Vectorized approach for faster image reconstruction for Muon Tomography

R. Sehgal[1,3],* R. Dey[3], S. P. Behera[1], P. K. Netrakanti[1], D. K. Mishtra[1], V. Jha[1,3], and L. M. Pant[2,3]
[1]*Nuclear Physics Division, Bhabha Atomic Research Centre, Mumbai - 400085, INDIA*
[2]*Technical Physics Division, Bhabha Atomic Research Centre, Mumbai - 400085, INDIA and*
[3]*Homi Bhabha National Institute, Anushakti nagar, Mumbai - 400094, INDIA*

Introduction

Muon tomography has proven to be a promising technique to scan the cargo containers and to detect the presence of high Z materials. It makes use of naturally available cosmic muons to do the scanning and image reconstruction. The existing reconstruction algorithms need to process a considerably high number of muon tracks, which makes the image reconstruction process a bit more computation-intensive. Iterative reconstruction algorithms need even more computation power, as these tracks needs to be processed multiple times before they finally converge. In this work, we will present a new way to process multiple muon tracks parallelly using the concept of vectorization. To demonstrate its capability, a vectorized version of the Point of Closest Approach (PoCA) algorithm has been implemented. The vectorized version is compared with the traditional scalar version of the algorithm, and timing performance are compared on the data generated using Geant4 [1] simulation.

Vectorization to exploit the hidden parallelism

Traditional programming is done using a Single Instruction stream Single Data stream (SISD) architecture, where the control unit applies a single instruction on single data at a time. The present CPU architecture has the capability where a Single instruction set can be applied on multiple data stream (SIMD)[2]. As computing technology evolves, different types of SIMD instruction sets are provided by different CPUs. SSE and AVX are the two most commonly available SIMD instruction sets. SSE provides a SIMD instruction set for XMM registers, which has a size of 128 bits (16 bytes), hence it is capable of processing 2 double-precision numbers of 64 bits, or four single-precision numbers of 32 bits. Similarly, AVX provides an instruction set for YMM registers, which is twice as long as the XMM register. By vectorizing the code, the multiple data elements can be processed in a single instruction cycle and hence time required to do the calculation can be reduced. In the present work, we have vectorized the PoCA reconstruction algorithm to calculate multiple PoCA points simultaneously, which results in reduced image reconstruction time.

Refactor scalar code for SIMD architecture

The basic constructs and data structures provided by any programming languages focuses on allowing the user to write any logic and hence Turing complete. With the advancement in the compiler design, the compiler tries to auto-vectorize the code, but the performance may vary, as the compiler does not know how to vectorize the full algorithm. Hence, code refactoring needs to be done, where the algorithms are explicitly vectorized. There are three ways to do SIMD programming: (i) Inline Assembly (ii) Using compiler intrinsic functions (iii) Using an external SIMD library, that provides API, to utilize the vector architecture. To implement the vectorized version of PoCA, we have used the third option mentioned above and Vc[3] library is

*Electronic address: `rsehgal@barc.gov.in`

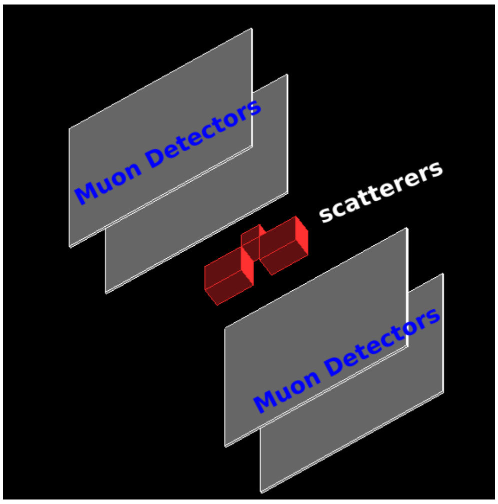

FIG. 1: Simulation setup used

chosen, as it provides the simple-to-use API to fully utilize the hardware's SIMD capabilities.

Results

The Results of vectorization are shown using the simulation setup shown in Fig. 1, where we have two muon detectors above and two muon detectors below the scatterers. These are used to get the incoming and outgoing muon tracks. A total of 10^5 reconstructed muon tracks are processed and PoCA point calculation is done using scalar as well as vectorized PoCA algorithm. Figure 2 shows the comparison of execution time between the scalar and the vectorized version, using the single and the double precision calculation, for SSE4.2, AVX and AVX2 instruction set architecture. It is clearly visible that vectorized version takes much less time as compared to scalar version. The effect of register size can also be clearly seen by comparing the timing of vectorized code while going from SSE4.2 to AVX. Going from AVX to AVX2 doesn't show much difference in timing of vectorized code, as register size remains the same. AVX2 provides some memory optimization, which results in little more performance gain.

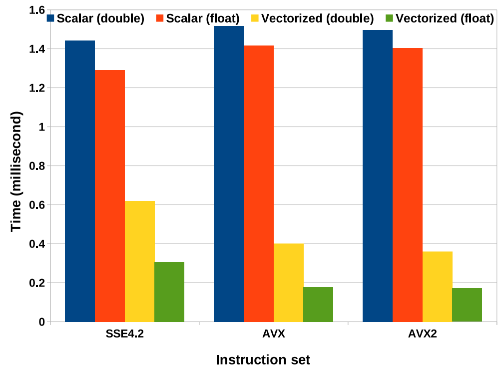

FIG. 2: Comparison of the timing of PoCA calculation using different instruction sets available on a machine.

Conclusions

In this paper, we have presented the preliminary results of the vectorization of the PoCA algorithm using the Vc as the SIMD library. The results of performance improvement obtained with vectorization are presented for SSE4.2, AVX, and AVX2 instruction set architecture. Using vectorized PoCA, multiple muon tracks are processed simultaneously, and a good speedup is obtained which depends on the vector size of CPU. In the future, we will try to vectorize more sophisticated reconstruction algorithm like Maximum Likelihood Expectation Maximization (MLEM), to do the more realistic image reconstruction in less time.

References

[1] GEANT4 collaboration, S. Agostinelli et al., *GEANT4: A Simulation toolkit*, *Nucl. Instrum. Meth. A* **506** (2003) 250.

[2] Johannes Hofmann et. al. *Proceedings of WPMVP '14* (2014)

[3] M. Kretz and V. Lindenstruth., *VC: A C++ library for explicit vectorization.*, *Sofware-Practice I& Experience*, 42, Issue 11, Wiley, New York, USA. (2012)