PROCEEDINGS OF SCIENCE

An implementation of hybrid parallel C++ code for the four-point correlation function of various baryon-baryon systems

Hidekatsu Nemura[*][†]

Center of Computational Sciences, University of Tsukuba, Tsukuba, Ibaraki, 305-8577, Japan

E-mail: `nemura.hidekatsu.gb@u.tsukuba.ac.jp`

for HAL QCD Collaboration

We present our recent effort to develop the computational code written in C++ to calculate the four-point correlation function of various baryon-baryon (BB) systems which is a primary quantity to study the nuclear force and hyperonic nuclear forces from lattice QCD. In the recent few years, the 2+1 flavor lattice QCD calculations have been widely performed. The flavor symmetry breaking effect would be a central issue so that a lot of BB channels have to be calculated. The situation is contrast to the study of flavor symmetric BB interactions where each channel is classified into only six kinds of flavor irreducible representation. This work is also aimed at the large volume calculation of the lattice QCD for the hyperonic nuclear forces which is performed at more closer point to the physical quark mass. A hybrid parallel code is implemented by utilizing the MPI and OpenMP together with the porting it to Bridge++ which is a recently developed new C++ code set for lattice QCD calculation. The present code works on BlueGene/Q and shows better performance at hybrid parallel executions rather than the flat MPI. We also discuss how the computational time is reduced for various BB channels by diagrammatic classifications.

31st International Symposium on Lattice Field Theory - LATTICE 2013
July 29 - August 3, 2013
Mainz, Germany

[*] Speaker.

[†]

1. Introduction

Nuclear force and hyperonic nuclear force are the fundamental problems in wide range of fields in physics, e.g., particle, nuclear and astrophysics. While high-precision phenomenological nuclear potentials are available which reproduce the nucleon-nucleon (NN) scattering data at low energies as well as the deuteron properties, phenomenological description of the hyperon-nucleon (YN) and the hyperon-hyperon (YY) interactions is not well constrained due to the difficulty of scattering experiments. The precise determination of NN, YN and YY interactions provides a large impact to the studies both of hypernuclei [1] and the hyperonic matter inside the neutron stars [2].

Recently, a new approach to study the inter hadron interaction from the lattice QCD has been proposed [3]. In this approach, the interhadron potential can be obtained from lattice QCD by measuring the Nambu-Bethe-Salpeter (NBS) amplitude and the observables such as the phase shifts and the binding energies are calculated through the resultant potential [4]. This approach has been applied to various baryonic interactions [5, 6, 7, 8, 9, 10, 11, 12], and the method is recently extended to systems in inelastic channels [13, 14]. The flavor symmetry breaking is a key issue when we study the various octet-baryon-octet-baryon (B_8B_8) interactions from the $2+1$ flavor lattice QCD calculation. In the circumstances, it is desirable to calculate a large number of B_8B_8 four-point correlation functions at the same time in a single lattice QCD calculation. Therefore, an efficient approach to perform such a complicated lattice QCD calculation is crucial.

The purpose of this paper is to present our recent effort to develop the hybrid-parallel computational program written in C++ which calculates the four-point correlation function of various B_8B_8 systems. This report is organized as follows: Section 2 briefly describes how to calculate the four-point correlation function by considering the $p\Lambda$ system as a specific example. The contraction algorithm considered in this report is different from the unified contraction algorithm [15] and it has been used to calculate the ΛN and ΣN potentials [16, 17, 18]. Section 3 develops the computer program to make it work both on CPS++ [19] and on Bridge++ [20]. The Bridge++ is a recently developed new C++ code set for lattice QCD calculation. Section 4 is devoted to generalize the present contraction algorithm to various B_8B_8 systems. In Sec. 5 we discuss the hybrid parallel computation of the four-point correlation function. Sec. 6 summarizes the report.

2. Formulation

For the HAL QCD approach we calculate the normalized four-point correlation function defined by

$$R^{(J,M)}_{\alpha\beta}(\vec{r},t-t_0) = \sum_{\vec{X}} \left\langle 0 \left| B_{1,\alpha}(\vec{X}+\vec{r},t) B_{2,\beta}(\vec{X},t) \overline{\mathscr{J}^{(J,M)}_{B_3B_4}(t_0)} \right| 0 \right\rangle / \exp\{-(m_{B_1}+m_{B_2})(t-t_0)\}, \tag{2.1}$$

where the summation over $\vec{X}$ selects states with zero total momentum. The $B_{1,\alpha}(x)$ and $B_{2,\beta}(y)$ denote the interpolating fields of the baryons such as

$$\begin{array}{ll} p = \varepsilon_{abc}\left(u_a C\gamma_5 d_b\right) u_c, & n = -\varepsilon_{abc}\left(u_a C\gamma_5 d_b\right) d_c, \\ \Sigma^+ = -\varepsilon_{abc}\left(u_a C\gamma_5 s_b\right) u_c, & \Sigma^- = -\varepsilon_{abc}\left(d_a C\gamma_5 s_b\right) d_c, \\ \Sigma^0 = \frac{1}{\sqrt{2}}\left(X_u - X_d\right), & \Lambda = \frac{1}{\sqrt{6}}\left(X_u + X_d - 2X_s\right), \end{array} \tag{2.2}$$

where

$$X_u = \varepsilon_{abc}\,(d_a C\gamma_5 s_b)\,u_c, \quad X_d = \varepsilon_{abc}\,(s_a C\gamma_5 u_b)\,d_c, \quad X_s = \varepsilon_{abc}\,(u_a C\gamma_5 d_b)\,s_c, \tag{2.3}$$

and $\overline{\mathscr{J}^{(J,M)}_{B_3B_4}(t_0)} = \sum_{\alpha'\beta'} P^{(J,M)}_{\alpha'\beta'} \overline{B_{3,\alpha'}(t_0) B_{4,\beta'}(t_0)}$ is a source operator which creates B_3B_4 states with the total angular momentum J, M. This normalized four-point function can be expressed as

$$R^{(J,M)}_{\alpha\beta}(\vec{r}, t-t_0) \, = \, \sum_n A_n \sum_{\vec{X}} \left\langle 0 \left| B_{1,\alpha}(\vec{X}+\vec{r},t) B_{2,\beta}(\vec{X},t) \right| E_n \right\rangle \mathrm{e}^{-(E_n - m_{B_1} - m_{B_2})(t-t_0)}, \tag{2.4}$$

where E_n ($|E_n\rangle$) is the eigen-energy (eigen-state) of the six-quark system with the particular quantum number (i.e., J^π, M, strangeness S and isospin I), and $A_n = \sum_{\alpha'\beta'} P^{(JM)}_{\alpha'\beta'} \langle E_n | \overline{B}_{4,\beta'} \overline{B}_{3,\alpha'} | 0 \rangle$.

In what follows, we introduce a highly abbreviated notation to indicate explicitly the color, spinor and spatial subscripts. For example, we may rewrite the interpolating field of proton as

$$\begin{aligned} p_\alpha(x) &= \varepsilon(c_1,c_2,c_3)(C\gamma_5)(\alpha_1,\alpha_2)\delta(\alpha,\alpha_3)u(\xi_1)d(\xi_2)u(\xi_3), \qquad (\xi_i = x_i\alpha_i c_i) \\ &= \varepsilon(1,2,3)(C\gamma_5)(1,2)\delta(\alpha,3)u(1)d(2)u(3). \end{aligned} \tag{2.5}$$

Here, in the last equation, the numbers in the parentheses show the indices of color for $\varepsilon(\cdot)$, the indices of Dirac spinor for $(C\gamma_5)(\cdot)$ and $\delta(\cdot)$ and the indices both of color, spinor and spatial coordinate for the quark fields $u(\cdot)$, $d(\cdot)$ and $s(\cdot)$. By using the abbreviated notations, the $p\Lambda$ four-point correlator is given by

$$\begin{aligned} &\sum_{\vec{X}} \left\langle 0 \left| p_\alpha(\vec{X}+\vec{r},t)\Lambda_\beta(\vec{X},t)\overline{\mathscr{J}_{p_{\alpha'}\Lambda_{\beta'}}(t_0)} \right| 0 \right\rangle \\ &= \sum_{\vec{X}} \tfrac{1}{6}\varepsilon(1,4,2)\varepsilon(5,6,3)\varepsilon(1',4',2')\varepsilon(5',6',3')(C\gamma_5)(1,4)\delta(\alpha,2)(C\gamma_5)(1',4')\delta(\alpha',2') \\ &\quad \times\{(C\gamma_5)(5,6)\delta(\beta,3)+(C\gamma_5)(6,3)\delta(\beta,5)-2(C\gamma_5)(3,5)\delta(\beta,6)\} \\ &\quad \times\{(C\gamma_5)(5',6')\delta(\beta',3')+(C\gamma_5)(6',3')\delta(\beta',5')-2(C\gamma_5)(3',5')\delta(\beta',6')\} \\ &\quad \times\langle u(1)d(4)u(2)d(5)s(6)u(3)\overline{u}(3')\overline{s}(6')\overline{d}(5')\overline{u}(2')\overline{d}(4')\overline{u}(1')\rangle. \end{aligned} \tag{2.6}$$

The Eq. (2.6) includes implicit summations such as $\sum_{c_1,\cdots,c_6}\sum_{c'_1,\cdots,c'_6}\sum_{\alpha_1,\cdots,\alpha_6}\sum_{\alpha'_1,\cdots,\alpha'_6}$; The number of iterations for each summation is $N_c = 3$ for the color or $N_\alpha = 4$ for the Dirac spinor. Combining the iteration due to the Wick contraction, the total number of iterations for such a correlator is in a naive counting $(N_c!N_\alpha)^4 \times N_u!N_d!N_s!$, where the N_u, N_d and N_s are the numbers of u-quark, d-quark and s-quark, respectively. Clearly, the above counting is too naive though curtailment of the number of iterations is not trivial. We now explain briefly how the number of iterations reduces when we calculate the four-point correlation function of the $p\Lambda$ system [16]. It is convenient to treat separately the contributions from $\overline{X}_u$, $\overline{X}_d$ and $\overline{X}_s$ in $\overline{\Lambda}_{\beta'}$ in the source. We consider a diagrammatic classification of the Wick contraction together with employing the Fast-Fourier-Transform (FFT) to the convolution.

$$R_{\alpha\beta}(\vec{r}) = \sum_{i=1}^{6} F_i \sum_{\vec{X}} \left([p^{(i)}_\alpha](\vec{X}+\vec{r}) \times [\Lambda^{(i)}_\beta](\vec{X}) \right) = \frac{1}{L^3} \sum_{\vec{q}} \left(\sum_{i=1}^{6} F_i \left(\widetilde{[p^{(i)}_\alpha]}(\vec{q}) \times \widetilde{[\Lambda^{(i)}_\beta]}(-\vec{q}) \right) \right) \mathrm{e}^{i\vec{q}\cdot\vec{r}}, \tag{2.7}$$

where $F_i = +\frac{1}{6}\mathrm{e}^{(m_p+m_\Lambda)(t-t_0)}$ for odd i, $-\frac{1}{6}\mathrm{e}^{(m_p+m_\Lambda)(t-t_0)}$ for even i. Fig. 1 shows the diagrammatic representation of Eq. (2.6). Six diagrams in Fig. 1 correspond to the six baryon-block pairs $([p^{(1)}_\alpha] \times [\Lambda^{(1)}_\beta]), \cdots, ([p^{(6)}_\alpha] \times [\Lambda^{(6)}_\beta])$. Note that the number of diagrams is reduced by the factor 2 since the exchange between identical quarks in each baryon operator in the sink shall be taken into

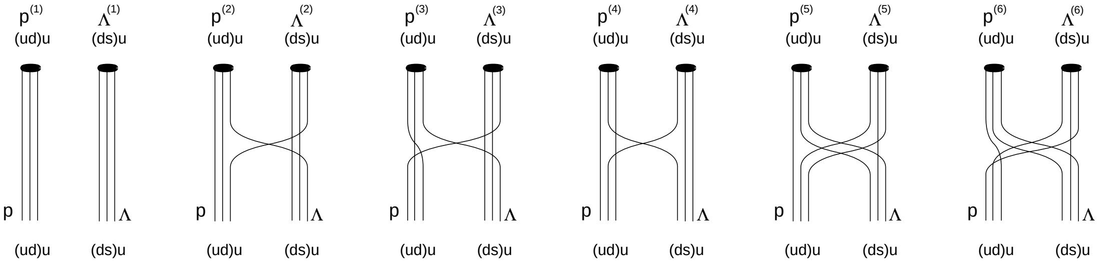

Figure 1: Diagrammatic representation of the four-point correlation function $\langle p\Lambda\overline{p\Lambda}\rangle$. The cyclic permutation for the quark fields $(ds)u \to (su)d \to (ud)s$ is taken into account in the interpolating field of Λ.

account in the construction of each baryon block $[p^{(i)}_{\alpha}]$ or $[\Lambda^{(i)}_{\beta}]$. By employing the diagrammatic classification, most of the summations can be performed prior to evaluating the FFT so that the number of iterations significantly reduces; The numbers of iteration are $\{1,9,36,144,144,36\}$ for the baryon blocks $\{([p^{(i)}_{\alpha}]\times[\Lambda^{(i)}_{\beta}]); i=1,\cdots,6\}$ when the operator $\overline{X}_u$ is considered in $\overline{\Lambda}_{\beta'}$ in the source whereas the numbers of iteration are $\{1,36,144,36,144,36\}$ ($\{1,36,144,144,36,9\}$) when the operator $\overline{X}_d$ ($\overline{X}_s$) is considered. Therefore, in order to obtain the four-point correlation function of the $p\Lambda$ system, only 370, 397 and 370 iterations should be explicitly performed when we calculate the contributions from $\overline{X}_u$, $\overline{X}_d$ and $\overline{X}_s$, respectively.

3. Implementation of C++ code together with the porting to Bridge++

The earlier versions of the C++ program to calculate the four-point correlation functions of ΛN and ΣN systems were implemented by employing the above computational technique. The code works with CPS++ [19] and it was used to calculate the ΛN and ΣN potentials [16, 17, 18]. Recently a new C++ code set for lattice QCD calculation is released which is called Bridge++ [20]. We have had a porting the present C++ program of the four-point correlation function to Bridge++ as well. The task of the porting is almost straightforward and now the program works on both Bridge++ and CPS++. In Table 1, we just show a set of correspondence list for several basic lattice parameters between CPS++ and Bridge++ which is one of few issues of the porting work.

Table 1: Correspondence list of basic lattice parameters between CPS++ and Bridge++

CPS++	Bridge++	
`GJP.Xnodes()`	`Communicator_impl::Layout::npe(0)`	the number of nodes in x-direction
`GJP.Ynodes()`	`Communicator_impl::Layout::npe(1)`	the number of nodes in y-direction
`GJP.Znodes()`	`Communicator_impl::Layout::npe(2)`	the number of nodes in z-direction
`GJP.Tnodes()`	`Communicator_impl::Layout::npe(3)`	the number of nodes in t-direction
`GJP.XnodeSites()`	`CommonParameters::Nx()`	the number of sites per node in x-direction
`GJP.YnodeSites()`	`CommonParameters::Ny()`	the number of sites per node in y-direction
`GJP.ZnodeSites()`	`CommonParameters::Nz()`	the number of sites per node in z-direction
`GJP.TnodeSites()`	`CommonParameters::Nt()`	the number of sites per node in t-direction
`GJP.XnodeCoor()`	`Communicator_impl::Layout::ipe(0)`	the MPI rank number in the x-direction
`GJP.YnodeCoor()`	`Communicator_impl::Layout::ipe(1)`	the MPI rank number in the y-direction
`GJP.ZnodeCoor()`	`Communicator_impl::Layout::ipe(2)`	the MPI rank number in the z-direction
`GJP.TnodeCoor()`	`Communicator_impl::Layout::ipe(3)`	the MPI rank number in the t-direction

4. Extension to various B_8B_8 channels

In order to study the B_8B_8 interactions on an assumption that the flavor $SU(3)$ is broken but the isospin $SU(2)$ is kept, we may have to have the following 52 four-point correlation functions. (The electromagnetic interaction is not taken into account in the present lattice calculation.)

$$\langle pn\overline{pn}\rangle, \tag{4.1}$$

$$\begin{array}{lll}
\langle p\Lambda\overline{p\Lambda}\rangle, & \langle p\Lambda\overline{\Sigma^+ n}\rangle, & \langle p\Lambda\overline{\Sigma^0 p}\rangle, \\
\langle \Sigma^+ n\overline{p\Lambda}\rangle, & \langle \Sigma^+ n\overline{\Sigma^+ n}\rangle, & \langle \Sigma^+ n\overline{\Sigma^0 p}\rangle, \\
\langle \Sigma^0 p\overline{p\Lambda}\rangle, & \langle \Sigma^0 p\overline{\Sigma^+ n}\rangle, & \langle \Sigma^0 p\overline{\Sigma^0 p}\rangle,
\end{array} \tag{4.2}$$

$$\begin{array}{llllll}
\langle \Lambda\Lambda\overline{\Lambda\Lambda}\rangle, & \langle \Lambda\Lambda\overline{p\Xi^-}\rangle, & \langle \Lambda\Lambda\overline{n\Xi^0}\rangle, & \langle \Lambda\Lambda\overline{\Sigma^+\Sigma^-}\rangle, & \langle \Lambda\Lambda\overline{\Sigma^0\Sigma^0}\rangle, & \\
\langle p\Xi^-\overline{\Lambda\Lambda}\rangle, & \langle p\Xi^-\overline{p\Xi^-}\rangle, & \langle p\Xi^-\overline{n\Xi^0}\rangle, & \langle p\Xi^-\overline{\Sigma^+\Sigma^-}\rangle, & \langle p\Xi^-\overline{\Sigma^0\Sigma^0}\rangle, & \langle p\Xi^-\overline{\Sigma^0\Lambda}\rangle, \\
\langle n\Xi^0\overline{\Lambda\Lambda}\rangle, & \langle n\Xi^0\overline{p\Xi^-}\rangle, & \langle n\Xi^0\overline{n\Xi^0}\rangle, & \langle n\Xi^0\overline{\Sigma^+\Sigma^-}\rangle, & \langle n\Xi^0\overline{\Sigma^0\Sigma^0}\rangle, & \langle n\Xi^0\overline{\Sigma^0\Lambda}\rangle, \\
\langle \Sigma^+\Sigma^-\overline{\Lambda\Lambda}\rangle, & \langle \Sigma^+\Sigma^-\overline{p\Xi^-}\rangle, & \langle \Sigma^+\Sigma^-\overline{n\Xi^0}\rangle, & \langle \Sigma^+\Sigma^-\overline{\Sigma^+\Sigma^-}\rangle, & \langle \Sigma^+\Sigma^-\overline{\Sigma^0\Sigma^0}\rangle, & \langle \Sigma^+\Sigma^-\overline{\Sigma^0\Lambda}\rangle, \\
\langle \Sigma^0\Sigma^0\overline{\Lambda\Lambda}\rangle, & \langle \Sigma^0\Sigma^0\overline{p\Xi^-}\rangle, & \langle \Sigma^0\Sigma^0\overline{n\Xi^0}\rangle, & \langle \Sigma^0\Sigma^0\overline{\Sigma^+\Sigma^-}\rangle, & \langle \Sigma^0\Sigma^0\overline{\Sigma^0\Sigma^0}\rangle, & \\
 & \langle \Sigma^0\Lambda\overline{p\Xi^-}\rangle, & \langle \Sigma^0\Lambda\overline{n\Xi^0}\rangle, & \langle \Sigma^0\Lambda\overline{\Sigma^+\Sigma^-}\rangle, & & \langle \Sigma^0\Lambda\overline{\Sigma^0\Lambda}\rangle,
\end{array} \tag{4.3}$$

$$\begin{array}{lll}
\langle \Xi^-\Lambda\overline{\Xi^-\Lambda}\rangle, & \langle \Xi^-\Lambda\overline{\Sigma^-\Xi^0}\rangle, & \langle \Xi^-\Lambda\overline{\Sigma^0\Xi^-}\rangle, \\
\langle \Sigma^-\Xi^0\overline{\Xi^-\Lambda}\rangle, & \langle \Sigma^-\Xi^0\overline{\Sigma^-\Xi^0}\rangle, & \langle \Sigma^-\Xi^0\overline{\Sigma^0\Xi^-}\rangle, \\
\langle \Sigma^0\Xi^-\overline{\Xi^-\Lambda}\rangle, & \langle \Sigma^0\Xi^-\overline{\Sigma^-\Xi^0}\rangle, & \langle \Sigma^0\Xi^-\overline{\Sigma^0\Xi^-}\rangle,
\end{array} \tag{4.4}$$

$$\langle \Xi^-\Xi^0\overline{\Xi^-\Xi^0}\rangle. \tag{4.5}$$

In order to extend the calculation for various B_8B_8 channels, we have implemented C++ program to perform both the Wick contraction and the FFT through the diagrammatic classification which are automatically done once the interpolating fields in the source and the sink (i.e., the quantum numbers of the system) are given.

5. Hybrid parallel C++ program

The message passing interface (MPI) is a message-passing standard designed for distributed memory parallel computer. In a MPI parallel computation, the communication among distributed computer systems is handled by a communicator object such as `MPI_COMM_WORLD`. Open Multi-Processing (OpenMP) is an application programming interface (API) to control the multithreading computation on the shared-memory multiprocessor. The master thread forks several slave threads when an OpenMP directive such as “`#pragma omp parallel`” appears in the program; Each thread concurrently executes the computation on the shared memory and it finally joins into the master thread at the end of the current block. The MPI and OpenMP are basically independent approaches to the parallel computation. In recent years, hybrid parallel computing on massive supercomputer such as BlueGene/Q is inevitable task to have a better computational performance.

We develop the hybrid parallel C++ program using both MPI and OpenMP to calculate the four-point correlation function for the various B_8B_8 interactions, which works on the BlueGene/Q in High Energy Accelerator Research Organization (KEK). In a hybrid parallel computer program, the function `MPI_Init_thread(int* argc, char ***argv, int required, int *provided)` is called instead of `MPI_Init(int* argc, char ***argv)`. The

Table 2: Measured elapsed time for various hybrid parallel computation of the 52 four-point correlation functions $\sum_{\vec{X}} \left\langle B_{1,\alpha}(\vec{X}+\vec{r},t)B_{2,\beta}(\vec{X},t)\overline{\mathscr{J}_{B_{3,\alpha'}B_{4,\beta'}}(0)}\right\rangle$, $(t = 0,\cdots,T-1)$, by using the 32 node of BlueGene/Q on a $L^3 \times T = 16^3 \times 32$ lattice, changing the number of MPI processes (tasks_per_node) and the number of threads (OMP_NUM_THREADS). A computational job consists of two steps; To calculate all of the single baryon blocks $[B_\alpha^{(I)}]$ together with its FFT (step-1), and to calculate the 52 four-point correlation functions by performing the summations of indices of color and spinor together with its inverse FFT (step-2).

[tasks_per_node]×[OMP_NUM_THREADS]	64×1	32×2	16×4	8×4	4×8	2×16	1×32
Step-1	00:14	00:16	00:09	00:09	00:07	00:06	00:06
Step-2	00:10	00:11	00:12	00:12	00:12	00:13	00:14
Total	00:24	00:27	00:21	00:21	00:19	00:19	00:20

third argument `required` specifies the desired level of thread support. According to the MPI standard [21], the possible values for the `required` are: `[ MPI_THREAD_SINGLE | MPI_THREAD_FUNNELED | MPI_THREAD_SERIALIZED | MPI_THREAD_MULTIPLE ]`. We take the `MPI_THREAD_MULTIPLE` together with partitioning the `MPI_COMM_WORLD` into a number of sub-communicators to perform the multiple MPI communications through the sub-communicators concurrently from each forked multithread.

In Table 2, we show a set of several elapsed times measured by using the 32 node job class of BlueGene/Q at KEK for the calculation of 52 channels of B_8B_8 four-point correlation function. The calculations are performed for a $L^3 \times T = 16^3 \times 32$ lattice with a gauge configuration provided by CP-PACS and JLQCD Collaboration [22]. In Table 2, the calculation of four-point correlation function is divided into two parts: First part is to calculate all of the single baryon blocks $[B_\alpha^{(I)}]$, $(B = p,n,\Sigma^+,\Sigma^0,\Sigma^-,\Xi^0,\Xi^-,\Lambda)$ together with its FFT (step-1). Each baryon block is shared if the same diagram appears through the 52 channels of B_8B_8 four-point correlation function. Second part is to calculate the 52 four-point correlation functions $\sum_{\vec{X}} \left\langle B_{1,\alpha}(\vec{X}+\vec{r},t)B_{2,\beta}(\vec{X},t)\overline{\mathscr{J}_{B_{3,\alpha'}B_{4,\beta'}}(0)}\right\rangle$, $(t = 0,\cdots,T-1)$ from the baryon blocks by performing the summations of indices of color and spinor together with its inverse FFT (step-2). Elapsed times are measured for various combinations of the number of MPI processes (`tasks_per_node`) and the number of threads (`OMP_NUM_THREADS`). The elapsed time indicated by “64 × 1” is due to so-called flat-MPI calculation. Sometimes the problem that any hybrid parallel executions are not faster than the flat-MPI calculation is issued in hybrid parallel computation. This is not the case and the present program exhibits almost stable performances for various combinations of the number of MPI processes and the number of threads.

6. Summary

In this paper, we present a recent effort to develop the computational code written in C++ that calculates the four-point correlation function of various B_8B_8 systems. The hybrid parallel executions of the lattice size $16^3 \times 32$ show reasonable performances at various combinations of the number of MPI processes and the number of threads. This would be an advantageous point to perform the future large scale computation of various B_8B_8 potentials at the physical quark mass point.

Acknowledgments

The author would like to thank CP-PACS/JLQCD collaborations and ILDG/JLDG [23] for allowing us to access the full QCD gauge configurations, and developers of Bridge++ [20]. The author also thank maintainers of `CPS++` [19]. Calculations in this report have been performed by using the Blue Gene/Q computer under the "Large scale simulation program" at KEK (No. 12-11). This research was supported in part by Strategic Program for Innovative Research (SPIRE), the MEXT Grant-in-Aid, Scientific Research on Innovative Areas (No. 25105505).

References

[1] H. Nemura, Y. Akaishi and Y. Suzuki, Phys. Rev. Lett. **89**, 142504 (2002) [arXiv:nucl-th/0203013].

[2] See e.g. J. Schaffner-Bielich, Nucl. Phys. A **804**, 309 (2008) [arXiv:0801.3791 [astro-ph]].

[3] N. Ishii, S. Aoki, T. Hatsuda, Phys. Rev. Lett. **99**, 022001 (2007).

[4] S. Aoki, T. Hatsuda and N. Ishii, Prog. Theor. Phys. **123** (2010) 89.

[5] K. Murano, N. Ishii, S. Aoki and T. Hatsuda, Prog. Theor. Phys. **125** (2011) 1225.

[6] H.Nemura, N.Ishii, S.Aoki and T.Hatsuda, Phys. Lett. B **673**, 136 (2009).

[7] T. Inoue *et al.* [HAL QCD collaboration], Prog. Theor. Phys. **124** (2010) 591.

[8] T. Inoue *et al.* [HAL QCD Collaboration], Phys. Rev. Lett. **106** (2011) 162002.

[9] T. Doi *et al.* [HAL QCD Collaboration], Prog. Theor. Phys. **127**, 723 (2012) [arXiv:1106.2276 [hep-lat]].

[10] T. Inoue *et al.* [HAL QCD Collaboration], Nucl. Phys. A **881**, 28 (2012) [arXiv:1112.5926 [hep-lat]].

[11] N. Ishii *et al.* [HAL QCD Collaboration], Phys. Lett. B **712**, 437 (2012) [arXiv:1203.3642 [hep-lat]].

[12] T. Inoue *et al.* [HAL QCD Collaboration], Phys. Rev. Lett. **111**, 112503 (2013) [arXiv:1307.0299 [hep-lat]].

[13] S. Aoki *et al.* [HAL QCD Collaboration], Proc. Jpn. Acad. B **87**, 509 (2011) [arXiv:1106.2281 [hep-lat]].

[14] S. Aoki, B. Charron, T. Doi, T. Hatsuda, T. Inoue and N. Ishii, Phys. Rev. D **87**, no. 3, 034512 (2013) [arXiv:1212.4896 [hep-lat]].

[15] T. Doi and M. G. Endres, Comput. Phys. Commun. **184**, 117 (2013) [arXiv:1205.0585 [hep-lat]].

[16] H. Nemura, N. Ishii, S. Aoki and T. Hatsuda [PACS-CS Collaboration], PoS **LATTICE2008**, 156 (2008) [arXiv:0902.1251 [hep-lat]].

[17] H. Nemura [HAL QCD Collaboration and PACS-CS Collaboration], PoS **LAT2009**, 152 (2009) [arXiv:1005.5352 [hep-lat]].

[18] H. Nemura [for HAL QCD Collaboration], PoS LATTICE **2011**, 167 (2011) [arXiv:1203.3320 [hep-lat]].

[19] Columbia Physics System (CPS), `http://qcdoc.phys.columbia.edu/cps.html`.

[20] Lattice QCD code Bridge++, `http://bridge.kek.jp/Lattice-code/index_e.html`.

[21] Message Passing Interface Forum, `http://www.mpi-forum.org/`.

[22] T. Ishikawa *et al.* [CP-PACS/JLQCD Collaboration], Phys. Rev. D **78** (2008) 011502(R).

[23] See `http://www.lqcd.org/ildg` and `http://www.jldg.org`