# Efficient Offload Computing for Large-scale Electronic Structures with Multiple Manycore PCI-E Devices

Extended Abstract

Yosang Jeong
Korea Institute of Science and Technology Information
Daejeon 34141, Republic of Korea
yosang.jeong@kisti.re.kr

Hoon Ryu*
Korea Institute of Science and Technology Information
Daejeon 34141, Republic of Korea
elec1020@kisti.re.kr

## ABSTRACT

Fast computations of large-scale sparse matrices is critical in many areas of computational science. Efficient offload computing with multiple manycore PCI-E devices is discussed with a focus on simulations of tight-binding electronic structures that involve $10^7 \times 10^7$ or larger sparse matrices. Schrödinger equations are solved in parallel with Lanczos method. To improve the speed with manycore devices, the hotspot of computations, sparse matrix-vector multiplications (MVmuls), is offloaded with asynchronous offload technic. We accomplish ~1.62x speed-up in total simulations (~2.64x in MVmuls) with two Intel Xeon Phi Knights Corner coprocessors per each node, compared to the case when only host CPUs are used. Asynchronous data-transfer technic we employed significantly mitigates the overhead due to data-transfer between host and multiple coprocessors, so the overhead with two coprocessors becomes just ~1.2x than that with a single coprocessor.

## CCS CONCEPTS

**•Computing methodologies → Massively parallel algorithms;**
**•Mathematics of computing →** *Partial differential equations;*

## KEYWORDS

Tight-binding simulations, Electronic structures, Manycore computing, Xeon Phi coprocessors, Multiple coprocessors

**ACM Reference format:**
Yosang Jeong and Hoon Ryu. 2017. Efficient Offload Computing for Large-scale Electronic Structures with Multiple Manycore PCI-E Devices. In *Proceedings of ACM Symposium on High-Performance Parallel and Distributed Computing, Washington D.C., USA, June 2017 (HPDC 2017),* 2 pages.
DOI: yy.yyy/yyy_y

## 1 INTRODUCTION

Manycore devices have obtained attention as they have potential to increase computing capacity of a single node compared to traditional CPU-based high performance computing (HPC) systems. While on-board manycore systems that do not need data-transfer via PCI-E are released recently, offload computing is still important in HPC communities since a single computing mode of 30% in latest top HPC systems use multiple PCI-E manycore devices such as General-Purpose Graphical Processing Units (GPGPU) and Intel Xeon Phi Knights Corner (KNC) coprocessors. [1]. While partial differential equations are the critical target of computations in various areas of computational science, it is not easy to find many research works that discuss performance enhancement of those operations in HPC systems, where each computing node has multiple PCI-E devices. This work covers strategies that are efficient for offload computing with multiple PCi-E devices, using Xeon Phi KNC coprocessors and an in-house Schrödinger equation solver that has been developed to simulate large-scale electronic structures. While this work focuses on a bit outdated manycore devices (KNC), the strategy presented would be still important as they can be directly applied to GPGPU devices or upcoming Intel Knights Landing (KNL) coprocessors.

## 2 METHODOLOGY

Electronic structures of nanostructures are represented with a $sp^3d^5s^*$ tight-binding approach [2] that assumes nearest-neighbor couplings. Domains of simulations are decomposed in a multi-dimensional way with a hybrid usage of Message Passing Interface (MPI) and OpenMP. Hamiltonian sparse matrices, which are stored in a compressed sparse row format [3], are then decomposed in a row-wise manner. Our Schrödinger equation solver, which computes normal eigenvalue problems in a numerical perspective, is implemented with Lanczos iterations [4] that involve sparse matrix-vector multiplications (MVMuls). To improve the performance of MVMuls with offload computing, each decomposed matrix in a single MPI process is copied into coprocessor(s), and an input/output vector is copied from host/coprocessors to coprocessors/host per each iteration such that host and PCI-E devices can share the computing load of MVMuls at the same time (Fig. 1(a)) [5]. The overhead of data-transfer (particularly for vectors) between host and multiple PCI-E devices in a single computing node is reduced by the technic of asynchronous data-transfer (Fig. 1(b)).

## 3 RESULTS AND DISCUSSION

The performance is benchmarked in a cluster testbed that consists of 3 computing modes connected with an infiniband network. Each computing node has 2, 10-core Intel Xeon E5-2670 v2 (2.5GHz) processors, 128G memory and 2 KNC 7120 coprocessors. The performance of offload computing, measured for end-to-end simulations of a Si:P quantum dot [6] that has a cuboid Si layer of 30×80×80

*Correspondence


*HPDC 2017, Washington D.C., USA*

DOI: yy.yyy/yyy_y

[100] unitcells (a ~15 million ×15 million Hamiltonian matrix), is shown in Fig. 2 with the third control factor "Coproc Load" indicating the fraction of MVmuls computed by coprocessors. In general, results show excellent scalability in multiple nodes regardless of Coproc Load and how many coprocessors are used. We observe that the wall-time becomes minimized at 65% and 80% of Coproc Load when a computing node has one (case 1) and two (case 2) coprocessor(s), respectively, where ~1.48x (case 1) and ~1.62x (case 2) speed-up are observed compared to when only host CPUs are used (Coproc Load = 0). The speed-up in simulations in mainly due to that of MVMuls, which turns out to be ~2.10x and ~2.64x in the case 1 and 2, respectively.

Fig. 3 shows the time consumed by MVmuls in two components, i.e., computation and data-transfer. Since PCI-E is a serial bus [7], the case 2, which transfers vectors to two coprocessors, is expected to have 2x overhead of data-transfer compared to the case 1. Due to the scheme of asynchronous data-transfer (Fig. 1), however, the overhead in the case 2 is just ~1.2x of that in the case 1. The speed-up of MVMuls (including data-transfer) in the case 2 becomes ~1.3x against the case 1, where the speed-up of computation is ~1.53x.

## 4 CONCLUSIONS

Strategies of efficient offload computing for large-scale electronic structure simulations are discussed. Technics of asynchronous offload presented in this work, which include simultaneous executions of large sparse matrix-vector multiplications by host and PCI-E devices, and asynchronous data-transfer between a single host to multiple PCI-E devices, lead non-negligible performance improvement. While here we used Xeon Phi KNC coprocessors as target PCI-E devices, technical details of this work are still applicable to GPGPU devices, and upcoming Xeon Phi KNL coprocessors.

## ACKNOWLEDGEMENTS

This work has been carried out as Intel Parallel Computing Center (IPCC) project funded by Intel Corporation, USA. KISTI-Accelerator-Testbed (KAT) clusters supported by Korea Institute of Science and Technology Information (KISTI) have been extensively used. H. Ryu appreciates J. H. Sohn for all the support for researches.

## REFERENCES

[1] Top 500 List. https://www.top500.org/lists/2016/11. (2016).
[2] J. M. Jancu, R. Scholz, F. Beltram. and F. Bassani. 1998. Empirical $spds*$ tight-binding calculation for cubic semiconductors: General method and material parameters. *Physical Review B* 57 (1998), 6493.
[3] A. Buluç, J. T. Fineman, M. Frigo, J. R. Gilbert, and C. E. Leiserson. 2009. Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks. In *Proceedings of the annual Symposium on Parallelism in Algorithms and Architectures (SPAA)*, 430-435.
[4] C. Lanczos 1950. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards* 45, 4 (1950), 255-282.
[5] H. Ryu, Y. Jeong, J. Kang, and K. Cho. 2016. Time-efficient simulations of tight-binding electronic structures with Intel Xeon Phi™ many-core processors. *Computer Physics Communications* 209 (2016), 79-87.
[6] B. Weber, Y. H. M. Tan, S. Mahapatra, T. F. Watson, H. Ryu, R. Rahman, L. C. L. Hollenberg, G. Klimeck, and M. Y. Simmons. 2014. Spin blockade and exchange in Coulomb-confined silicon double quantum dots. *Nature Nanotechnology* 9 (2014), 430-435.
[7] D. Mayhew, and V. Krishnan. 2003. PCI express and advanced switching: evolutionary path to building next generation interconnects. In *Proceedings of the IEEE Symposium on High Performance Interconnects (HOTI)*, 21-29.

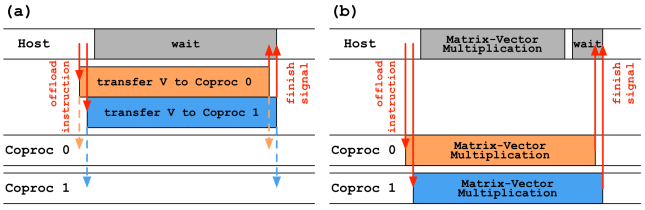



**Figure 1: Strategies for asynchronous offload computing. (a) A scheme of asynchronous data-transfer. Host transfers vectors to the first coprocessors, then to the second coprocessors without waiting the completion of the first transfer. (b) A scheme of asynchronous MVMuls, with which MVMuls are performed simultaneously in host and coprocessors.**

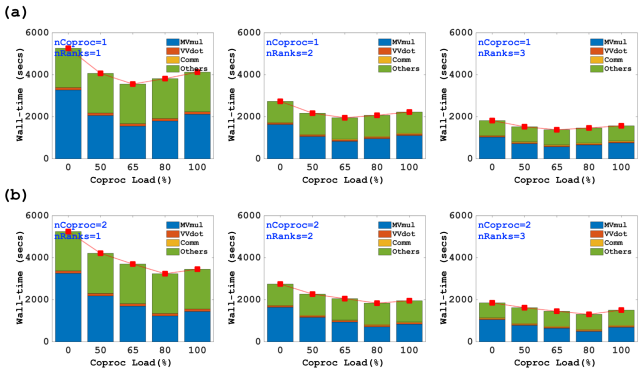



**Figure 2: Performance of simulations with 1/2/3 MPI processes (nodes), when a single MPI process uses (a) one coprocessor and (b) two coprocessors. A Coproc Load of 65%/80% gives the best speed with 1/2 coprocessor(s), where corresponding speed-up compared to the CPU-only case becomes ~1.48x/~1.62x. The scalability is quite nice regardless of Coproc Load and how many coprocessors are used.**

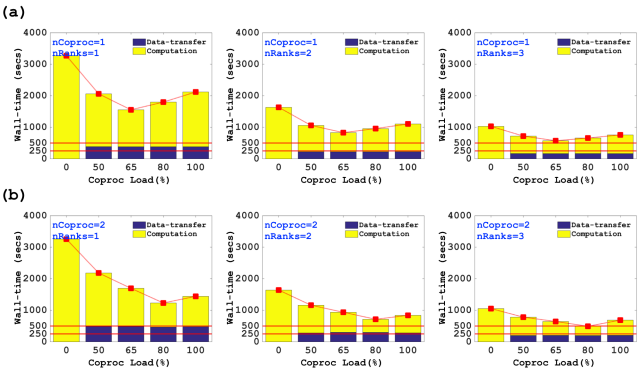



**Figure 3: Performance of MVmuls with 1/2/3 MPI processes (nodes), when a single MPI process uses (a) one coprocessor and (b) two coprocessors. With two coprocessors, the overall speed (computing+data-transfer) is improved by ~1.3x on average, compared to that with one coprocessor, and the average overhead of data-transfer increases by just ~1.2x (not 2x) due to the scheme of asynchronous data-transfer we used.**

# Efficient Offload Computing for Large-scale Electronic Structures with Multiple Manycore PCI-E Devices

Yosang Jeong and Hoon Ryu* (*Correspondence: elec1020@kisti.re.kr)
Korea Institute of Science and Technology Information, Daejeon 34141 Republic of Korea.

## Introduction

### Latest Trends in HPC: Offload Computing

- 29 of Top 100 HPCs use multiple PCI-E devices (Intel® KNC coprocs. or Nvidia® GPU devices) / node (Ref. [1])
- Intel® Knights Landing (KNL) will be also available soon as PCI-E devices
- Involves data-transfer between host and PCI-E devices
- **Issue in performance: The overhead of multiple data-transfer to multi PCI-E devices**

### Nanostructure Modeling: Needs for HPCs

- Characteristics of nanoscale materials affected by:
  - Structural confinement: Quantum physics
  - Roughness/Crystal orientations etc.: Atomistic effects
- Tight-binding (TB) model (Ref. [2])
  - 10/20 orthogonal basis to model one atom
  - Easy to handle atomistic effects
- **Issue in large-scale computing: needs for HPCs**
  - Experimentally realizable nanostructures: size of a few tenth of nanometers involving multi-million atoms
  - Sizes of system matrices: proportional to the number of atoms in structures (scaling factor = # of basis)

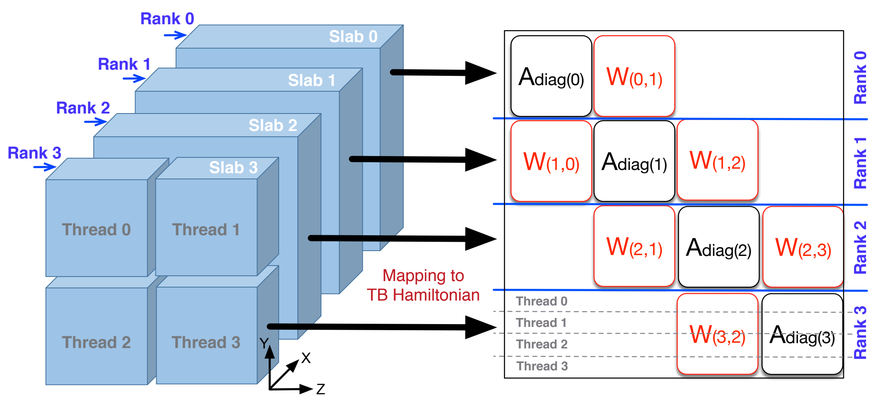
## Methodologies

### Numerical Algorithm: Schrödinger Solver

- Lanczos method (Ref. [4]): Normal eigenvalue prob.
- **Issues: Sparse matrix-vector multiplier (MVmul)**

### Parallelization and Offload Computing

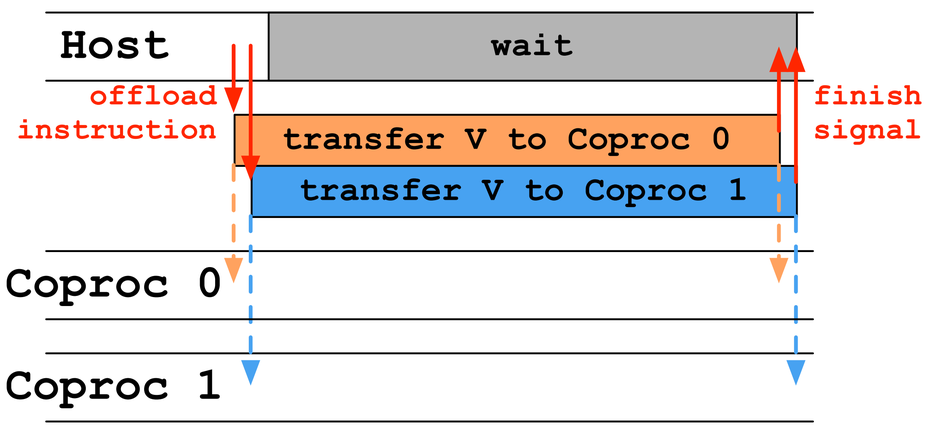
**- Scheme of Domain Decomposition**




[Scheme of Domain Decomposition]

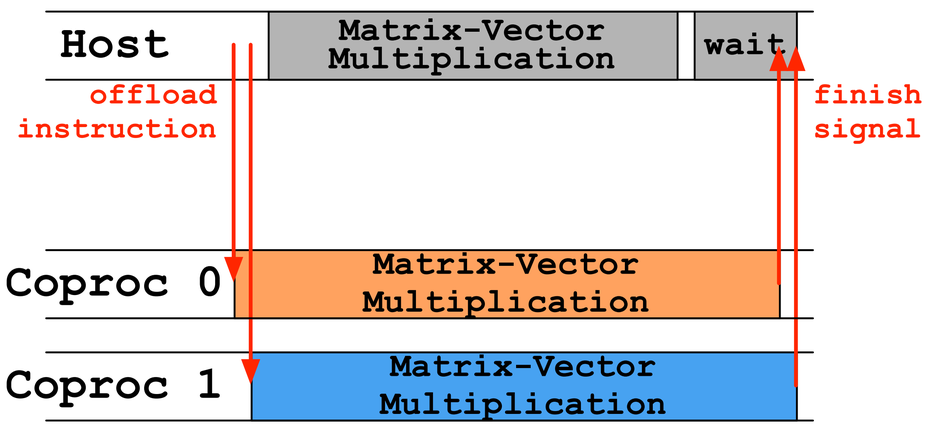
- Decompose along X-direction with MPI
- Decompose along Y and Z-direction with OpenMP

**- Offload technic w/ Multiple coprocessors**




[Asynchronous Data-transfer]

- Host transfer vector to all coprocs. in every iteration
- Overhead in offload computing with multiple coprocs. can be reduced with asynchronous data-transfer




[Asynchronous Matrix-Vector Multiplication]

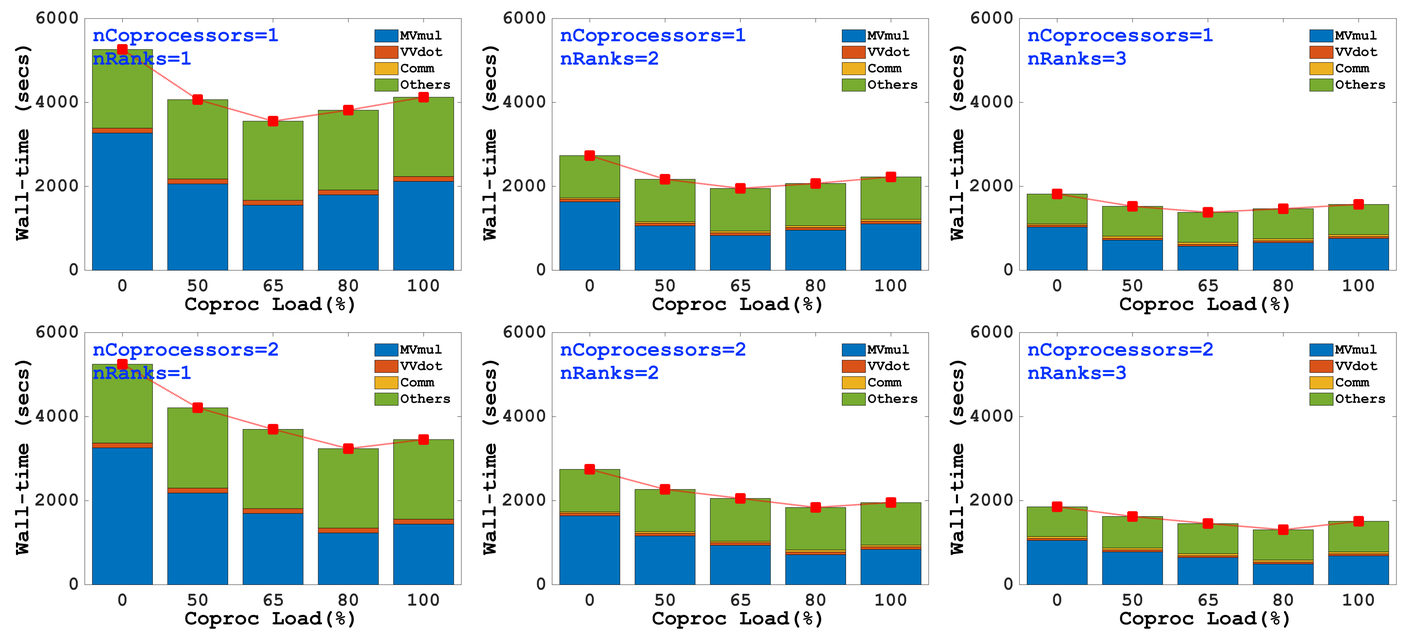
- Host and coprocs. share computing load of MVmul.
  - Matrices are decomposed in a row-wise (Ref. [5])

## Results

### Performance Test Problem and Computing Environment

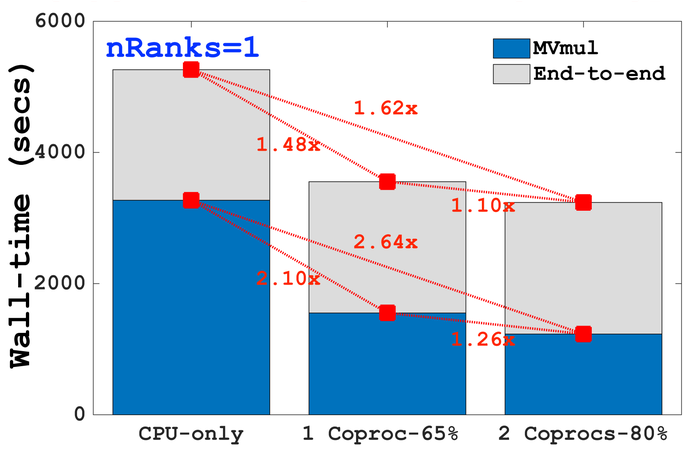
- A phosphorus atom embedded in a 30x80x80 [100] unitcell silicon layer (a cuboid Si:P quantum dot – Ref. [6]): has ~1.5M atoms and involves a ~15Mx15M Hamiltonian matrix with a 10-band spds* TB model
- Intel® Xeon E5-2670 v2(10 core) x2 w/ Xeon Phi KNC 7120A x2 per Node
- 20 threads per MPI rank / 240 threads per Coprocessor

### Performance Improvement w/ Offload Computing



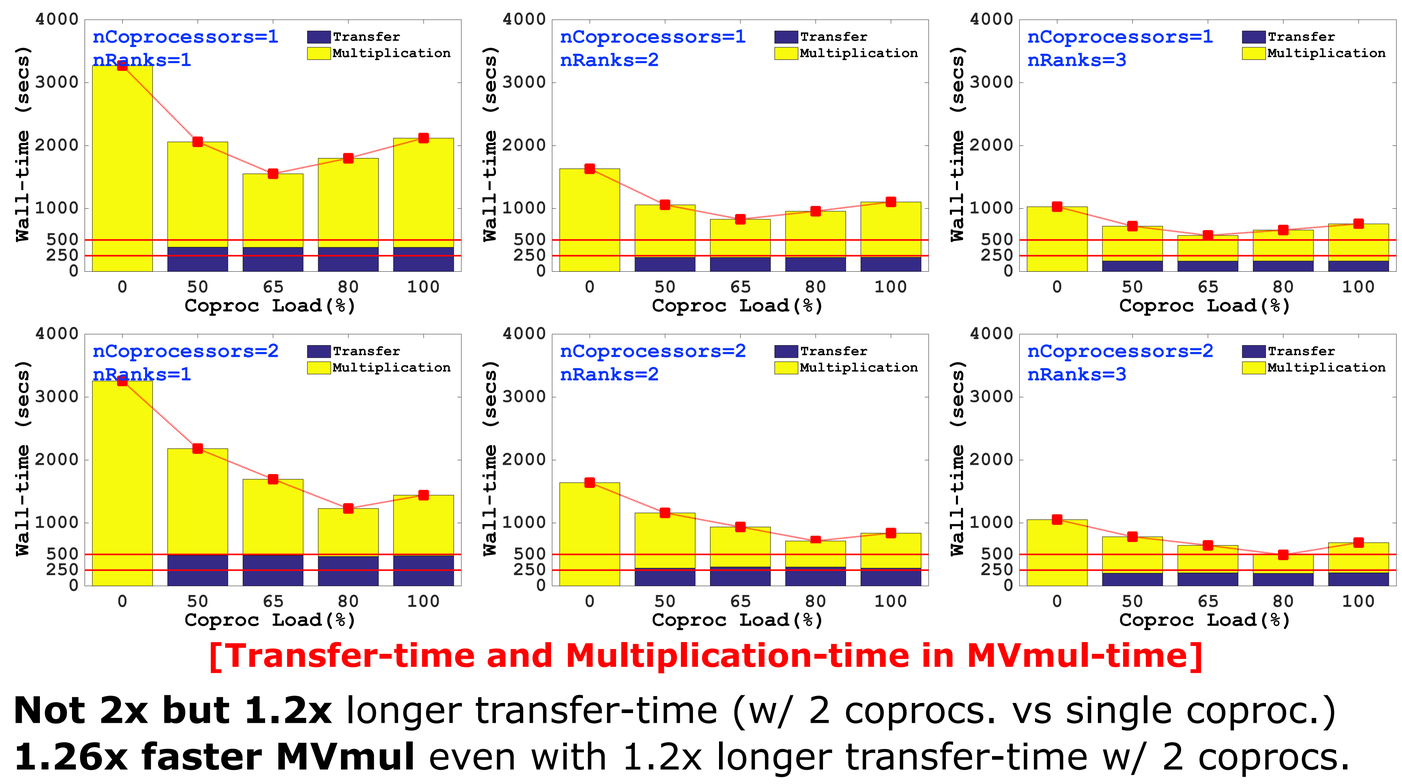

[Computing-time with Single Coprocessor (Up) and 2 Coprocessors (Down)]

- **Coproc Load**: The ratio of computing load (MVmul) in KNC coprocs.
- Optimal performance is observed at
  - Coproc load of 65% (1 coproc.)
  - Coproc load of 80% (2 coprocs. [a])
  - a) 40% per each coproc.
- Excellent strong scalability
  - Tested up to three computing nodes
  - 2.6x faster at 65% load in 1 coproc.
  - 2.5x faster at 80% load in 2 coprocs.




[Optimal Performance in a Single Node]

### Reduction in Transfer-time w/ Asynchronous Data-transfer




[Transfer-time and Multiplication-time in MVmul-time]

- **Not 2x but 1.2x** longer transfer-time (w/ 2 coprocs. vs single coproc.)
- **1.26x faster MVmul** even with 1.2x longer transfer-time w/ 2 coprocs.
  - MVmul-time here includes transfer-time.

## Acknowledgements

This work has been carried out as Intel® Parallel Computing Center (IPCC) project funded by Intel Corporation, USA. KISTI-Accelerator-Testbed (KAT) clusters supported by Korea Institute of Science and Technology Information (KISTI) have been extensively used. H. Ryu appreciates J. H. Sohn for all the support for researches.

## Reference

[1] Top 500 List https://www.top500.org/lists/2016/11/
[2] Phys. Rev. B **57**, 6493 (1998)
[3] Proc. SPAA, 430 (2009)
[4] J. Res. Natl. Bur. Stand. **45**, 255 (1950)
[5] Comput. Phys. Commun. **209**, 79 (2016)
[6] Nat. Nanotechnol. **9**, 430 (2014)
[7] Proc. HOTI, 21 (2003)