2017 13th International Conference on Semantics, Knowledge and Grids

Design a Novel Memory Network for Processor-in-Memory Architectures

Slo-Li Chu*, Wen-Chih Ho, Chien-Fang Chen, Kai-Wei Ceng, Ming-Han Liu

Department of Information and Computer Engineering

Chung Yuan Christian University

Chung Li, Taiwan

[1] slchu@cycu.edu.tw, [2]g9877606@cycu.edu.tw, [3]g9677606@cycu.edu.tw, [4]g10477012@cycu.edu.tw, [5]g10477011@cycu.edu.tw

***Abstract*—The growing requirement of data-intensive computing makes the problem of insufficient memory bandwidth more critical. The advantages of multicore architectures and advanced parallel computers are limited. The new kind of architecture, Processor-in-Memory (PIM), is developed to solve the above challenge by integrating the computing logics and tiny processors into the DRAM chip. The data processing capability of the memory subsystem can be improved. However, the bandwidth of the conventional interconnection networks can not satisfy the bandwidth consumption of multiple PIM modules. Therefore, a new memory network, MemGrid, is proposed for connecting multiple PIM memory modules and CPUs. The proposed MemGrid network has the capabilities of high scalability and low diameter. The connection topologies of MemGrid network with the corresponding network switch architecture are discussed. The experimental results show that the MemGrid network can achieve better performance than other interconnection networks in variant accessing patterns and configurations.**

Keywords: Memory Network, Processor-in-Memory Architecture, Multicore Architecture.

I. INTRODUCTION

The growing computation requirements of social media and video streaming make the modern high performance computers aim for the data-intensive operation, instead of conventional computing-intensive operation. The effect of Memory Wall [1] becomes dominate. Therefore modern high performance computers [2] [5] attempt to improve the performance and throughput of the memory subsystem. The novel computer architecture, Processor-in-Memory (PIM), is proposed to overcome the performance bottleneck of the memory subsystem [3]. These architectures integrate processing elements or simple processor into memory chip to reduce the data movement among host processors and memory chips. The workload of host processors can be reduced. The effect memory bandwidth can be improved accordingly.

Due to the dramatically improvement of semiconductor and 3D-stacking packaging technologies, many PIM architectures are proposed, such as Hybrid Memory Cube [4] and TrueNorth [5], are merging processing logics and memory cells into a single chip to enhance the performance of accessing memory. Meanwhile the raw data that are stored into memory bank can be preprocessed to enlarge the actual bandwidth of memory subsystem. Due to the advantage of data-intensive capability, these architectures are also suitable for the modern artificial intelligence applications, such as semantic linking space, pattern recognition, and cyber-physical-socio intelligence [8]. However, in order to extend the capabilities of parallelism and computation performance, these PIM-based computer systems are consisted of many PIM memory modules. The communication mechanism among these PIM modules become a major problem while achieving better performance from memory subsystem.

The interconnection networks of the conventional multicore architectures focus on the communication among processors. Hence the memory data are transferred via the inter-processor network and processed by the attached processor. However, in the PIM-based architecture, the processing elements or computing cores inside the PIM module have the capability of accessing memory banks. The large amount of data exchange among these PIM modules may congest the inter-processor network of the multicore computer system.

Accordingly, this paper proposes a new memory interconnection network, called MemGrid, which enables the capabilities of planer and low communication distance. The packets of remote memory access can be completed rapidly, without the transportation of the attached processors. The performance of the PIM-based computer system can be improved. The network topology and network switch architecture are discussed. The experimental results show that the proposed MemGrid memory network obtains better performance than conventional on-chip networks, such as Mesh [9] and Torus [9] while the range of memory accessing is vast. The required communication distance is shorter than these two widely adopted interconnection networks.

II. RELATED WORKS

0-7695-6304-X/17/31.00 ©2017 IEEE
DOI 10.1109/SKG.2017.00018

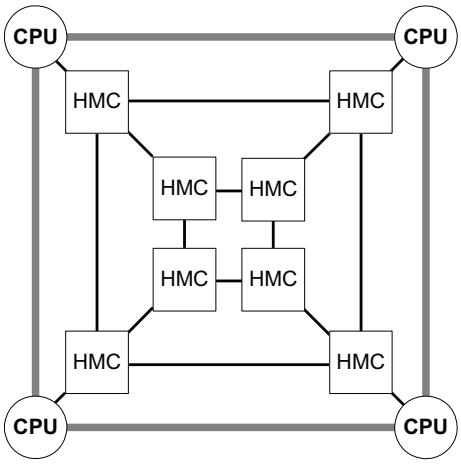
The Processor-in-Memory architectures integrate the computation circuits or tiny processing cores into DRAM chip, to reduce the bandwidth requirement of transferring the unprocessed data by computing the data locally. Therefore many researches adopt the concept of PIM to improve the memory access bandwidth and data processing capability of developing high-performance computing systems. The related studies are discussed as follows.

Hybrid Memory Cube (HMC) [4], proposed by Micron, is a new memory standard, which can improve the bandwidth and features of memory chip. The bandwidth and performance of memory subsystem can be enlarged. The HMC module is consisted of several DRAM layers that can be divided into several partitions. The DRAM layers are connected and stacked by Through-Silicon Via. The DRAM layers belong to the same partition is managed by a Vault Controller. The Vault Controller also helps to forward the memory access requests. Due to the HMC memory standard support serialized packet transmission, the memory access requests can be packed as the memory request packets without managing the detailed DRAM control signals. Many industries develop their new computer systems and platforms that adopt HMC memory modules. Besides, the study [7] focuses on Hybrid Memory Cube module and proposes an interconnection network, which integrates memory-centric network and distributor-based network to shorten the diameter between processor and HMC modules.

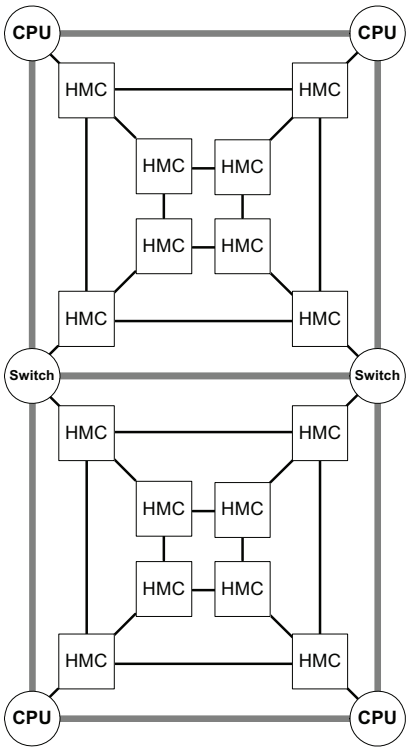
TrueNorth [5] architecture, proposed by IBM, is kind of Neuromorphic computer architecture. TrueNorth is designed for simulating the operating mechanism of animal brain, and accelerate the cognitive computing and recognition. TrueNorth chip is consisted of 4096 processing cores, to be the neurosynaptic core. Each neurosynapatic core contains 256 Axons and 256 Neuron, and integrates programmable synapses. The neurosynapatic cores are connected by the Mesh on-chip network. The TrueNorth architecture can simulate the parallel operations of the brain but requires redesign the algorithm and program to fulfill the architecture characteristics.

III. The MemGrid Network Overview

The Processor-in-Memory architecture combines the computing cores and DRAM cells into the DRAM module, the computation of the data-intensive operations can be accomplished in the DRAM module, without the redundant transfers between DRAM and external CPU. The effective memory bandwidth can be enlarged. However, the conventional memory network is designed for improve the bandwidth of memory data transferring between CPU and DRAM. The bandwidth requirement of communication between PIM modules is not considered. Hence a novel memory interconnection network, MemGrid, is proposed in this paper. Based on the advantages of proposed Self-Similar Cubic [6] multicore interconnection network, the proposed MemGrid network has the capabilities of flatting into chip and low diameter. Compare to conventional on-chip network, such as Mesh and Torus, the MemGrid has superior performance. Besides, MemGrid network is suitable for inter-PIM communication. The Hybrid Memory Cube [4] memory architecture is adopted as the PIM memory model in this paper. The MemGrid topologies and the architecture of network switch are discussed later.



(a)



(b)

Fig. 1 The topologies of MemGrid networks. (a) The topology of MemGrid Network with eight HMC modules. (b) The topology of MemGrid network with sixteen HMC modules.

Figure 1 illustrates the two topologies of MemGrid network that are discussed in this paper. Figure 1 (a) shows a basic MemGrid configuration, which contains four CPUs and 8 HMC memory modules. Since this configuration has symmetric connections in X-axis and Y-axis, the diameters of the packets are low. The number of HMC memory modules can be enlarged easily. The second MemGrid configuration is as shown in Figure 1 (b), which contains four CPUs and sixteen HMC memory modules. This configuration illustrates another type of scaling up number of HMC memory modules in an asymmetric approach. Due to the numbers of horizontal and vertical HMC modules are not the same, the diameters of the memory request packets will be increased if the source and destination HMC modules will not in the same MemGrid block. But this topology has more flexibility for adding more HMC memory modules.

In these two MemGrid network topologies, in order to meet the high speed requirements of inter-CPU communication, there are dedicated paths in the MemGrid network and form as the mesh topology. When CPUs attempts to access one of the HMC memory modules, they can choose the path in the MemGrid block via the inter-HMC cubic paths, or choose the inter-CPU mesh paths, depend on the distance and congestion. The memory request packets of HMC modules can directly transport via inter-HMC cubic network. Accordingly, MemGrid network can provide more bandwidth to satisfy the requirement of Processor-in-Memory architecture that enable the in-memory computing and memory accessing. The architecture of MemGrid HMC switch will be discussed later.

IV. The Architecture of the MemGrid HMC Switch

In the MemGrid memory network, the HMC module is attached on the network via MemGrid HMC Switch. The packets of memory access request that generated from CPU are forwarding via network switch, and then dispatch to the corresponding Vault Controller and DRAM banks. The required external channels for attaching on MemGrid network, the channel buffers, the internal network, and the routing mechanism of the network switch have to be designed. Hence the MemGrid HMC Switch is proposed. The architecture of MemGrid HMC Switch is illustrated as in Figure 2.

The proposed MemGrid HMC Switch module comprises multiple Channels and corresponding Input/Output queues, the Internal Network, and the Switch Routing Unit. Three of these Channels connect to neighbor HMC modules, and one for connecting to CPU, and one for connecting to its own HMC memory module. All Channels equip an Input and Output Buffer, which store the packets for further processing. The input packets temporarily kept in the Input Buffer and then send the request to the Switch Routing Unit, to obtain the next Output Buffer by the current status and the final target of the packet. The packets that are sent to Output Buffer can directly transferred to the next HMC modules while the targeting Input Buffer is empty. All of the Input Buffers and Output Buffers are attached on the Internal Network, which can arbitrate and forward packets from source Input Buffer to the target Output Buffer in the MemGrid HMC switch. The Switch Routing Unit accepts the all the requests from the packets in the Input Buffers that are have to determine the target Output Buffer by the proposed routing algorithm, and find a suitable path and the further stop.

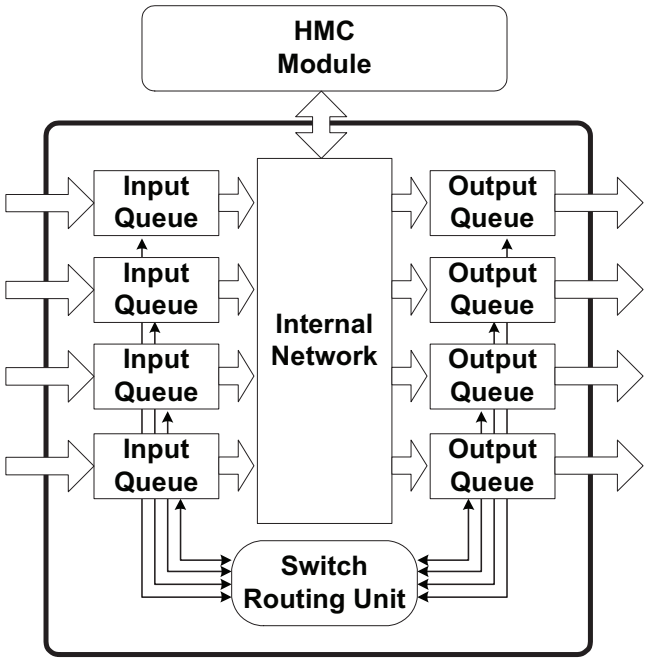


Fig. 2 The architecture of the MemGrid HMC Switch.

V. Analysis of Network Diameters

In order to compare the traveling distances of the memory access packets on the memory networks, two configurations that contain eight HMC memory modules and sixteen HMC modules are illustrated respectively. Three networks, included Mesh [9], MemGrid, and Torus [9], are discussed in each configuration. The adopted metric of traveling distance is Maximum Hop Count. The comparison results are listed in Figure 3.

First we compare the maximum hop counts of the three networks by the configuration of eight HMC memory modules. In these networks, Mesh has larger maximum hop count than MemGrid and Torus because the numbers of

HMC memory module of X-axis and Y-axis are not even. Although the same situation is occurred in Torus network, the wrap-around links of Torus relax the problem. The maximum hop counts of MemGrid and Torus are the same accordingly.

Then we discussed the traveling distance among these three networks by sixteen HMC modules configuration. Due to the horizontal and vertical HMC module numbers of MemGrid network are imbalanced; the maximum hop count is increased. But the maximum hop count of MemGrid is still remained the same as in Torus network without the effect of wrap-around links. Compared to the Mesh network, MemGrid network still has less maximum hop count. The advantage of MemGrid network can be revealed accordingly.

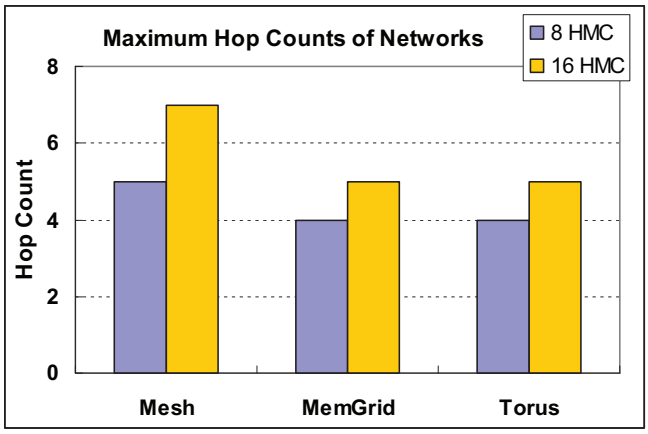


Fig. 3 The maximum hop counts of Mesh, MemGrid, and Torus networks

VI. Analysis of Network Link Numbers

The higher link number of the interconnection network will enlarge the chip fabrication complexity and chip area. The chip cost will be increased accordingly. Figure 4 compares the link numbers of three networks, which includes Mesh, MemGrid. Then we adopts two configurations of different number of HMC memory modules, includes eight HMC memory modules and sixteen HMC memory modules. According to Figure 4, in the eight HMC configurations, the Mesh network has lower link number than other two networks due to the network switch only requires four external links to complete the Mesh topology. The Torus network requires more links to build its wrap-around path to shorten the maximum hop count as we discuss former. The MemGrid network also has more links than the Mesh network due to it contains the CPU-to-CPU links to provide high speed channels for inter-CPU communication.

In the configuration of sixteen HMC memory modules, the Mesh network has lower link number than other networks because Mesh network switch only needs four external links to other neighbor network switches. Although Torus network require wrap-around paths to connect the beginning and ending nodes, the increasing links still less then the requirement of the CPU-to-CPU paths in the MemGrid network. Accordingly, MemGrid has the highest link number than other two networks. But MemGrid has the direct paths for inter-CPU communication.

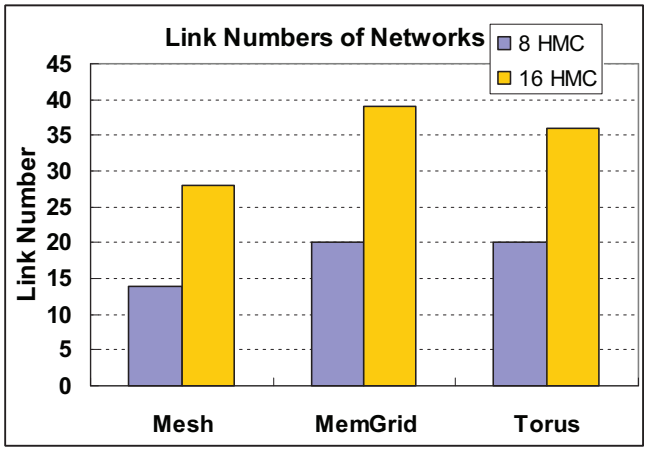


Fig. 4 The link numbers of Mesh, MemGrid, and Torus networks.

VII. Experimental Results

In order to evaluate the performance difference among the proposed MemGrid, Mesh, and Torus networks, we design the models of CPU pattern generator, HMC memory module, the proposed MemGrid HMC switch, the network switches for Mesh and Torus networks. These models are created by using SystemC modeling language. These models are all cycle-accurate. The proposed MemGrid network, Mesh network, and Torus network are composed by the above models. The model of HMC memory module is based on the study [4], which consists of four Vault Controllers. The Vault Controller can access the attached four DRAM banks. These Vault Controllers are connected by internal network. As the HMC memory module connects to the MemGrid HMC Switch or other network switch, they are able to communicate with each other via the memory networks. The HMC memory module is also adopted in Mesh and Torus network to evaluate the performance difference among these three networks. The experimental results of the configurations with eight HMC modules and sixteen HMC modules are discussed as following.

This paper proposes four synthetic patterns with variant memory accessing ranges, which includes the accessing range from 0 to 32MB (u32), the accessing range from 0 to 64MB (u64), the accessing range from 0 to 128MB (u128), and the accessing range from 0 to 256MB (u256),

respectively. These memory access patterns are sent by four CPU, and access the HMC memory module that handle the targeting memory address. Then the targeting DRAM bank will send the memory data or acknowledgement back to the CPU. All memory accessing locations of the patterns are distributed uniformly random and obey the given memory access range. The total packet number of a pattern is 4000.

Figure 5 compares the execution times of four synthetic patterns, which are executed in MemGrid, Mesh, and Torus networks. The adopted configuration is eight HMC memory modules. In these four synthetic patterns, the MemGrid network has better performance than Mesh and Torus networks due to the hop counts of the packets are less than other two networks. In the memory accessing ranges of u32 and u64, the wrap-around links of the Torus network may help to reduce the diameters of the packets while their targeting HMC memory modules are limited. Accordingly, the Torus network has better performance than then Mesh network. While the memory accessing range is increased, as the synthetic pattern of u128, the performance advantage of the Torus network will be decreased due to the congestion of packets on the wrap-around links. So as in the synthetic pattern of u256, the performance of the Mesh network is slightly better than the Torus network. Therefore, the MemGrid network is more suitable the configuration of eight HMC memory modules and situations of the wider memory accessing range. The overall execution time of the MemGrid network is less than other two networks.

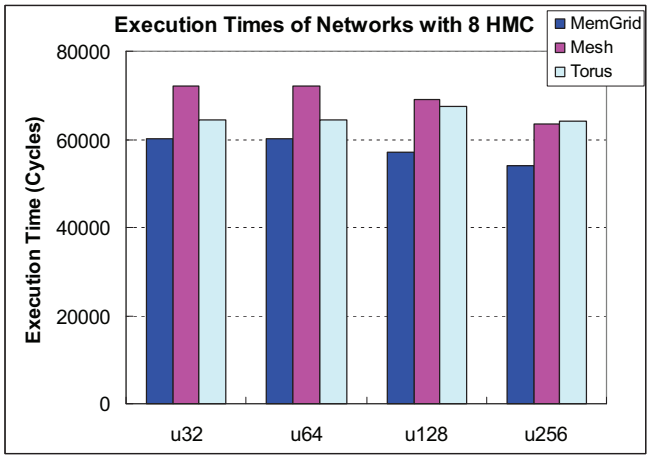


Fig. 5 The execution times of MemGrid, Mesh, and Torus networks with 8 HMC modules.

The performance difference among MemGrid, Mesh, and Torus, with the configuration of sixteen HMC memory modules, is shown in Figure 6. In this configuration, two MemGrid basic blocks are merged to construct the configuration of sixteen HMC memory modules. Due to the numbers of HMC modules in the X-axis and Y-axis are not even, the diameters of the packets will be increased. While the accessed HMC memory modules are concentrated, as the pattern of the u32, the execution time of the MemGrid network is slightly higher than the Torus network due to the longer diameter of the packets. As the access ranges are more extensive, such as in the patterns of u64 and u128, the MemGrid network obtains better performance than Mesh and Torus networks. However, when the accessing range of the packets is vast, such as the pattern of u256, it may degrade the transporting performance of the packets. That may cause by the larger diameter of the packet that accesses the HMC memory modules outside the MemGrid basic block. The required traveling distances of the packets are increased, the consuming delay is enlarged. But the performance is still better than the Mesh and Torus networks. Accordingly, the MemGrid has outperforming performance even when the topology is asymmetric.

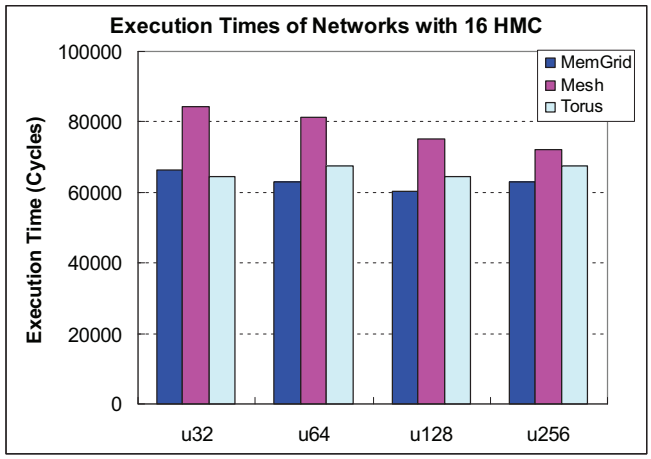


Fig. 6 The execution times of MemGrid, Mesh, and Torus networks with 16 HMC modules.

The above experiments show that the MemGrid memory network is able to achieve higher performance than the conventional networks, such as Mesh and Torus networks. The MemGrid network not only can obtain the performance from the narrow accessing ranges but also suitable for wider ranges of memory accessing. The hardware cost and fabrication complexity will not increased a lot, compared to Mesh and Torus networks. If the number of HMC memory modules is suitable for MemGrid network, the diameter and link number can be reduced accordingly.

VIII. CONCLUSIONS

This study proposed a new memory interconnection network, MemGrid, for the Processor-in-Memory (PIM) architecture. The MemGrid network can solve the problem of communication among the PIM memory modules in the PIM architecture. The MemGrid network has the high

scalability and low diameter characteristics for connecting multiple PIM memory modules. This paper also develops two configurations of variant HMC memory modules to demonstrate the differences among three kinds of networks, by comparing the diameter and link number. The experimental results reveal that the MemGrid network obtains higher performance than other networks by applying variant patterns of memory accessing range. In the configuration of eight HMC memory modules, the MemGrid network can achieve 1.21X speedup than the Mesh network, and 1.18X speedup than the Torus network. In the configuration of sixteen HMC memory modules, the proposed MemGrid can obtain 1.28X speedup and 1.07X speedup than the Mesh and Torus networks respectively. The advantage of the MemGrid network can be revealed accordingly.

ACKNOWLEDGMENT

This work is supported in part by the Ministry of Science and Technology of Republic of China, Taiwan under Grant MOST 105-2221-E-033-047.

REFERENCES

[1] J. Hennessy and D. Patterson, “Computer Architecture-A Quantitative Approach.” 4th Ed., Morgan-Kaufmann, 2006.
[2] T. Yoshida. "SPARC64™ XIfx: Fujitsu's Next Generation Processor for HPC." 2014 IEEE Hot Chips 26 Symposium (HCS), 1-31. IEEE, 2014.
[3] D. Patterson, , T. Anderson, N. Cardwell, R. Fromm, K. Keeton, C. Kozyrakis, R. Tomas, and K. Yelick, ”A Case for Intelligent DRAM.” IEEE Micro, Mar./Apr., 33-44. IEEE 1997.
[4] Hybrid Memory Cube Consortium. "Hybrid Memory Cube Specification 1.0." Last Revision Jan (2013).
[5] P. Merolla, J. Arthur, R. Alvarez-Icaza, A. Cassidy, J. Sawada, F. Akopyan, et. al., “A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface.” Science, 345(6197), 668-673. 2014.
[6] S.-L. Chu, G.-S. Lee, Y.-W. Peng. “Self Similar Cubic: A Novel Interconnection Network for Many-Core Architectures.” In Proc. 2012 Fifth International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), 303-310. IEEE, 2012.
[7] G. Kim, J. Kim, J. Ahn, , J. Kim, “Memory-Centric System Interconnect Design with Hybrid Memory Cubes.” In Proc. 22nd International Conference on Parallel Architectures and Compilation Techniques, 145-156. IEEE, 2012.
[8] H. Zhuge, “Semantic Linking through Spaces for Cyber-Physical-Socio Intelligence: A Methodology.” Artificial Intelligence, 175(5-6), 988-1019. 2011.
[9] W. J. Dally, and B. P. Towles. “Principles and Practices of Interconnection Networks.” Elsevier, 2004.