Research and development of the evolving architecture for beyond the Standard Model

Kihyeon Cho, Jangho Kim and Junghyun Kim
National Institute of Supercomputing and Networking, Korea Institute of Science and Technology Information, Daejeon 34141, South Korea
Email: cho@kisti.re.kr

Abstract. The Standard Model (SM) has been successfully validated with the discovery of Higgs boson. However, the model is not yet fully regarded as a complete description. There are efforts to develop phenomenological models that are collectively termed beyond the standard model (BSM). The BSM requires several orders of magnitude more simulations compared with those required for the Higgs boson events. On the other hand, particle physics research involves major investments in hardware coupled with large-scale theoretical and computational efforts along with experiments. These fields include simulation toolkits based on an evolving computing architecture. Using the simulation toolkits, we study particle physics beyond the standard model. Here, we describe the state of this research and development effort for evolving computing architecture of high throughput computing (HTC) and graphic processing units (GPUs) for searching beyond the standard model.

1. Introduction
Research in the 21st century is a data-centric analysis of scientific experiments, unifying experiments, theory, and computation. Data-intensive science unifies theory, experiment, and numerical simulations by using exploration tools that link a network of scientists with their datasets [1]. Results are analyzed by using a shared computing infrastructure. Thousands of years ago science focused on experiments to describe natural phenomena [2]. In the last few hundreds of years, science became more theoretical [2]. In the last few decades, science has become more computational, focusing on simulations. Today, science can be described as more data-intensive in nature, requiring a combination of experiments, theory, and computing [3].

Even if the standard model remains successful owing to the discovery of Higgs bosons at the large hadron collider (LHC) [4,5], there still exist efforts to develop phenomenological BSM models. In figure 1, we schematically show a unified physics-simulation-computing research model for searching for new BSM physics based on an evolving computing architecture. This is not simply a collection of physics, simulation, and computing methods, but a fusion of research for achieving a more efficient research process. The first step is to develop simulation toolkits for meeting the evolving computing architecture. The second step is to develop simulation toolkits for generating several orders of magnitude more simulation events of the new BSM physics compared with those required for the Higgs bosons. The last step is using these toolkits for searching for the new BSM physics.
2. Evolving architecture for the BSM physics

2.1. Beyond the standard model

The discovery of Higgs bosons [4,5] validated the standard model. However, the model is not yet fully regarded as a complete description. What is the next step? What is the unification force? Why are there masses? What is the origin of dark mass and dark energy? What causes matter and anti-matter asymmetry?

The vision of the future of particle physics has been revised by the community [6]. Five scientific issues arose from a community-wide study: 1) the Higgs boson, 2) the mass of neutrino, 3) the dark matter, 4) the dark energy and 5) the unknown. Except for the Higgs boson, the remaining four issues correspond to the BSM physics. As shown in figure 2 [7], the production cross-section of the Higgs boson is $10^{-3} \sim 10^{-1} nb$. However, the cross-section of one possible new physics event is 1,000 times smaller than that of the Higgs boson, which is equivalent to being 1 trillion times smaller than the cross-sections of the Standard Model events. To meet this required number of simulation events, there is a need to develop a novel simulation toolkit.

Figure 2. The proton-(anti) proton cross-sections, showing the required simulation numbers compared with those of the models [7].

---

Figure 1. Schematic of the evolving computing architecture for the BSM physics.
2.2. Simulation

Simulations are an important addition to experiments. Most of the applications running on the world wide LHC computing grid (WLCG) are not reconstructions, but simulations. During Run 1 at the LHC experiment, the LHC simulation generates on the order of $10^{10}$ events by using 250,000 cores. As a challenge to high-luminosity LHC, we need at least five times more computing power using the current budget. Therefore, there is a need to develop simulation toolkits based on the evolving computing architecture.

Figure 3 shows the schematic of high-energy physics simulations as an example. The new physics model provides Feynman’s rules of calculation. It is produced by MadGraph [8], which is followed by process creation. PYTHIA [9] generates the events, which is followed by Geant4 [10] for detector simulation. Finally, MadAnalysis [11] and ROOT [12] are used for analysis.

![Figure 3. The flowchart of simulation toolkits.](image)

The vision for high-energy physics (HEP) simulation is to have massively parallelized particles and to comply with different architectures such as GPUs and multi-integrated chips (MICs). We also need to draw the community’s interest toward the collateral effort.

2.3. Computing

The increasing computational needs for BSM physics require full advantage of cost-effective computing solutions. Rapidly evolving computing architecture and increasing data volumes also require effective crosscutting solutions [6].

The characteristics of evolving computing architecture imply that servers control the GPU and MIC. The GPU and MIC share memory. It is a heterogeneous platform. Overview of key components of the evolving computing architecture for a simulation is not only computing engines but also algorithms of the underlying physics and environment. Using this, we study high-performance simulations that are optimal algorithms for coprocessors such as GPUs and many cores, solving problems for sampling with composition in Geant4.

In this paper, we show an example of GPU machines as the evolving computing architecture. As the simulation toolkits, we apply the finite volume effects and non-perturbative renormalization (NPR) methods of lattice quantum chromodynamics (QCD). Using these tools, we study the BSM physics.
3. Results
We use GPU machines as the evolving computing architecture. Since lattice QCD is a very good method for application to evolving computing architecture and simulation methods, we used it for the finite volume effects and NPR method. Using these tools, we studied the BSM physics. Table 1 summarizes the computing architecture for the BSM physics for this paper.

Table 1. The computing architecture for the BSM physics for this analysis.

| Computing | Simulation Tool | BSM               |
|-----------|----------------|-------------------|
| GPU       | Finite volume effects of $B_K$ in lattice QCD | BSM parameter $B_K$ |
| GPU       | NPR for $B_K$ in lattice QCD                | BSM parameter $B_K$ |

3.1. Finite volume effects on $B_K$
In the kaon system, the charge-parity ($CP$) violation parameter $\epsilon_K$ is very well-known from experiments and if the SM contribution to $\epsilon_K$ can be sufficiently accurately determined, the difference between experiments and SM theory can powerfully constrain the BSM theories [13]. Hence, the contributions to $\epsilon_K$ have to be very accurately calculated from the SM theory by using lattice simulations.

Lattice simulations are performed on the finite volume lattices. Hence, the effect of finite volume must be considered for exactly estimating the $\epsilon_K$ error. We calculate the finite volume correction from the $\mathcal{O}(\mathcal{O})$ Stueckelberg perturbation theory [14]. These correction terms are as follows:

$$\delta_1^F(M^2) = 4 \sum_{n \neq 0} \frac{K_1(|n|ML)}{|n|},$$
$$\delta_3^F(M^2) = 2 \sum_{n \neq 0} K_0(|n|ML),$$

where $K_0$ and $K_1$ are modified Bessel functions of the second kind, $M$ is the pion mass, $L$ is the box size in the spatial direction, $n = (n_1, n_2, n_3, n_4)$ and $|n| = \sqrt{n_1^2 + n_2^2 + n_3^2 + \left(\frac{L_T}{L} n_4\right)^2}$. $L_T$ is the Euclidean time direction box size.

To calculate these correction terms, we must truncate the sum over $n$. We use the following criterion with $\epsilon = 1.0 \times 10^{-14}$ (double precision):

$$[4\pi |n|^2] K_1(|n|ML) \geq \epsilon \times [6K_1(ML)]$$
$$[4\pi |n|^2] K_0(|n|ML) \geq \epsilon \times [6K_0(ML)]$$

Note that we assume $L_T \gg L$. Hence, $6K_1(ML)$ and $6K_0(ML)$ are the values of $\delta_1^F$ and $\delta_3^F$ at $|n| = 1$ because there are 6 cases: $n = (\pm 1, 0, 0, 0), (0, \pm 1, 0, 0), (0, 0, \pm 1, 0)$. $4\pi |n|^2$ is the density of vector $n$. To find the maximum of $|n|$, we use the vector $n = (i, 0, 0, 0)$ with the above criterion and increment the integer $i$ until the vector $n$ fails to satisfy the criterion. Then, we define the spatial direction radius $r_s$ as the maximum of $i$ and temporal direction radius $r_t$ as $r_t \equiv r_s \times \left(\frac{L}{L_T}\right)$. Hence, the
number of \(|n|\) satisfying the above criterion is \(n_{\text{max}} \equiv (2r_\ell + 1)^3 \times (2r_\ell + 1)\). When the pion mass \(M\) is small, the value of \(n_{\text{max}}\) becomes large. We have to calculate Bessel functions for every vector \(n\) that satisfies \(|n| < n_{\text{max}}\) and sum over them for the same mass. This calculation requires a huge computing power. If we use a single core central processing unit (CPU) (Intel i7 920), the calculation will take \(~5\) days per dataset and \(~2\) months for 9 datasets that we have. Hence, it is very difficult to calculate the finite volume correction by using a CPU machine. Therefore, we are using a GPU machine to accelerate the code, and we are using the compute unified device architecture (CUDA) library to control the NVIDIA GPU. Many threads and blocks in GPU make possible simultaneous calculation of the same operation for different data. In the finite volume effect calculation, Bessel functions have to be calculated for the various vectors \(n\) with fixed pion masses, and the calculation results are summed over. Figure 4 explains the simultaneous calculation of Bessel functions.

In figure 4, the x-axis is \(n_{\text{max}} = (2r_\ell + 1)^3 \times (2r_\ell + 1) + \alpha\) for a fixed pion mass. Here, \(\alpha\) is a dummy quantity. In the CUDA programming, the size of a memory array is preferred to be \(2^i\), where \(i\) is some integer. Hence, we add a dummy to \(n_{\text{max}}\) to make the array size \(2^i\). If the pion mass is small, \(n_{\text{max}}\) becomes large. We calculate the Bessel functions with \(n\) vectors included in the same colored region in figure 4, simultaneously using many threads and blocks. After that, we accumulate them by using a multi-thread.

We use NVIDIA GTX 480 GPU to compute the finite volume effect; this computation takes \(1\) h and \(43\) min for one dataset and \(1\) day for all datasets. The GPU code is about \(128.6\) times faster than the CPU code. The performances of the CPU and GPU are compared in table 2. Here, the theoretical peak double precision performance of GTX 480 is \(168\) GFLOPS.

![Figure 4](image-url)
The average of the shortest paths connecting the gauge links is calculated with the lattice results. To compare the lattice calculation results with experiments, the lattice results should be converted into continuous outcomes by using the matching factor. The most dominant error of calculation results with experiments, the lattice results should be converted into continuous outcomes by using the matching factor. The most dominant error of (2015) obtained by using one-loop perturbation theory. The NPR method can reduce this error to the level of 2%.

Consequently, CUDA programming enables us to modify and check the validity of our analysis code. In this analysis, we optimized the finite volume effect code by 38% of theoretical peak performance of GTX 480. Using this tool, we can obtain the finite volume correction at NLO in SU(2) staggered chiral perturbation theory in double precision without approximating the BSM parameter $B_K$.

### 3.2. NPR method to calculate the matching factor of $B_K$

The CP violation parameter $B_K$ can be calculated from lattice simulations. To compare the lattice calculation results with experiments, the lattice results should be converted into continuous outcomes by using the matching factor. The most dominant error ($\approx 4.4\%$) of $B_K$ comes from the matching factor obtained by using one-loop perturbation theory. The NPR method can reduce this error to the level of 2%.

To calculate the matching factor of $B_K$ by using the NPR method, we need to calculate the Green’s function of four-fermion operators relevant to $B_K$. There are two types of color contractions: one-color and two color staggered four-fermion operators.

The one-color staggered four-fermion operator is $\Theta^{a1}(y) = \sum_{A,B} \sum_{c_1,c_2} \sum_{c_3,c_4} \left[ \tilde{c}_1(y_A) \left( y_{S_1} \otimes \xi_{F_1} \right)_{AB} c_2(y_B) \right] \left[ \tilde{c}_3(y_C) \left( y_{S_2} \otimes \xi_{F_2} \right)_{CD} c_4(y_D) \right] U_{AB;C,C}(y) U_{BC;C}(y).

Further, the two-color staggered four-fermion operator is $\Theta^{a2}(y) = \sum_{A,B} \sum_{c_1,c_2} \sum_{c_3,c_4} \left[ \tilde{c}_1(y_A) \left( y_{S_1} \otimes \xi_{F_1} \right)_{AB} c_2(y_B) \right] \left[ \tilde{c}_3(y_C) \left( y_{S_2} \otimes \xi_{F_2} \right)_{CD} c_4(y_D) \right] U_{AB;C,C}(y) U_{BC;C}(y).

Here, $\alpha = \left( y_{S_1} \otimes \xi_{F_1} \right) \left( y_{S_2} \otimes \xi_{F_2} \right)$ is the operator index, $c_i$ are the color indices, $y$ represents the hypercube coordinate with the lattice spacing $2a$. The indices $A, B, C$, and $D$ are hyper-cubic vectors. For example, $A = (1, 1, 0, 0)$. We use the notation of $y_A = 2y + A$. The gauge link $U_{AB;C,C}(y)$ is an average of the shortest paths connecting $y_A$ and $y_B$. $y_S$ represents the spin and $\xi_F$ the taste. $\tilde{c}(y_A)$ and $\tilde{c}(y_B)$ are staggered quark fields. The definition of $\left( y_{S_1} \otimes \xi_{F_1} \right)_{AB}$ is as follows:

$$\left( y_{S} \otimes \xi_{F} \right)_{AB} \equiv \frac{1}{4} \text{tr} \left[ y_A y_S y_B y_F \right],$$

where $y_A = y_A^1 y_A^2 y_A^3 y_A^4$.
To obtain the matching factor by using the NPR method, we need to calculate the Green’s function in the momentum space. We define the reduced momentum $\bar{p}$ as follows:

$$\bar{p} \in \left( -\frac{\pi}{a}, \frac{\pi}{a} \right)^4, \quad \bar{p} = \bar{p} + \pi_A,$$

where $\pi_A = \frac{\pi}{a} A$.

The one-color Green’s function in the momentum space for a specific gauge ensemble is as follows:

$$H_{\ell_1 c_1 \ell_2 c_2 \ell_3 c_3 \ell_4} (\bar{p}_1 + \pi_A, \bar{p}_2 + \pi_B, \bar{p}_3 + \pi_C, \bar{p}_4 + \pi_D; \ell, m_1, m_2, m_3, m_4) = \sum_{E_\ell} (2a)^4 \sum_y e^{-i\bar{p} y} \times \left[ a^4 \sum_{x_1} e^{(\bar{p}_1 + \pi_A) x_1} S_{\ell_1 c_1 \ell_1' c_1'} (x_1, y_E, m_1) \right] \left[ \frac{(y_{S_1} \otimes \xi_{\ell_1})_{EF}}{\gamma} \right] \left[ a^4 \sum_{x_2} S_{\ell_2 c_2 \ell_2' c_2'} (y_F, x_2, m_2) e^{-(\bar{p}_2 + \pi_B) x_2} \right] \times \left[ U_{\ell, GH} \right]_{c_1' c_1} \left[ U_{\ell, FG} \right]_{c_2' c_2},$$

where $N_{\text{conf}}$ is the number of gauge configurations and $m_1$ is the quark mass. We set $m_1 = m_2 = m_3 = m_4 = m$ in our simulations.

The one-color Green’s function calculation occupies 97% of the running time of this program. Hence, we convert this part into the GPU code by using CUDA. The performances of CPU and GPU are compared in table 3.

### Table 3. Comparison between CPU and GPU.

| Program                  | CPU  | GPU  | Optimization | CPU vs. GPU |
|--------------------------|------|------|--------------|-------------|
| Specification            | GFLOPS | VGA (peak performance in double precision) | GFLOPS | Optimization | GPU |
| Non-perturbative         | Core i7-4820K | 1.13 | GTX 480 (168 GFLOPS) | 66.6 | 40% | 58.9 |
| renormalization          |      |      | GTX 580 (198 GFLOPS) | 76.2 |     | 67.2 |
|                          |      |      | GTX Titan Black (1707 GFLOPS) | 113.4 |     | 100.3 |

The calculation using the GPU code is 58.9 times faster than that using the CPU code. When the GPU code ran on the GTX Titan Black, it was about 100 times faster than the CPU code. When we used 16 CPUs for a $20 \times 3$ gauge ensemble, it took 5.6 days for one gauge configuration. On the other hand, when we used 16 GPUs (GTX 580), the calculation took 2 h. For the data analysis, we need the NPR data for 5 quark masses, 10 external momenta, and 30 gauge ensembles at least. Hence, if we use 16 CPUs, it will take 23 years as follows:

$$5 \times 10 \times 30 \times 5.6 = 8400 \text{ days} = 23 \text{ years}.$$

However, when we use 16 GPUs, the calculation takes only 125 days. In our results, the NPR code is optimized for GTX 480 graphic card. We obtain the 40% peak performance of GTX 480.

Using this tool, we can reduce the systematic error of $B_K$ to the level of 2%. The exact calculation of $B_K$ can provide the constraints on the BSM physics.
4. Conclusions
Physics goes BSM. Computing needs solutions for the evolving architecture. To fill the gap between physics and computing, we need to focus on simulation R&D. Based on the evolving computing architecture of GPU, we study the simulation toolkit and algorithms of lattice QCD. Then, by using these tools, we study the BSM physics.

Acknowledgement
The authors would like to thank Soon Yung Jun and Weonjong Lee. Jangho Kim was supported by a Young Scientists Fellowship through the National Research Council of Science & Technology (NST) of Korea.

References
[1]  Lin S C and Yen E 2009 J. Korean Phys. Soc. 55 2035
[2]  Hey T 2006 Proc. 20th CODATA Conf. (Beijing)
[3]  Cho K 2011 Comput. Phys. Commun. 182 1756
[4]  CMS collaboration 2012 Phys. Lett. B 716 30
[5]  ATLAS collaboration 2012 Phys. Lett. B 716 1
[6]  Particle Physics Project Prioritization Panel (P5) 2014, Building for Discovery
[7]  Stirling J, see the plot in http://www.hep.ph.ic.ac.uk/~wstirlin/plots/plots.html
[8]  Alwall J et al Preprint arXiv:1405.0301 [hep-ph]
[9]  Sjostrand T et al Preprint arXiv: 1410.3012 [hep-ph]
[10] Agostinelli S et al 2003 Nucl. Instr. Meth. Phys. Res. Sec. A: Accelerators, Spectrometers, Detectors and Associated Equipment 506 250
[11] Eric C, Fuks B and Serret G 2013 Comput. Phys. Commun. 184 222
[12] Antoncheva I et al 2009 Comput. Phys. Commun. 180 2499
[13] Bae T et al 2013 PoS(LATTICE2013) 476
[14] Kim J et al 2012 Phys. Rev. D 83 117501
[15] Bae T et al 2012 Phys. Rev. Lett. 109 041601
[16] Jeong H et al 2014 PoS(LATTICE2014) 286