M=M(r=R)$ are shown in the Fig. 1 for particular cases of realistic models of equation of state HDD [4], DD2 [5], MKVOR and KVORcut02 [6, 7]. These particular models were used as a reference models in investigation of the parallel algorithms.

A parallel algorithm for solving the TOV equations for a set of equations of state (EoS) models has been developed using Message Passing Interface (MPI) [10]. It has been applied to Bayesian

*e-mail: ayriyan@jinr.ru

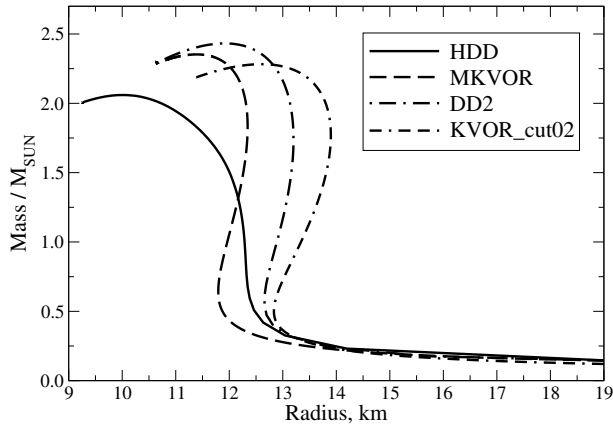


Figure 1. The neutron star mass-radius relations as solutions of TOV equations for different realistic models of equation of state actually used in modern astrophysical applications.

analysis of modern model of stellar nuclear matter [11], where thousands models have been analyzed using the most selective $M-R$ constraints.

2 Parallel algorithm

Solving (integrating) TOV equations for one Equation of State model on current CPU is rather fast. With our settings of parameters we needed about 1500 integration steps (calls to Merson integrator) and it takes about 1.5s to solve such set of equations. The complexity comes from the fact, that we need to calculate several thousands of such models. Fortunately this problem is embarrassingly parallel. Each one model can be solved independently on others. This allows us to use MPI approach.

We have decided to use master-slave model. On the input we have one large text file containing data of EoS models. The number of models is not limited. The main process loads data and distributes it among rest of the processes. Master after reading one set of equations and parameters of the model sends these data to the first available slave process and continues to read following model. After all models have been read, master waits until all slave processes finish their calculations and collects results from slaves by combining results slaves have produced into one large output file.

Slave processes, after obtaining data from master, are responsible in analyzing data and checking their integrity (whether input sequences are monotonous as they should be). After this step, the TOV equations are integrated using the Runge-Kutta-Merson method [8, 9]. Obtained results are written into output files with predefined names. In this step we use the fact, that master and slave processes share the same file system and we don't need to communicate the results back to the master which reduces the load and requirements on master.

3 Results

The analysis of parallel calculations has been done for testing set of 3162 EoS models on two computational systems. This set of modern realistic EoS models has been studied using Bayesian Analysis [11]. The cluster of Joint Supercomputer Center of the Russian Academy of Sciences (JSCC) with 4× Intel Xeon Phi 7250 using AVX512 (red line on presented figures) and heterogeneous cluster

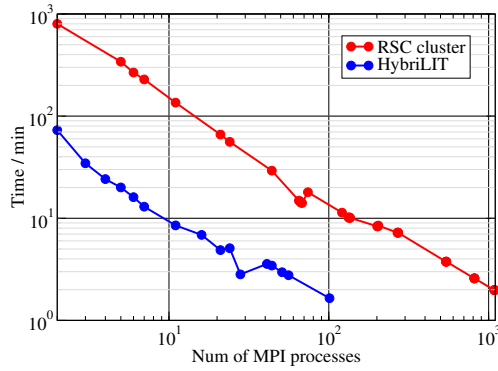


Figure 2. The computational time of parallel computations.

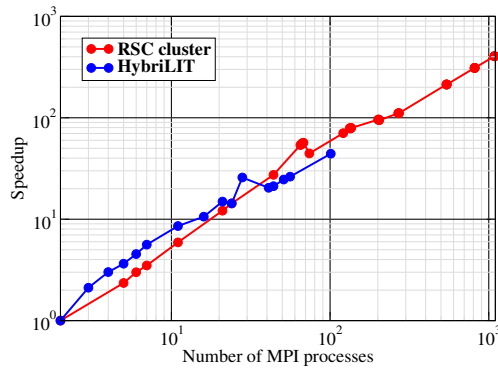


Figure 3. The speedup of parallel computations.

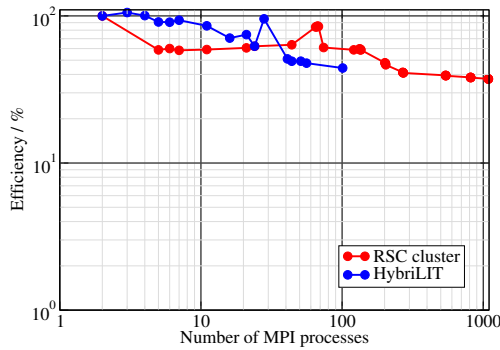


Figure 4. The efficiency of parallel computations.

“HybriLIT” cluster with $2\times$ Intel Xeon E5-2695v3 using AVX2, as it doesn’t support AVX512 (blue line). To build our program we have used Intel Compiler with optimization flag -O3.

The computational time, the speedup, and the efficiency of the parallel calculations are shown on Figs. 2, 3, and 4, respectively. On Fig. 2 we can see, that computational time on HybriLIT cluster is by 7-10 times lower than on JSCC cluster even though the base frequency per core of HybriLIT is just 1.4–2.4 times higher than on JSCC cluster. We can see noticeable slow down in speedup (Fig. 3) and efficiency (Fig. 4), which occurs after increasing number of computational processes beyond 28 for HybriLIT and 68 for JSCC cluster. This number of processes coincides with number of individual cores of used CPUs. To use higher number of processes, hyper threading had to be used.

One can see interesting behavior when approaching to the maximal number of physical cores, when the efficiency of calculation reaches 95% for HybriLIT cluster and about 85% for JSCC cluster. This phenomenon can be because of the hardware architecture or a special feature of our data or both of them. Anyway it will need further investigation to identify.

4 Conclusions and Discussion

We have introduced the algorithm solving TOV equations for set of EoS models, which is parallelized on the level of different models. Such algorithm is important for investigation of models of cold and dense nuclear matter. On testing set of modern realistic EoS models we have shown how it scales when using with MPI on modern CPUs. This tests allowed us also to compare two different types of computational nodes. Ones based on Intel Xeon Phi 7250 multi-core processors and ones based on Xeon E5-2695v3 processors. We could see, that efficiency of Intel Xeon Phi is higher when using more than 28 cores, which is number of physical processors provided by $2\times$ Xeon E5-2695v3 which we had in our testing system. Nevertheless the computational times were much shorter when using Xeon E5. The real advantage of using Xeon Phi would be when running much larger number of MPI processes in parallel (up to 1088 for our setting of $4\times$ Xeon Phi).

Note, that the difference in running times for both systems for the same number of threads cannot be explained only by difference in frequencies. The performance of the Intel Xeon Phi 7250 is 10~11 times smaller than of Xeon E5-2695v3 when using the same amount of threads while Xeon E5-2695v3 has only about 1.64 times higher frequency. Such dramatic difference of the performances can be explained by the features of the architecture, and first of all, in memory access organization. For more details see the extensive performance comparison of the modern Intel architectures [14].

For problems with several thousands of models or with larger number of free parameters, also calculation of single model as well as transfer of data will start to play role. In this case we will consider to use different approach to load files (move from text to binary files) and to transfer data – probably send just pointers to where the data is stored and how to obtain it. This will reduce communication costs, which are for small problems negligible. Also we will consider CUDA as even calculating solution to TOV equations of single EoS model is embarrassingly parallel. We have not used MPI to integrate TOV equations for single EoS model in parallel as the communication costs would be much higher compared to the calculation itself, but using CUDA we are not be bound by waiting for MPI signals and therefore we can get some performance boost.

Acknowledgements

The research was carried out under financial support of the Russian Science Foundation (project No. 17-12-01427).

References

- [1] Fridolin Weber. *Pulsars as Astrophysical Laboratories for Nuclear and Particle Physics* (CRC Press, Boca Raton, 1999), 696 pp.
- [2] R. C. Tolman, Phys. Rev. **55**(4), pp. 364–373 (1939)
- [3] J. R. Oppenheimer and G. M. Volkoff, Phys. Rev. **55**(4), pp. 374–381 (1939)
- [4] D. Blaschke, H. Grigorian, and D. N. Voskresensky, Phys. Rev. C **88**, 065805 (2013)
- [5] S. Typel, G. Röpke, T. Klähn, D. Blaschke, and H. H. Wolter, Phys. Rev. C **81**, 015803 (2010)
- [6] K. A. Maslov, E. E. Kolomeitsev and D. N. Voskresensky, Phys. Rev. C **92**(5), 052801 (2015)
- [7] K. A. Maslov, E. E. Kolomeitsev and D. N. Voskresensky, Nucl. Phys. A **950**(64) (2016)
- [8] R. H. Merson, Proc. Symp. Data Processing, Weapons Res. Establ. Salisbury, Salisbury (1957) pp. 110–125
- [9] J. Christiansen, Numer. Math. **14**, pp. 317–324 (1970)
- [10] *MPI: A Message-Passing Interface Standard Version 3.1.* (2015), <http://mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf>
- [11] D. Alvarez-Castillo, A. Ayriyan, S. Benic, D. Blaschke, H. Grigorian, and S. Typel. European Physical Journal A **52**:69, (2016), doi:10.1140/epja/i2016-16069-2
- [12] Joint Supercomputer Center of the Russian Academy of Sciences, <http://www.jscc.ru/eng/index.shtml> (14.11.2017)
- [13] Heterogeneous cluster “HybriLIT”, <http://hybrilit.jinr.ru/en/> (14.11.2017)
- [14] I. Jabbie, G. Owen, and B. Whiteley. SIAM Undergraduate Research Online (SIURO) **10**, (2017), doi:10.1137/17S015896