Experimental Considerations Towards Effective Memory Bandwidth Evaluation on Large-Scale ccNUMA Systems

Pavel Drobintsev, Vsevolod Kotlyarov, Aleksei Levchenko, and Evgeniy Petukhov

Peter the Great St. Petersburg Polytechnic University, Saint Petersburg, Russia
vpk@spbstu.ru

Abstract. In order to predict the performance of a wide range of scientific applications on current high-end ccNUMA architectures, this paper introduces benchmark-related modeling considerations for memory bandwidth and hybrid MPI/OpenMP performance. We use HPCG, state-of-the-art benchmark, in order to create a workload representative for a multitude of computational and communication tasks. We ran our model validation experiments on real ccNUMA machine with 12Tb RAM in single operating system image mode to define the boundaries of problem size and demonstrate improved indicators for the target architecture as compared with the fundamental model. Our model will permit to evaluate reliably the performance of contemporary and future ccNUMA systems with more than 20Tb of RAM and to compare their experimental results with other problem-oriented architectures worldwide.

Keywords: benchmarking · ccNUMA · HPCG · memory bandwidth · NUMA effects

1 Introduction

The current Cache-Coherent Non-Uniform Memory Access (ccNUMA) systems are able to provide a larger amount of random access memory per node with a single operating system image than it is accessible on an usual cluster. Asymmetric ccNUMA nature raises a number of potentially overwhelming strong NUMA effects such as memory hot-spotting, the substantial penalty of incorrect NUMA assignment, varying complex multilevel structure of latency and mismatch of data access models and actual distribution of data in memory [9, 4, 22]. These factors have a multidirectional impact on memory bandwidth, which continues to be a major system challenge for memory-bound scientific applications. Deducing memory bandwidth from the theoretical peak one for a specific computing procedure is a sophisticated problem [14]. Thereby, hypothetical prediction of ccNUMA systems memory bandwidth is unconvincing.

Our ultimate goal is to measure reliably the performance of current and future ccNUMA systems. In this work, we present only preliminary considerations for the experimental benchmarking, modeling and predicting of ccNUMA memory bandwidth. The High Performance Conjugate Gradients (HPCG) Benchmark was used for creating a workload with the low ratio of computations to data access that is representative for the major communication and computational patterns [6]. As we extend the existing HPCG performance model, we predict the effective memory bandwidth of real system with a globally addressable memory, so-called *jumbonode* equipped with 12Tb of RAM and loaded as a single operating system image. We shall compare the obtained results with other problem-oriented architecture worldwide and predict the effective memory bandwidth of future ccNUMA machines. We also demonstrate valuable technical ccNUMA-related aspects of launching a hybrid HPCG.

The remaining sections of this paper are organized as follows. In Section 2 we shall mention the most important previous works including the reference model. In Section 3 we shall describe the factors considered by us that are able to extend the existing general-purpose model for the ccNUMA architecture. The model validation and experimental results are discussed in Section 4. Finally, we summarize our conclusions in Section 5, where we also consider the aspects of future development of the model.

2 Background and Related Work

We review the previous work in NUMA- and HPCG-related aspects, which will help us to take into account more challenges proposed by the ccNUMA architecture, namely (1) hybrid MPI/OpenMP performance modeling, (2) NUMA effects, which have impact on performance and (3) HPCG-related publications including reference model of HPCG performance.

Wang et al. [19] presents a model, which predicts both memory bandwidth usage and optimal core allocations. Luo et al. [13] provides valuable insights into off-socket and inter-socket bandwidth modeling to analyze performance of different thread and data placements. A hybrid approach for the development of high-level performance models of large-scale computing systems, which combines mathematical modeling and discrete-event simulation has presented in [17]. Work [18] shows advantages of hybrid OpenMP/MPI programming on large-scale NUMA clusters. Other work on performance modeling of communication and computation in hybrid MPI/OpenMP applications is carried out by [1].

As for the HPCG, we already have a number of important works since 2013. Dongarra et. al [6] describes allowed and disallowed HPCG optimizations. Several studies, [24, 3, 11, 10, 12, 5, 2] have been done to describe an early experience of HPCG optimizations on large systems like Tianhe-2, Angara, Sunway TaihuLight System, etc.

A general-purpose performance model [14] of the HPCG Benchmark includes the execution time for main kernels, namely for Symmetric Gauss-Seidel smoother (SymGS), Sparse Matrix Vector Multiplication (SpMV), Vector Update, Global Dot Product (DDOT), as well as Multigrid preconditioner (MG). Together with the model of two communication procedures, the complete model allows us to predict HPCG performance reliably. As implied by the foregoing, HPCG can provide insight into comparsion of ccNUMAs with the results of other problem oriented architectures (non-ccNUMA). The evolutionary aspects and experimental application of the mentioned works are contributions of this work.

3 The Extended Model Features

The contribution made by our work is the prediction of ccNUMA system memory bandwidth by using an reference model from work [14]. The main performance challenges on ccNUMA are (1) locality of data access, (2) the amount of data sharing between threads and (3) effective memory bandwidth [21]. The effective memory bandwidth from main memory participating in the model of all computing procedures is of a greater significance. Our contribution is also in using hybrid HPCG, not only pure MPI like in model [14]. In spite of the facts that HPCG is well balanced at the MPI level, the performance of pure MPI realization is higher and OpenMP does not provide support for ccNUMA, our core point is that the hybrid version is an additional great challenge for ccNUMA architectures per se, providing emergence of a number of effects detrimental to performance, such as memory hot-spotting. Table 1 shows the estimated range of model options that have been considered by us or have such prospect. The features of our model include (1) the execution time in seconds for main kernels (SYMGS, SpMV, etc.) previously presented in [14] and extended in this work to take into account the effective memory bandwidth and interconnect latency, and (2) the effects of hybrid MPI+OpenMP parallelism in ccNUMA environment. In this paper, we describe only the experimental aspects of effective bandwidth evaluation.

Table 1. Comparison of model editions

Model Features	Reference	Extended
$SYMGS_{exec\_time(sec)}$	Considered	$+BW_{eff}$
$SpMV_{exec\_time(sec)}$	Considered	$+BW_{eff}$
$WAXPB_{exec\_time(sec)}$	Considered	$+BW_{eff}$
$DDOT_{exec\_time(sec)}$	Considered	$+BW_{eff}$
$Allreduce, Halo_{exec\_time(sec)}$	Considered	$+IC_{latency}$
Hybrid MPI+OpenMP	Not considered	Considered
Effective bandwidth	Not considered	Considered
IC latency	Not considered	Considered
Optimization techniques	Not considered	Future work

We already know total execution time from the non-hybrid HPCG model [14]:

$$Iter_{time(sec)} = MG + SpMV(depth = 0) + 3(DDOT + WAXPB) \tag{1}$$

Hybrid HPCG is more memory-bound, than pure MPI and can deliver better performance [15], especially in case of the ccNUMA. For OpenMP, execution time proposed by Wu and Taylor for hybrid MPI/OpenMP scientific applications [20] is rewritten as follows:

$$Perf = (Ref_{MPI} + OMP) \times \frac{Total_{exec\_time(sec)}}{Comp_{exec\_time(sec)} + Comm_{exec\_time(sec)}} \tag{2}$$

where OMP represent the model for intranode OpenMP performance:

$$OMP = T_{c1} + (BW_n - 1)\frac{T_{c2} - T_{c1}}{BW_2 - 1} \tag{3}$$

Here we use Eqn. 3 to model the OpenMP application execution time on n cores based on the performance for single and dual cores (T_c) and memory bandwidth ratio (BW_n) [20].

The effective memory bandwidth can be deduced from reference model [14] for every HPCG kernel as follows.

$$BW_{SYMGS}(Bytes/sec) = \frac{(nx \times ny \times nz)/2^{3\times d} \times (20 + 20 \times 27)(Bytes)}{SYMGS_{exec\_time(sec)}} \times 2 \tag{4}$$

$$BW_{SpMV}(Bytes/sec) = \frac{(nx \times ny \times nz)/2^{3\times d} \times (20 + 20 \times 27)(Bytes)}{SpMV_{exec\_time(sec)}} \tag{5}$$

$$BW_{WAXPB}(Bytes/sec) = \frac{(nx \times ny \times nz)/2^{3\times d} \times 24(Bytes)}{WAXPB_{exec\_time(sec)}} \tag{6}$$

$$BW_{DDOT}(Bytes/sec) = \frac{(nx \times ny \times nz)/2^{3\times d} \times 16(Bytes)}{DDOT_{exec\_time(sec)}} \tag{7}$$

where the most expensive routine is SYMGS [16].

While computing procedures were modeled exhaustively, important factors obtained empirically remain. Second of them, after effective memory bandwidth from main memory, is IC latency, whose influence on the prediction is considered as insignificant by the authors of work [14]. We evaluate empirically IC latency by KNEM, a Linux kernel module enabling high-performance intra-node MPI communication for large messages [8].

Table 2. Target system configuration

Architecture details	Standalone server	Single OS image macronodes		
		Minimal	Medium	Jumbonode
RAM	188Gb	752Gb	3Tb	12Tb
NUMA node(s)	6	24	96	384
Board/Socket/Core(s)	1/3/48	4/12/192	16/48/768	64/192/3072

4 Experimental Results and Discussion

Since ccNUMA having more than 3Tb memory size are an exotic systems and it seems complex to obtain a set of various gigantic ccNUMA systems, we use our target system in different configurations presented in Table 2.

For a more in-depth study of NUMA-related challenges, we performed our early-stage experiments with hybrid HPCG running on macronodes from 188Gb of RAM (48 cores) with aggregation of macronode memory to 3Tb of RAM (768 cores) and with subsequent integration into a single macronode with up to ≈12Tb of RAM (3072 cores) at the final stage.

A standalone server is based on AMD Opteron Processor 6380, interconnect has a 3D Torus topology. We use Linux 4.12 with patchset for support of Block Transfer Engine driver for NumaChip node controllers, which provide large number of outstanding memory transactions, memory controller for the cache and memory tags, a cross-bar switch for the interconnect fabric and a number of interconnect fabric link controllers.

Hybrid HPCG run on ccNUMA system with 12Tb is in itself nontrivial problem, which has not been previously described, to the best of our knowledge. Operating system as well as HPCG have been compiled with optimized *libgomp*, which supports stack and thread local storage (TLS) to keep local to more than 1024 threads. A private stack with size up to 2Gb is allocated to each HPCG thread for increasing of problem size, which is very relevant. All MPI processes mapped by NUMA nodes to reduce memory traffic and keep the data close to the cores [15]. Generation of instructions to prefetch memory is used for increasing performance of loops that access large arrays. Load is balanced for improving efficiency of OpenMP application, distributing threads through all accessible NUMA nodes, using more FPU and reducing load on the memory interface and L3 cache. Generation of instructions to prefetch memory is used for increasing performance of loops that access large arrays. The largest allowable size of the problem was 256 × 256 × 256. All start-up options described above have a significant impact on HPCG performance on ccNUMA. Figure 1 shows the results of modeling with the help of the fundamental model that does not take these characteristics into account.

Figure 2 compares our predictions with the actual measured results of hybrid HPCG on the jumbonode with 12Tb of RAM, and the predictions by the reference model have been put to comparison too. In contrast to the results of work [14], the hybrid HPCG scales non-linearly on ccNUMA system; non-uniformity

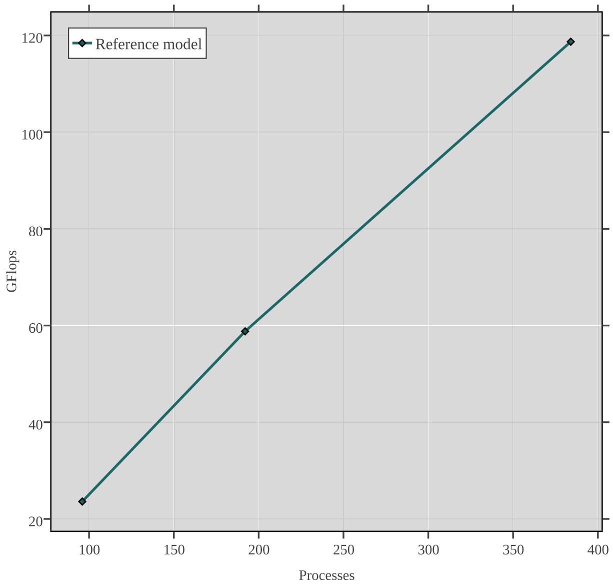

Fig. 1. Reference model prediction for HPCG

of the system results in surface separation whose causes will be studied. And finally in Figure 3 we show the comparison of the modeled ccNUMA bandwidth and the STREAM Benchmark results.

As to predicting the performance of future ccNUMA systems with more than 20Tb memory size, the view taken is that HPCG will remain memory-bound in the future as well. Having about 7Tb memory consumption upon HPCG start with the maximum jumbonode task size, we expect a proportionally high memory consumption since future ccNUMAs will have at least 4Gb per core. IC latency, whose weight in the general HPCG model is insignificant, will grow. Based on our model, we expect the performance of at least 400GFlops for macronode with 20Tb of RAM. In respect of current non-ccNUMA machines, HPCG offers a single metric for comparing various problem-oriented architectures and reduces the gap between them created by LINPACK. E.g., the experimental ccNUMA demonstrates a satisfactory HPCG performance as compared to the results of technical report [5] for “The Sunway TaihuLight supercomputer” [7], suggesting that the ccNUMA memory is slightly slower as compared to the current TOP500 leaders.

5 Concluding Remarks and Future Work

In this work, we presented an experimental approach to contemporary ccNUMA systems memory bandwidth evaluation. HPCG Benchmark was used to create a

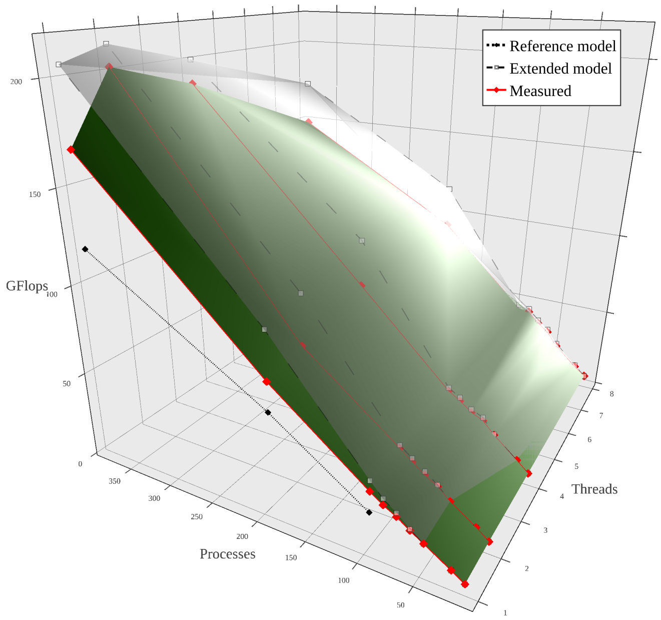

Fig. 2. Modeled and measured HPCG results on the target full-sized jumbonode with 12Tb of RAM

workload comparable to the contemporary scientific applications. The existing HPCG performance model was extended by considering hybrid MPI/OpenMP and supplemented by the factors influencing memory bandwidth. As a result, the effective memory bandwidth of an real ccNUMA system with 12Tb of RAM was predicted. The approach used by us can be applied to comparing of current and future ccNUMA machines.

As implied by the foregoing, the divergence between the actually obtained using STREAM Benchmark results and the deduced from reference model ones is up to 12%. As was demonstrated in Section 4, the whole software environment was optimized on a wide scale, namely the Linux kernel, gcc, libgomp, etc. However, large-scale optimizations of the HPCG itself are still possible. In the near future we plan to concentrate for realization of the existing HPCG optimizations for ccNUMA case as “improving the performance of HPCG will improve the performance of real applications” (J.Dongarra, et al. [6]).

First of all, we consider the refinement of the cache locality model with the help of the novel HPCG optimization technique proposed in the paper [2], namely coloring along two areas XY at a time in SymGS. Among other improvements a number of works argue to replace the default CSR matrix storage format with simplified SELLPACK for SpMV and SYMGS kernels [24, 2]. Table 3 shows the expected speedup. Also recent work [23] demonstrates new data redeployment model which allows to reduce the remote memory access overhead

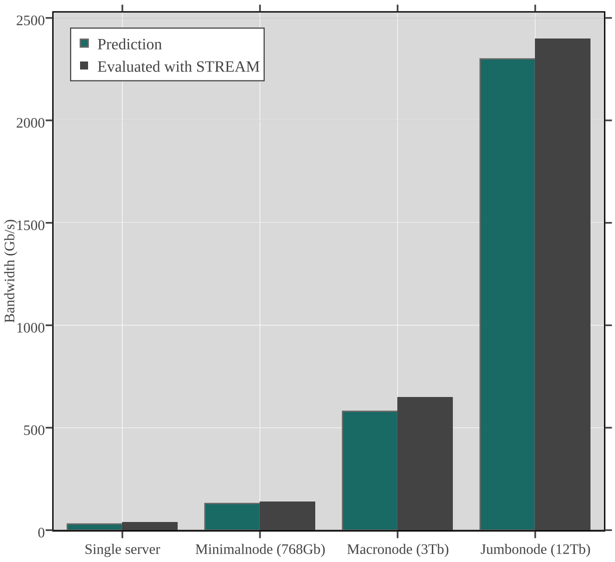

Fig. 3. STREAM benchmark results vs. prediction

Table 3. Planned optimizations and expected speedup

Optimization Techniques	Expected Results
Coloring along two areas XY at a time in SYMGS [2]	×3 SYMGS performance
CSR → ELLPACK [2]	5% speedup in SYMGS and SPMV
Data redeployment [23]	Better performance for large scale problem

for computation-intensive applications with large size of problem in ccNUMA architecture. These optimizations presume an analysis that will allow to study better the challenges proposed by the ccNUMA architecture. Finally, we plan to propose an IC latency model for ccNUMA systems in the near future.

Acknowledgments. This work was financially supported by the Ministry of Education and Science of the Russian Federation within the framework of the Federal Targeted Programme for Research and Development in Priority Areas of Advancement of the Russian Scientific and Technological Complex for 2014-2020 (№ 14.584.21.0022, ID RFMEFI58417X0022). The results were obtained using ccNUMA system in Supercomputer Center of Peter the Great St. Petersburg Polytechnic University.

References

1. Adhianto, L., Chapman, B.: Performance modeling of communication and computation in hybrid mpi and openmp applications. In: 12th International Conference on Parallel and Distributed Systems - (ICPADS'06). vol. 2, pp. 6 pp.– (2006)
2. Agarkov, A., Semenov, A., Simonov, A.: Optimized implementation of HPCG benchmark on supercomputer with "Angara" interconnect. In: Voevodin, V., Sobolev, S. (eds.) Proceedings of the 1st Russian Conference on Supercomputing — Supercomputing Days 2015. CEUR Workshop Proceedings, vol. Vol-1482, pp. 294–302. Research Computing Center, Moscow State University, CEUR-WS.org, Moscow (Sep 28–29, 2015), http://ceur-ws.org/Vol-1482/294.pdf
3. Chen, C., Du, Y., Jiang, H., Zuo, K., Yang, C.: HPCG: Preliminary evaluation and optimization on Tianhe-2 CPU-only nodes. In: Computer Architecture and High Performance Computing (SBAC-PAD), 2014 IEEE 26th International Symposium on. pp. 41–48 (Oct 2014)
4. Diener, M., Cruz, E.H., Navaux, P.O.: Modeling memory access behavior for data mapping. International Journal of High Performance Computing Applications (2016), http://hpc.sagepub.com/content/early/2016/04/13/1094342016640056.abstract
5. Dongarra, J.: Report on the Sunway Taihulight System. Tech. Rep. UT-EECS-16-742, Oak Ridge National Laboratory,Department of Electrical Engineering and Computer Science, University of Tennessee (Jun 2016)
6. Dongarra, J., Heroux, M.A., Luszczek, P.: High-performance conjugate-gradient benchmark: A new metric for ranking high-performance computing systems. International Journal of High Performance Computing Applications 30(1), 3–10 (2016)
7. Fu, H., Liao, J., Yang, J., Wang, L., Song, Z., Huang, X., Yang, C., Xue, W., Liu, F., Qiao, F., Zhao, W., Yin, X., Hou, C., Zhang, C., Ge, W., Zhang, J., Wang, Y., Zhou, C., Yang, G.: The Sunway TaihuLight supercomputer: system and applications. Science China Information Sciences 59(7), 1–16 (2016), http://dx.doi.org/10.1007/s11432-016-5588-7
8. Goglin, B., Moreaud, S.: Knem: A generic and scalable kernel-assisted intra-node mpi communication framework. J. Parallel Distrib. Comput. 73(2), 176–188 (Feb 2013), http://dx.doi.org/10.1016/j.jpdc.2012.09.016
9. Li, T., Ren, Y., Yu, D., Jin, S., Robertazzi, T.: Characterization of input/output bandwidth performance models in NUMA architecture for data intensive applications. In: 2013 42nd International Conference on Parallel Processing. pp. 369–378 (Oct 2013)
10. Liu, F., Yang, C., Liu, Y., Zhang, X., Lu, Y.: Reducing communication overhead in the high performance conjugate gradient benchmark on Tianhe-2. In: Distributed Computing and Applications to Business, Engineering and Science (DCABES), 2014 13th International Symposium on. pp. 13–18 (Nov 2014)
11. Liu, Y., Zhang, X., Yang, C., Liu, F., Lu, Y.: Accelerating HPCG on Tianhe-2: A hybrid CPU-MIC algorithm. In: 2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS). pp. 542–551 (Dec 2014)
12. Liu, Y., Yang, C., Liu, F., Zhang, X., Lu, Y., Du, Y., Yang, C., Xie, M., Liao, X.: 623 Tflop/s HPCG run on Tianhe-2: Leveraging millions of hybrid cores. International Journal of High Performance Computing Applications 30(1), 39–54 (2016), http://hpc.sagepub.com/content/30/1/39.abstract
13. Luo, H., Brock, J., Li, P., Ding, C., Ye, C.: Compositional model of coherence and NUMA effects for optimizing thread and data placement. In: 2016 IEEE Interna-

tional Symposium on Performance Analysis of Systems and Software (ISPASS). pp. 151–152 (April 2016)

14. Marjanović, V., Gracia, J., Glass, C.W.: Performance modeling of the HPCG benchmark. In: Jarvis, A.S., Wright, A.S., Hammond, D.S. (eds.) High Performance Computing Systems. Performance Modeling, Benchmarking, and Simulation: 5th International Workshop, PMBS 2014, New Orleans, LA, USA, November 16, 2014. Revised Selected Papers. pp. 172–192. Springer International Publishing, Cham (Dec 5–9, 2015)
15. Nakajima, K.: Flat MPI vs. Hybrid: Evaluation of parallel programming models for preconditioned iterative solvers on "T2K Open Supercomputer". In: 2009 International Conference on Parallel Processing Workshops. pp. 73–80 (Sept 2009)
16. Park, J., Smelyanskiy, M., Vaidyanathan, K., Heinecke, A., Kalamkar, D.D., Liu, X., Patwary, M.M.A., Lu, Y., Dubey, P.: Efficient shared-memory implementation of high-performance conjugate gradient benchmark and its application to unstructured matrices. In: SC14: International Conference for High Performance Computing, Networking, Storage and Analysis. pp. 945–955 (Nov 2014)
17. Pllana, S., Benkner, S., Xhafa, F., Barolli, L.: Hybrid performance modeling and prediction of large-scale computing systems. In: Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on. pp. 132–138 (March 2008)
18. Tsuji, M., Sato, M.: Performance evaluation of OpenMP and MPI hybrid programs on a large scale multi-core multi-socket cluster, T2K open supercomputer. In: 2009 International Conference on Parallel Processing Workshops. pp. 206–213 (Sept 2009)
19. Wang, W., Davidson, J.W., Soffa, M.L.: Predicting the memory bandwidth and optimal core allocations for multi-threaded applications on large-scale NUMA machines. In: 2016 IEEE International Symposium on High Performance Computer Architecture (HPCA). pp. 419–431 (March 2016)
20. Wu, X., Taylor, V.: Performance modeling of hybrid MPI/OpenMP scientific applications on large-scale multicore cluster systems. In: Computational Science and Engineering (CSE), 2011 IEEE 14th International Conference on. pp. 181–190 (Aug 2011)
21. Yang, R., Antony, J., Rendell, A.P.: A simple performance model for multithreaded applications executing on non-uniform memory access computers. In: High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on. pp. 79–86 (June 2009)
22. Zeng, D., Zhu, L., Liao, X., Jin, H.: A Data-Centric Tool to Improve the Performance of Multithreaded Program on NUMA, pp. 74–87. Springer International Publishing, Cham (2015)
23. Zhang, M., Gu, N., Ren, K.: Optimization of computation-intensive applications in cc-NUMA architecture. In: 2016 International Conference on Networking and Network Applications (NaNA). pp. 244–249 (July 2016)
24. Zhang, X., Yang, C., Liu, F., Liu, Y., Lu, Y.: Optimizing and scaling HPCG on Tianhe-2: Early experience. In: Sun, X.h., Qu, W., Stojmenovic, I., Zhou, W., Li, Z., Guo, Huaand Min, G., Yang, T., Wu, Y., Liu, L. (eds.) Algorithms and Architectures for Parallel Processing: 14th International Conference, ICA3PP 2014, Dalian, China, August 24-27, 2014. Proceedings, Part I. pp. 28–41. Springer International Publishing, Cham (2014)