Lattice QCD Simulations in Parallel

Dafina Xhako[1*], Niko Hyka[2], Rudina Zeqirllari[3], Enkelejd Caca[4]

Abstract

Lattice Quantum Chromodynamics (LQCD) is the lattice discretization of QCD theory, the mathematically formulation that describes the strong interactions between elementary particles. It is based on complex numerical algorithms that comes from a similarity one by one among the path integral in statistical mechanics and the Markov chain in the Monte Carlo algorithm. Solving QCD theory with Monte Carlo simulations it has a large computational cost and often these kinds of simulations have to be done in supercomputer with high computation speed and power. In addition to the complexity of formulating the theory in lattice, the computational cost is increased by the fact that in order to be as close as possible to the continuous limit we have to do simulations in as large lattices as possible. For these purposes we bring in this paper a study that present the efficacity of FermiQCD software when it is used in parallel cores. This paper gives a very good start point for lattice QCD community that a very optimal way to win time and computational cost it is to use parallel simulations using the most appropriate software for such kind of calculations such as FermiQCD. In our paper, FermiQCD software testing was done with quenched quantum chromodynamics simulations, in SU(3) gauge calibrations. One of the main advantages of FermiQCD over other libraries is the fact that it is based on a simple object-oriented programming structure as opposed to a "procedural" design. The results shows that this software is on of the optimal parallelized soft as for now in the research field of lattice QCD community. We found out that this software has a very good scaling for number of cores up to five. The computing time of the computation decreases exponentially with the increase of the number of processors used for a node, for a fixed lattice volume. All parallel simulations are done under High Performance Computing Project for South East European countries.

Key Words: Lattice, Parallel, QCD.

DOI Number: 10.14704/nq.2022.20.8.NQ44017 **NeuroQuantology 2022; 20(8):153-159**

Introduction

Quantum Chromodynamics (QCD) is the theory that describes the behavior of strong interactions and has been the result of many years of research. QCD, as a quantum field theory and part of the standard elementary particle model, describes the interaction between quarks and gluons. Among the typical properties of quarks is the fact that they are characterized by a "color" charge and interact by means of strong interaction (color force). For short distances (high energies), QCD exhibits the features of asymptotic freedom, while for long distances (low energies), quarks and gluons are enclosed in neutral-colored groups called hadrons. In 1964 was formulated the hypothesis that hadrons are composed of other particles that make up the quark model [1],[2],[3]. In the 1970s, experiments with particle accelerators confirmed the existence of quarks, although isolated quarks were never observed in nature. Quarks are found permanently enclosed in hadrons or in neutral-colored groupings. The origin of quark confinement has not been directly revealed but a mechanism is the force that holds two quarks with color charge together. The main application of QCD theory it is lattice QCD (LQCD) [2],[3]. Also, exists other important phenomena and applications of Quantum Chromodynamics such are: the perturbative methods in gauge theories, the study of asymptotic freedom of quarks and more as are mentioned in [3].[4],[5], [8], [9, [10], [11], [12] Lattice Quantum Chromo-Dynamics (LQCD) is the formulation of the calibration field theory in a discrete space-time, proposed by Wilson (1974).

Corresponding author: Dafina Xhako
Address: [1,4] Department of Physics Engineering, Faculty of Mathematical Engineering and Physical Engineering, Polytechnic University of Tirana, Tirana, Albania. [2] Department of Diagnostics, Faculty of Medical Technical Sciences, Medical University of Tirana, Tirana, [3] Department of Physic, Faculty of Natural Sciences, University of Tirana, Tirana, 1000, Albania

E-mail: dxhako@fim.edu.al

The natural method of quantization of a network theory is that of the formalism of the track integral, according to which the system takes the form of a four-dimensional statistical model. Subject fields are treated as variables located at the network nodes, while calibration fields are placed at the connections that join the nodes. An important advantage of network formulation is the fact that the track integral can be numerically calculated through Monte Carlo simulations [6], [7], [8], [9]. Numerical calculations on the lattice with Monte Carlo methods are quite costly [13],[14], [15], [16], [17]. Meanwhile, on the other hand, the increase in precision due to theoretical improvements in techniques has been accompanied by an increase in computing power. Consequently, parallel computing techniques are a pretty good alternative to gain in time and computational cost.

Materials And Methods

Parallel calculations are performed using a special software called FermiQCD [2], [3], [4], [5] which is a collection of classes, functions and parallel algorithms for network QCD written in C ++. It contains optimized algorithms implemented using MPI (Message Paging Interface) [6] but MPI calls are hidden in the high levels of algorithms that make up FermiQCD. FermiQCD also works on a single processor. The language types (classes) at FermiQCD include complex numbers (mdp_complex), matrices (mdp_matrices), grids (mdp_lattice), gluon fields, fermions (gauge field, farm field), spreaders (propagandist farm), calibration action functions, action, farm action) etc. One of the main advantages of FermiQCD over other libraries is the fact that it is based on a simple object-oriented programming structure as opposed to a "procedural" design. Below we list briefly some of the advantages of using this software for digital QCD simulations.

The programs (codes) written in FermiQCD are easy to write, read and modify as the syntax used is closely related to the mathematical syntax used in the Quantum Field Theory articles and books.

Programs are "portable", in the sense that they can be compiled with any other ANSI C ++ compiler. Specific hardware optimizations are coded in the library and are hidden by the high level of programming.

Communications In Fermiqcd Are Based On Mpi.

Programs are easy to correct because using FermiQCD objects and algorithms does not require explicit use of markers. All memory management is performed by the objects themselves.

We first studied and tested FermiQCD software on individual local area computers for small networks up to 44 on ready-made examples included in the package. The source for its download is available on the software website (http://web2py.com/fermiqcd). To use it later you do not need installation but in advance requires the installation of Mercurial software, further instructions are given on the official website of the software. Software testing was then extended to large parallel machines such as clusters in Sofia, Bulgaria. Access to the latter was part of the calculations for an LQCD computing project by the TirLatt group (a group of lattices QCD researchers in Tirana, Albania) selected by HP-SEE (High-Performance Computing Infrastructure for South East Europe) Research Communities). A preparatory work was done with the study and familiarization with programming in parallel with MPI. Parallel compilation and execution commands were recognized and tested.Wee have used also for statistical manipulations of data taken the MATLAB [18].

Results And Discussion

The computer time of parallel cluster calculations was first studied for the number of different processors used as well as for networks of different sizes with ready-made examples of the package, in order to test its scaling for even denser networks and for the number of processors. different. The results are given tabulated in Table 1.

Table 1. The obtained values of calculation time in seconds depending on the number of processors for lattices with different volumes with FermiQCD

Number of processors (np)	The Lattice volume 84	The Lattice volume 124	The Lattice volume 164	The Lattice volume 204
1	102,928	539,974	1699.8	3875.01
2	50.1882	245,913	775,098	1949.18
3	40.6041	187,391	575,832	1331.94
4	24.0793	137,427	414.68	940,543
5	27.4751	127,294	429,739	802,563
6	24.9939	92.3678	315,519	831,246
7	24.7644	83.7491	314.14	610,789
8	12.5531	84.0548	225,209	599,266
9	19.4817	125,336	313.202	893,248
10	19,475	126,841	313,366	616,307
11	19,441	128,586	311,705	626,435
12	19.5794	66.9843	317,799	625,546
13	19.7308	66,227	313,243	626,113
14	19.7594	67.4796	314,101	620,389

We are also giving the results in graphical form below. Specifically, Figures 1 to 4 show the computational time for a number of different processors from 1 to 14 for 8 ^ 4, 12 ^ 4, 16 ^ 4, 20 ^ 4 lattices. The exponential decrease of the computation time with the increase of the number of processors for all networks is clearly visible. It is also noticed that for number of processors over 4, the calculation time is saturated.

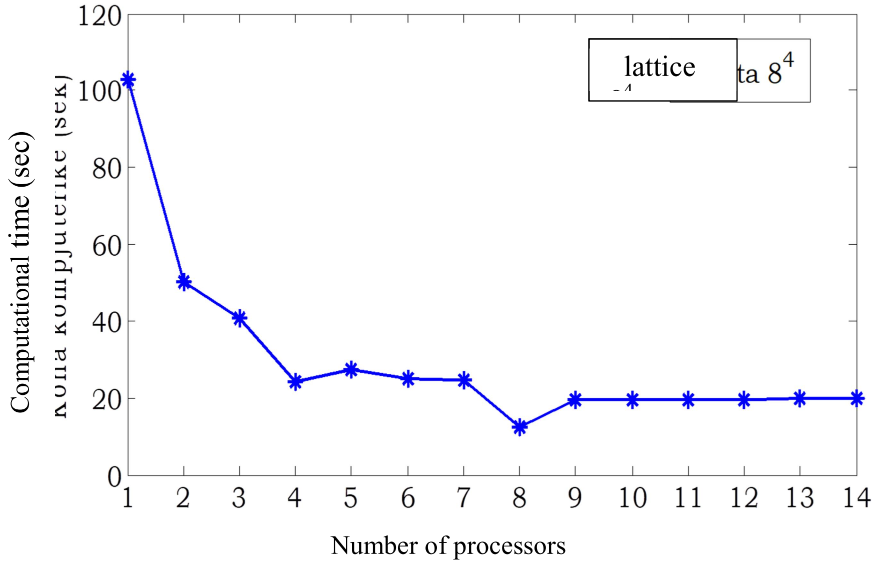

Figure 1. Dependence of computer computation time on the number of processors used during the simulation of one of the FermiQCD codes for 8 ^ 4 lattice volume

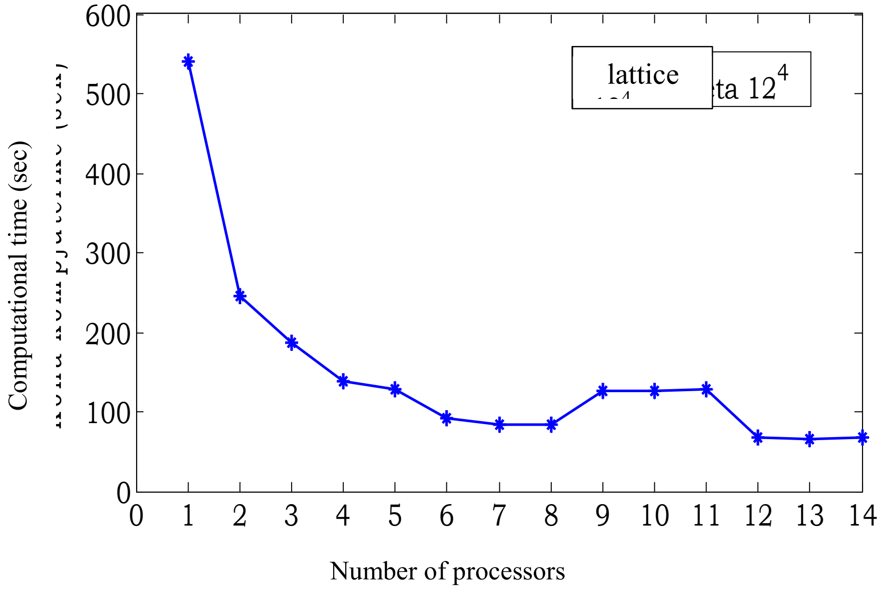

Figure 2. Dependence of computer computation time on the number of processors used, while simulating one of the FermiQCD examples for 12 ^ 4 lattice volume.

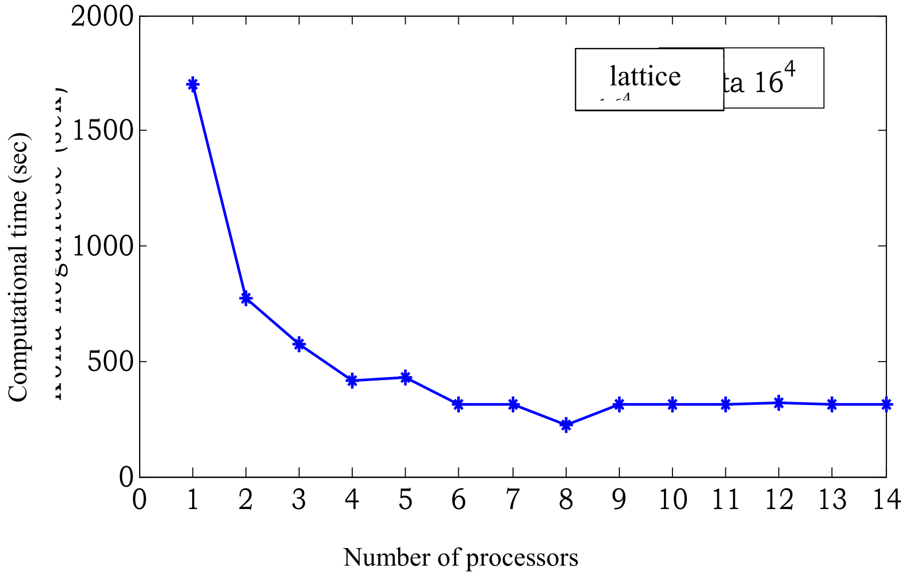

Figure 3. Dependence of computer computation time on the number of processors used, during the simulation of one of the FermiQCD examples for 16 ^ 4 lattice volume.

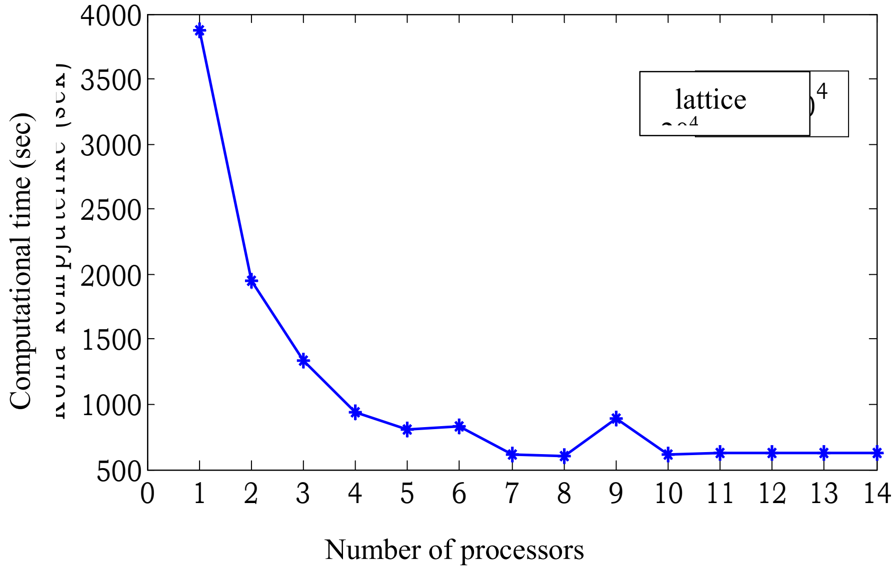

Figure 4. Dependence of computer computation time on the number of processors used during the simulation of one of the FermiQCD examples for 20 ^ 4 lattice volume.

We also studied the dependence of the computational time of the computer on the size of the network used in the simulations for a fixed number of processors, graphically presented in Figure 5.

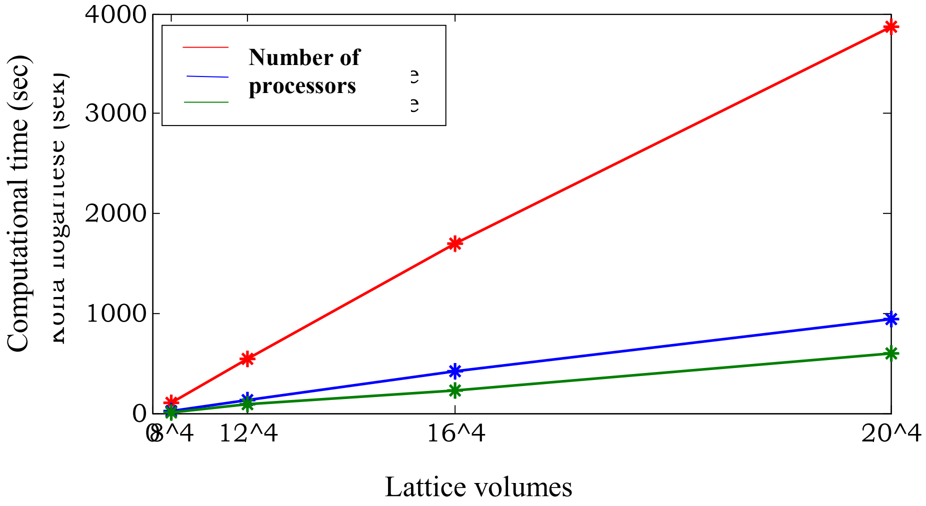

Figure 5. Dependence of computing time from lattice volume, during simulation of one of FermiQCD examples for fixed number of processors 1,4,8.

It is clear that as the network volume increases the computation time increases. Figure 5 shows some case studies for different fixed processors, such as 1,4,8. We also performed the FermiQCD acceleration and efficiency test. If T (n, 1) is the computational time of the fastest known sequential algorithm and T (n, p) is the computational time of the parallel algorithm executed on p-processors, where n is the size of the input parameters (lattice volume), then the acceleration test is determined by the results of the FermiQCD acceleration test for networks with different volumes are given in Figure 6.

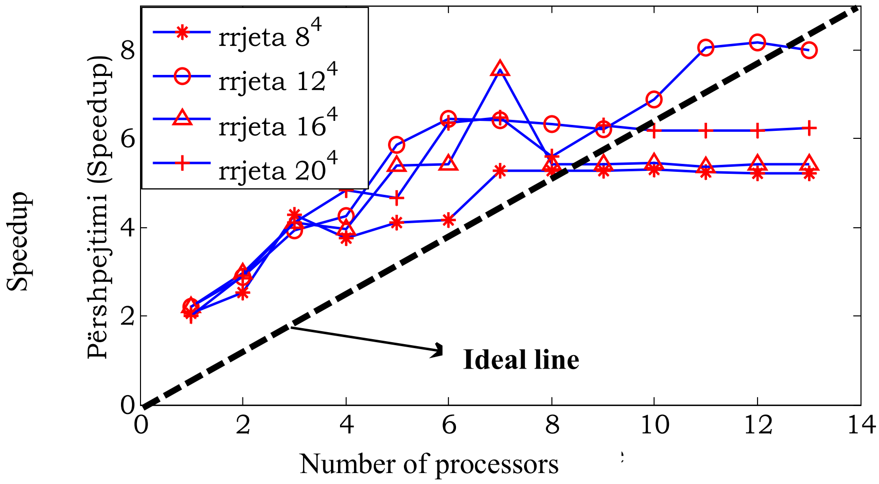

Figure 6. FermiQCD speedup test by the number of processors used in the simulation for different lattice volumes

The ideal acceleration would be obtained if S (p) = p, so if we doubled the number of processors, the execution time would be doubled. Another test that measures the performance of a parallel algorithm is test efficiency $E(p) = S(p)/p$. The results of the FermiQCD efficiency test for networks with different volumes are given in Figure 7.

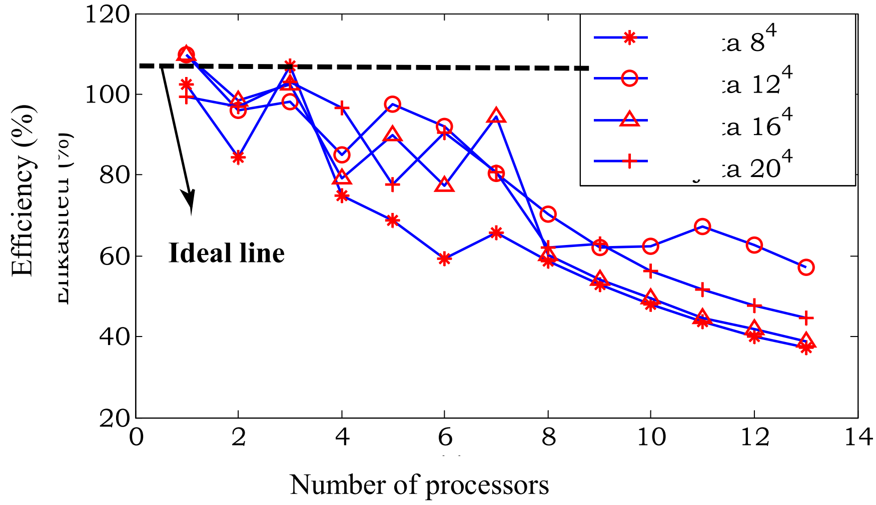

Figure 7. Efficiency test (percentage test) expressed as a percentage of FermiQCD by the number of processors used in the simulation for networks of different volumes

CONCLUSIONS

The Parallel computing techniques are quite efficient in lattice QCD calculations. In the simulations on our individual computers with very small 4 ^ 4 (rare) networks it would take about 8 hours to get Wilson's rectangular loops (r = t = 1 ... 4) for 100 configurations. Meanwhile the same parallel calculation would take a few seconds. Thus, an important direction where we need to focus on future calculations in network QCD is parallel calculations. The software chosen for our parallel calculations is one of the best software today in the field of network QCD. From his description we emphasized the simplicity but also the high level of language used. The tests performed showed that FermiQCD is optimal for parallel network QCD simulations. The software scales very well up to number of processors used np = 5. The computing time of the computation decreases exponentially with the increase of the number of processors used for a node, for a fixed network.

ACKNOWLEDGMENTS

This work is done to honor the memoriam of prof. Artan Borici who passed away from the Covid19

pandemic in April 2021. All simulations are done using one of the supercomputer clusters of the HP-SEE project (High-Performance Computing Infrastructure for South East Europe's Research Communities), the cluster in Sofia, Bulgaria (BG-HPC) and in the Department of Engineering Physics at the Faculty of Engineering Mathematics and Engineering Physics (FIMIF), Polytechnic University of Tirana, Albania.

REFERENCES

Barkai D., Creutz M., Moriarty KJM (1984): Monte Carlo study of SU (3) gauge theory on a 124 lattice, Phys. Rev. D29, 1207-1212

Creutz M. (1980): Monte Carlo study of quantized SU (2) gauge theory, Phys. Rev. D21, 2308-2315.

Di Pierro M. (2001): Parallel Programming with Matrix Distributed Processing, Computer Physics Communcations, 141, 98-148.

Di Pierro M. (2002): FermiQCD: A tool kit for parallel lattice QCD applications, Nuclear Physics B- Proceedings, Supplement, 106-107, 1034-1036

Di Pierro M., El-Khadra A., Mackenzie P., Okamoto M., Oktay M. (2004): www.fermiqcd.net, Nuclear Physics Proceedings Supplement. 129: 832-834

Kuiper R. (2008): Introduction to Parallel Computing and the Message Passing Interface, (MPI) July, Lectures

Wilson KG (1974): Confinement of Quarks, Phys. Rev. D10 2445

M.Gellamn, A schematic model of baryons and mesons, Volume 8, number 3, Physics Letters (1964)

C. Gattringer and B. Lang, Quantum Chromodynamics on the Lattice Springer, ISBN: 3642018491,343 pages (2009).

M. Beneke, Applications of QCD Int. J. Mod. Phys. A 15S1, 632-656, eConf C990809, 632-656, (2000)

K. M. Creutz and K.J.M. Moriarty, Phase transition in SU(6) lattice gauge theory, Phys. Rev. D26, 2166-2168 (1982).

K. Schilling, G.S. Bali, The Static Quark-Antiquark Potential: A ``Classical''. Experiment on the Connection Machine CM-2, Int.J. Mod.Phys.C4:1167-1177, (1993)

S. Deldar, The Potentials of the pure SU(3) representations, Phys. Rev. D62, 34509 (2000).

G. Bali, Casimir scaling of SU(3) static potentials, Phys. Rev. D62, 114503 (2000)

G. Parisi, R. Petronzio, and F. Rapuano, A measurement of the string tension near the continuum limit, Phys.Lett., B128:418, (1983)

K. Schilling and G.S. Bali, The Static potential: Lattice versus perturbation theory in a renormalon based approach, Int. J. Mod.Phys.C4:1167-1177, (1993).

S. Deldar, Potentials between static SU(3) sources in the fat-center-vortices model, Phys. Rev. D62, 34509, (2000).

A. Boriçi, QCDLAB: Designing Lattice QCD Algorithms with MATLAB, High Energy Physics - Lattice (hep-lat), arXiv:hep-lat/0610054, (2006)