

# Fast computation of double precision sparse matrix in BCRS and DD vector product using AVX2

Toshiaki Hishinuma<sup>1</sup>, Akihiro Fujii<sup>1</sup>, Teruo Tanaka<sup>1</sup>, and Hidehiko Hasegawa<sup>2</sup>

<sup>1</sup> Faculty of Informatics, Kogakuin University, Tokyo, Japan

<sup>2</sup> Faculty of Lib., Info. & Media Sci., University of Tsukuba, Tsukuba, Japan  
em13015@ns.kogakuin.ac.jp

**Abstract.** In a double precision sparse matrix and a double-double precision (DD) vector product (DD-SpMV) :  $\mathbf{y} = \mathbf{A}\mathbf{x}$  in CRS format using AVX2, the non-contiguous memory access, processing for the remainder and the summation elements in the SIMD register are factors that affect performance. BCRS format with block size equal to the SIMD register's length or its multiples reduces performance degradation. However, since BCRS format consists of zero-elements, it may result in increased computations. We accelerated DD-SpMV in BCRS format using AVX2.

**Keywords:** SpMV, BCRS format, High precision arithmetic

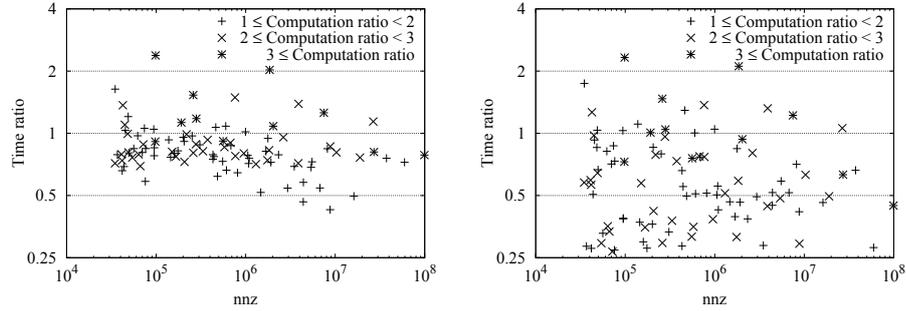
## 1 Introduction

DD arithmetic is a high precision arithmetic system[1]. It uses two double precision variables to implement one quadruple precision variable. We have accelerated DD arithmetic using AVX[2].

Intel AVX2 computes four double precision variables simultaneously. In the compressed row storage (CRS) format[3], SpMV using AVX2 requires processing for the remainder (1, 2, 3) in each row. In SpMV in CRS format using AVX2, the loading of  $\mathbf{x}$  is non-contiguous. Therefore, SpMV in CRS format must relocate  $\mathbf{x}$  for using AVX2. In storing  $\mathbf{y}$ , DD-SpMV using AVX2 requires the summation of the elements in the SIMD register in each row. Processing for the remainder and summation of elements in the SIMD register are factors that affect performance.

The block CRS (BCRS) format[3] partitions the matrix A into  $r \times c$  small dense submatrices (called blocks), which may include some zero-elements. DD-SpMV using AVX2 in BCRS format can reduce factors that affect performance in DD-SpMV in CRS format. However, since BCRS format includes some zero-elements, it may result in increased computations.

We implement BCRS 1x4 and BCRS 4x1. BCRS 1x4 reduces processing for the remainder and can improve memory access. BCRS 4x1 reduces processing for the remainder and summation of elements in the SIMD register, which can improve memory access. We evaluate DD-SpMV in BCRS format using AVX2 with the block size optimized for AVX2.



**Fig. 1.** Elapsed time ratio of DD-SpMV using BCRS format (left : BCRS 1x4 compared to CRS, right : BCRS 4x1 compared to CRS)

## 2 Experimental results

The CPU is a 4-core 8-thread Intel Core i7 4770 3.4 GHz 16 GB. OS is CentOS 6.4 and the compiler is an Intel C/C++ compiler 13.1.0. Compiler options `-O3`, `-xCORE-AVX2`, `-openmp`, and `-fp-model precise` are used. We used a set of 100 matrices that obtained from the University of Florida Sparse Matrix Collection.

Figure 1 shows the elapsed time ratio of DD-SpMV in BCRS format using AVX2. For 83 matrices, the elapsed time of BCRS 4x1 is less than that of CRS. For 94 matrices, the elapsed time of BCRS 4x1 is less than that of BCRS 1x4. In the small size matrix (less than  $3.0 \times 10^6$ ), the elapsed time of BCRS 4x1 is more than that of CRS, because small size matrices make good use of the cache. The effect of BCRS 4x1 is good and BCRS 4x1 is effective for large size matrices.

## 3 Conclusion

We accelerated the double precision sparse matrix in BCRS format and double-double precision vector product using AVX2. The best storage format for AVX2 is BCRS 4x1, which can reduce factors that affect performance in DD-SpMV in CRS format. In the poster session, we will introduce the trade-off between reducing factors that affect performance and increased computations.

## References

1. Bailey, D. H.: High-Precision Floating-Point Arithmetic in Scientific Computation, computing in Science and Engineering, pp. 54-61 (2005).
2. Hishinuma, T., Fujii, A., Tanaka, T., and Hasegawa, H.: AVX acceleration of DD arithmetic between a sparse matrix and vector, the Tenth International Conference on Parallel Processing and Applied Mathematics, Warsaw, Poland (2013.9)
3. Barrett, R., et al.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM pp. 57-65 (1994)