Data Compression on 2D and 3D Network-on-Chips for CMP

Yuan He,[†] Hiroki Matsutani,[††] Hiroshi Sasaki[††]
and Hiroshi Nakamura[††]

The three-dimensional Network-on-Chip (3D NoC) is an emerging research topic exploring the network architecture of 3D ICs that stack several smaller wafers or dies in order to reduce the wire length and wire delay. Although 3D IC architectures have been extensively studied so far, they have underestimated the negative impacts of vertical interconnects, such as their footprint sizes and the routability degradation. Thus, their vertical bandwidth is still a major concern and it severely degrades system performance. Because such limitations come from the physical design constraints, to mitigate the performance degradation, we have no other choice but to reduce the amount of communication data, especially for those data moving vertically. In this paper, therefore, we carry out a study on data compression in 3D NoC architectures with a comprehensive set of scientific workloads. Firstly, motivated by evaluating data compression on a 2D NoC model with multiple link widths, we find data compression is more effective when narrower links are implemented; and this is the case for 3D IC. Secondly, after applying data compression for three topology settings on 3D NoC, we observe that with more layers and smaller footprint sizes, data compression can improve the performance by up to 13% and reduce the per-packet latency for 12%. Thirdly, with our adaptive compression on 3D NoC, we found some insights on how performance may be affected on 3D NoCs by its communication characteristics. Note that in the case of adaptive compression, an average latency reduction of 4% more than static compression can be observed.

1. Introduction

As semiconductor technology improves, the number of processing cores integrated on a single chip has continually increased. Commercial or prototype chips that have 64 or more processing cores have already been produced[19)20)]. Network-on-Chips (NoCs)[21)] with packet-switched network structures have been widely used instead of traditional bus-based on-chip interconnects to connect many cores.

Recently, the concept of NoCs is being extended to ICs that have three-dimensional structures, namely 3D NoC[22)], in order to mitigate the wire delay and wire energy, which are increasingly posing severe problems to modern VLSI design. Although the wire delay has been mitigated by inserting inverting buffers (i.e., repeaters) on long wires, the buffers themselves add gate delay and consume energy; thus repeater insertion is not a fundamental solution to the problem. In the 3D ICs, a number of wafers or dies are stacked very closely (e.g., 5μm to 50μm); thus a 3D structure significantly reduces wire length, wire delay, and wire energy compared to the same sized 2D structure.

For these reasons, 3D NoC is an emerging research topic, and its network topology[23)], router architecture[24),25)], and routing strategy[26)] have already been extensively studied.

However, many studies on 3D IC architectures have underestimated the negative impacts of vertical interconnects, as reported in 5). Unfortunately, these vertical interconnects, such as through-silicon vias (TSVs) and microbumps, also consume a certain amount of area. In addition, they affect the routability of wires negatively, since some vertical interconnects interfere with metal layers. Thus, although 3D IC technologies are believed a sound technology beyond Moore's Law, their vertical bandwidth is still a major concern. In practice, we find that such vertical bandwidth limitation can severely degrade system performance, by up to 132% (see Section 2).

Because the vertical bandwidth limitations come from the physical design constraints as mentioned above, to mitigate the performance degradation, we have no other choice but to reduce the amount of communication data, especially for those data moving vertically. In this paper, therefore, we carry out a study on data compression in 3D NoC architectures with a comprehensive set of scientific workloads.

The contributions of this paper are the following. Firstly, to the best of our knowledge, this is the first work to characterize and evaluate the effect of data compression on 3D NoCs for CMPs. Secondly, we are the first to introduce and explore adaptive control of data com-

[†] Graduate School of Engineering, The University of Tokyo
[††] Graduate School of Information Science and Technology, The University of Tokyo

pression on 3D NoCs. Thirdly, with our evaluation results, we show that data compression can improve the network latency and system performance by up to 12% and 13%, respectively. Furthermore, with our adaptive compression, we observe a further 4% improvement on network latency for CMPs implemented with 3D NoCs, and we also found that the performance improvement is highly implementation and application specific.

The remainder of this paper is organized as follows. Section 2 briefly surveys 3D IC technologies and introduces the 3D NoC model we focus on. Section 3 discusses the compression technique to be used and our adaptive compression scheme to be investigated. The experimental platform, including the simulation model and workloads, is described in Section 4, while Section 5 is devoted to evaluation results and insights into the effects of data compression on 3-D NoCs for CMP. Section 6 overviews related work, and finally, this work is concluded in Section 7.

2. 3D NoC Design and Its Limitations

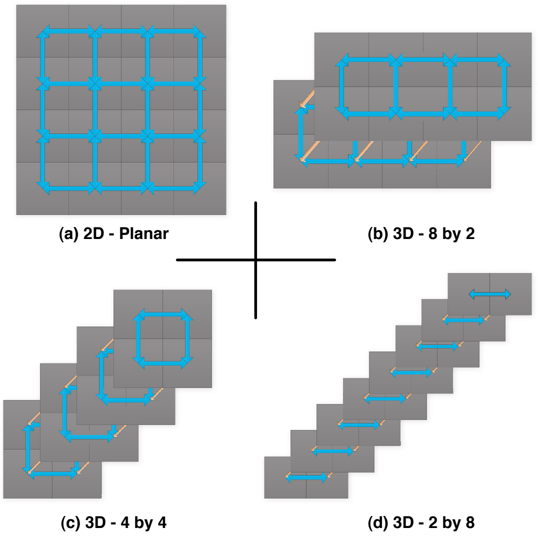


Fig. 1 2D and 3D NoC Tolopologies

3D ICs bring us many benefits like increased system integration, reduced wire length and increased data locality, but how different wafers or dies are stacked vertically remains an open question for the research community and the industry. Various interconnection technologies of 3D ICs have been developed for the purpose of vertical stacking, such as wire-bonding, micro-bump[1),2)] and through-silicon via (TSV)[3),4)].

- `Wire-bonding` is a die-to-die interconnection formed with bonding wires. It has a footprint recorded from 35 to 100 um[4)]. It is the most common approach and has been highly utilized by System-in-Package designs. The limitation is the number of wires and their density as only edges of a chip is used for the purpose of bonding. Obviously, the bonding wire length can be the cause of a considerable communication delay.
- `Micro-bump` forms a die-to-die interconnection through solder balls. It has a footprint known to be from 10 to 100 um[4)]. This approach is generally limited to stack only two dies with face-to-face connections but it can also be used to form the connections of more than two dies with face-to-back design although this is believed inefficient because of factors like heat.
- `Through-silicon via (TSV)` is a wafer-level interconnection making use of via-holes formed through multiple wafers. The footprint of TSV is 5 to 50 um thus it has the potential of offering a better interconnection density than wire-bonding and micro-bump. However, it suffers from high manufacturing cost due to the fact that an extra process to form these interconnects[4)]. Another constraint of TSV comes from routing, as TSV interconnects interfere with gates and wires. So considering yield and cost, the number of TSV interconnects has major impact in design and it should be well thought ahead of manufacturing[5)].

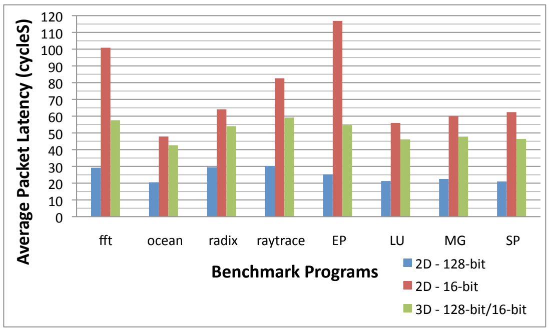


Fig. 2 Latency Degradation with Link Limitations

As briefly explained above, all three interconnection technologies of 3D ICs has a limitation of going vertical, that is, the die-to-die or wafer-to-wafer interconnection can become a bandwidth bottleneck. With larger numbers of such interconnects, we are facing the difficulty of design complexity and cost of manufacturing. To depict this vertical bandwidth limitation, we employ a 3D NoC model with heterogeneous link widths. For links on the same plane (die or wafer), their widths are always the same, but for vertical links which are used to move data between dies or wafers, we model them as having smaller bit width compared to horizontal links. In our study, we also try to capture the effects from different numbers of layers (dies/wafers). We have our 3D NoCs modelling 2, 4 and 8 layers. An example of the baseline 2D NoC and three 3D NoC configurations are illustrated in **Fig. 1**.

Moreover, **Fig. 2** presents an example of how this link limitation can affect the network performance. We see that in the case of ocean, radix, LU, MG and SP, the average packet latency in a 3D NoC having 128-bit of planar links and 16-bit of vertical links are approaching the latency numbers encountered in a 2D NoC with links evenly sized at 16-bit. We also find that under the same configurations, the system performance is being degraded by more than 80% on average for these workloads. Thus, vertical link bandwidth limitation is a major bottleneck for any CMPs considering moving to 3D design.

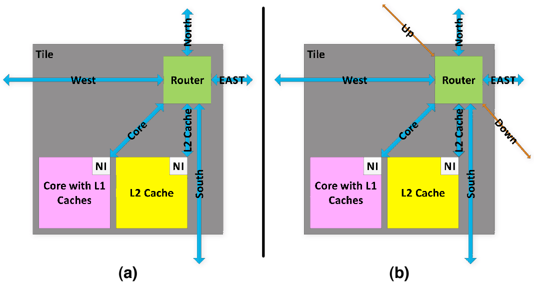


Fig. 3 Tile Design on 2D and 3D NoCs

In our network model, as shown in **Fig. 3**(b), basic building blocks (tile) of our 3D NoCs are connected with each other by routers and links. for comparison purpose, we also include the tile of a 2D design, whose router is at most having six ports and two of them are used to connect to a processor core and an L2 cache bank. For 3D NoCs, two more ports may be added to the router and through two additional links, different dies/wafers are connected. The network routing scheme is also re-defined since X-Y routing for 2D is not sufficient for the 3D design. As explained in the last paragraph, because of the layer-to-layer bandwidth limitation, our 3D NoC models narrower vertical links. More details on configurations of the 3D NoC model are covered in Section 4 where we discuss about the simulation model.

3. Data Compression on NoCs

Data compression is a popular architectural technique and it has been applied in many fields to conserve on-chip/off-chip bandwidth, to enlarge cache/memory capacity or to reduce communication latency. In our work, we use data compression as a solution of bandwidth conservation and latency reduction for 3D NoCs. In this section, we discuss the compression technique in details. We are going to introduce the compression algorithm, frequent pattern compression, at first. After which, we will focus on implementation issues of this compression algorithm. And finally, our proposal of adaptive control on compression will be presented.

3.1 Compression Algorithm and Implementation

There are several state-of-the-art data compression algorithms which has been applied on Network-on-Chips, including frequent pattern compression (FPC)[10] and frequent value compression (FVC)[11],[12]. In this paper, we select frequent pattern compression because of its simplicity and effectiveness. Frequent pattern compression (FPC) is a significance-based compression scheme having small compression/de-compression overheads; unlike frequent value compression (FVC), it also has no synchronization overhead. FPC compresses frequent patterns appeared in data packets. In our case, there are seven such patterns with which we seek to compress each 32-bit of data, the description of these patterns are in 6) and the reason of selecting these patterns is their frequencies. For all seven data patterns, we assign a 3-bit index to each pattern. Along with another index for uncompressed data words, there are in total eight indexes. For example, a data word of 32 zeros will be replaced with an index of 000 after compression, while an 8-bit sign-extended data word will be replaced with an index of 001 plus the 8-bit data. FPC has advantages of high compression ratio and parallel compression. For a 128-bit data packet, we can always split it into 4 parts and compress all parts with four compression circuits at one time. But since FPC employs variable length compression, the de-compression will have to be done in a serial manner.

Regarding the implementation, similar to 10)∼12), data compression/de-compression circuits in our work are assumed to be implemented in network interfaces (NI) of our 3D NoCs. With this design, at NIs, any injecting data traffic will be compressed and receiving data traffic will be recovered; but it is important to note that the enhanced NIs will also have area, latency and energy overheads.

As we mentioned earlier, the compression process of FPC can be done in parallel for several data words at a time. And as stated in 6), this compression process is only taking one cycle per data word, thus with multiple parallel encoders, the timing overhead of compression is one cycle per packet. For de-compression, since FPC is a variable length compression scheme, it is unable to carry out the de-compression in parallel. But as proposed in 10), it is able to overlap the network latency with part of this de-compression latency. In detail, the receiving and de-compression pipeline is designed to work with only a fraction of the packet received. After such a reception of the header flit containing indexes of all compressed words, there is pre-computation process in order to obtain the length of compressed data before its arrival. Hence, the de-compression does not need to rely on receiving the entire compressed packet. By applying this improvement, the de-compression timing overhead can be kept in two cycles per packet.

In 10), it is recorded that with 45 nm process, the area overhead and dynamic power consumption of compressor/de-compressor circuits are 0.183 mm^2 and 0.273 W, respectively. In our paper, power issue is not discussed and is left for future work.

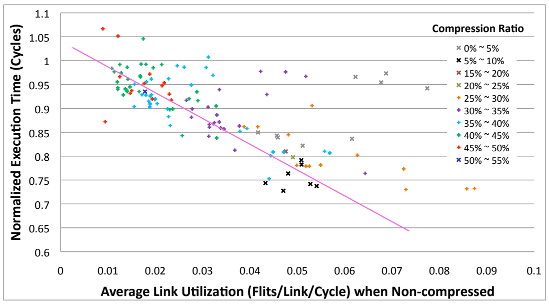


Fig. 4 Efficiency of Data Compression on 2D NoCs with Different Link Widths and Compression Ratios

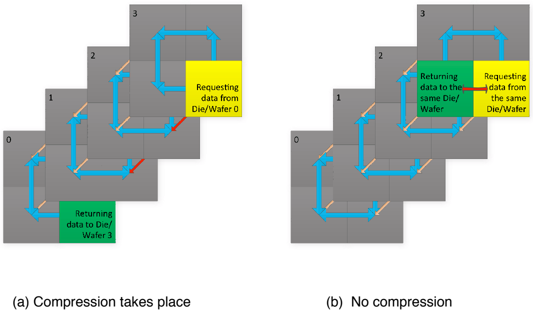
To show the efficiency of this compression algorithm, we have evaluated it on 2D NoCs with 4 different sizes of link width, 16-bit, 32-bit, 64-bit and 128-bit. All other simulation parameters are the same as the 3D model later with the exception of a 6-port router and X-Y routing. Our result is summarized in **Fig. 4** with the data from all benchmarks aggregated in this single graph. In this figure, different colours represent different ranges of compressibility; note that for the same benchmark program, it is possible to have different compression ratios, since smaller flit size maintains less fragmentation, thus having higher Compression ratios. We have two kinds of marker symbols in this figure, "+" is used to mark those data points which are more mission-oriented while "×" represents points which are either outliers or outstanding. For example, all compression ratios lower than 10% come from EP which is less compressible. By plotting the normalized execution time after data compression versus the non-compressed average link utilization, we seek for a stable relationship between the network load and effectiveness of compression. After disregarding those points marked by "×" and fitting a line to all remaining data, we identify a strong link between the network load and compression effectiveness, that is to say, the smaller the flit size (on-chip bandwidth) is, the larger the performance gain. And by following this observation, here comes part of our motivation that we propose to compress vertical going communication.

3.2 Proposed Adaptive Compression on 3D NoCs

In Section 2, we have discussed our 3D NoC model and its vertical bandwidth limitation. To help mitigating this limitation, we present an adaptive compression scheme for CMPs with 3D NoC. Based on FPC, our adaptive compression scheme utilizes compressibility and location based mechanisms to control the compression process. For any data packet waiting to be injected to the network, we have set up two policies to determine whether the compressor should be used or not.

- `Compressibility based control` is very simple, but it requires the compression process. After the actual compression, we can simply identify the size of the compressed packet; if it is known that the compressed packet is bigger than or equal to the size of the original packet, then the network interface disregards the compressed packet and it instead flitisize the original packet. Note that although we can avoid the effect of negative compression on the network and decompressors, we still suffer the compression timing overhead in this case. However, with a mere compression latency of one cycle, it is worth the time to avoid any negative compression.
- `Location based control` is more complex and it targets specifically at the vertical bandwidth limitation of 3D NoCs we model. As shown in Section 2, 3D NoC brings

us smaller package size hence shorter wire length. But data packets travelling across dies/wafers will suffer a limited vertical bandwidth. Thus, we design a new compressor which always considers the locations of source and destination nodes in order to decide if it should compress any packet. More specifically, the location based adaptive compressor in our work compresses any layer-crossing data packets while it ignores any planar traffic like what is shown in **Fig. 5**. The (a) part represents inter-layer communication which incurs compression while inner-layer communication in the (b) part causes no compression.


3
2
1
0
Requesting data from Die/Wafer 0
Returning data to Die/ Wafer 3
3
2
1
0
Returning data to the same Die/ Wafer
Requesting data from the same Die/Wafer

(a) Compression takes place (b) No compression

Fig. 5 Location based control of data compression on 3D NoCs

4. Experimental Platform

In this section, we are going to explain the experimental platform in detail. Firstly, we will quantify the parameters of our simulation model; and secondly, we are going to briefly introduce the workloads tested in our simulation.

For our 3D NoC model, our simulation is carried out for a 16-core SNUCA CMP system with shared L2 cache using the Multifacet GEMS simulator[13] built on Wind River Simics[14]. To correctly simulate data compression and its effect on NoCs, we have modified the detailed network model of GEMS, Garnet[15]. Each core has a pair of dedicated instruction/data L1 caches and the L2 cache is divided into 16 banks. The coherence model of caches includes MOESI protocol with 2 distributed on-chip directories. Directories are used to maintain coherence of memory hierarchies and as memory controllers; in our simulation, directory entry access costs 6 cycles, same as the L2 cache. The router has a fixed 3-stage pipeline and wormhole flow control; the network interface is implemented with a 2-stage pipeline. Compression always consumes one cycle of latency while de-compression takes two cycles.

Table 1 Simulation Configurations

Component	Parameter
Processors	16
L1 Cache	Each core has a total of 64KB of private L1 cache (split I and D), which is 4-way set-associative and has 64 bytes per line and 1 cycle of access latency.
L2 Cache	Shared L2 cache divided into 16 banks. Each bank is 256KB, 16-way set-associative and has 64 bytes per line and 6 cycles of access latency.
Memory	4GB of DRAM with 160 cycles of access latency.
Topology	16 nodes organized in three 3D Mesh topologies, 4 by 2 by 2 layers, 2 by 2 by 4 layers and 2 by 1 by 8 layers.
Router	3-stage pipeline with X-Y-Z routing.
Link	Uneven link width is implemented; the planar link width is 128-bit and the vertical link width is 16-bit.

The simulation parameters also assume each core has 64KB of L1 cache split for instruction and data. Each L2 cache bank is 256 KB. Three 3D topologies are evaluated. One is having eight cores per die and two stacked dies which forms a 4 by 2 by 2 3D Mesh. And the other two are 4 cores stacked as 4 layers and 2 cores stacked as 8 layers, respectively. They form a 2 by 2 by 4 and a 2 by 1 by 8 3D Mesh topologies, one by another. Note that all planar links for 3D NoCs are 128-bit wide and all vertical link are 16-bit wide. Routers in this 3D NoC model employ deterministic X-Y-Z routing and 2 more ports are needed as connections to routers at neighbor dies/wafers if compared with conventional routers of 2D NoCs. For simplicity, configurations are summarized in **Table 1**.

In order to have a diverse performance evaluation, we selected eight workloads from SPLASH-2[16] and NPB 3[17] suites for our simulations with 16-core input.

5. Results

Depending on the on-chip bandwidth requirement and compressibility of a workload, data compression on NoCs can bring several benefits. In this section, we are going to make clear how these benefits look like in practice. We are going to discuss and quantify both network and application centric improvements for 3D NoCs with the two compression schemes we have. Note that for adaptive compression, we applied both rules we proposed in Section 3.

Firstly, we quantify the benefit comes from data compression when it is applied on 3D NoCs for the improvement on the network. We try to discover how much the packet latency is reduced with either static compression or our pro-

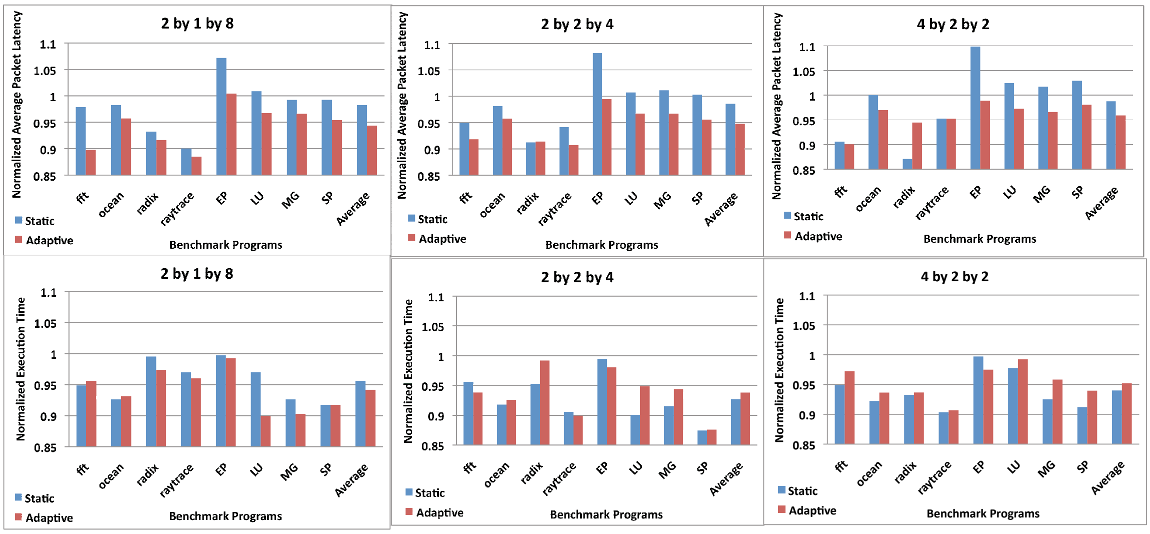


Fig. 6 Latency Reduction and Performance Improvement with Static/Adaptive Compression on 3D NoCs

posed adaptive scheme. Looking at the upper three graphs in **Fig. 6**, we see that the reduction of average packet latency is almost always positive with our proposed adaptive compression scheme. It can be identified that only 1 out of 24 cases exhibits negative reduction while static compression in the same context having 10 out of 24 cases suffering negative effects on latency reduction. Our proposed adaptive scheme outperforms static compression in 22 of 24 cases. It can also be seen that static compression achieves up to 13% of latency reduction with an average of 2%. Overall, our proposed adaptive scheme outperforms traditional static scheme by an additional 4% on average with a maximum of up to 12%. We believe that with more sophisticated adaptive management, this benefit can be even larger.

Secondly, for the system performance, in the lower part of Figure 6, we see more interesting results that the latency reduction we receive from data compression does not well affect the system performance. To explain this, there are a few things we need to consider. Firstly, there is traffic which can affect the network latency more than the system performance, which are cache write-backs. Compressing these write-backs will not directly affect any cache/memory access latency but will enlarge the on-chip bandwidth. Recall our 3D NoC model, it always has the memory controllers implemented on the bottom die/wafer, which means most of the cores and caches have to send write-backs across the dies/wafers, which can cause huge amount of latency but not directly affects the performance.

Looking at ocean in Fig. 6, we found that for all three topologies, static compression outperforms adaptive and in **Fig. 7**, we observe that ocean has a high misses per thousand-instruction with a relatively smaller amount of write-backs per miss. Our explanation is, although small, this write-backs has committed to the latency reduction when being compressed as in the top 3 graphs in Fig. 6 but the high amount of misses only resulted in small amount of vertical going traffic because of its communication pattern is more likely to be die/wafer-wise. This is observed in 27), where the authors found that cores prefer to communicate to their neighbours in terms of exchanging data and cache references.

There are also some more interesting observations. We see that for *EP*, it has a large number of write-backs but a small number of misses; also, its latency reductions through adaptive compression excel static compression more than in all other applications and its performance is always better in terms of adaptive compression. First, this is exactly what we have described in the last paragraphs, where a lot of write-backs result in a good amount of latency reduction. Second, we think through compressing so many write-backs, the bandwidth of the network is highly improved and this indirectly affects the other layer-crossing traffic which are indeed cache and memory accesses, and this in consequence improves the overall performance.

For LU, MG and SP, topology makes a good impact; what we see is, with 8 layers, adaptive compression performs better than static but after evaluating it with 4 layers and 2 layers, this is not true any more. Meantime, we observe very similar amount of write-backs per miss and misses per thousand-instruction for these three workloads.

For fft, it is interesting to see that at 4 layers, adaptive is better but at 2 or 8 layers, static is better. We believe we can find the answer when we look at the core assignments to each layer in more details.

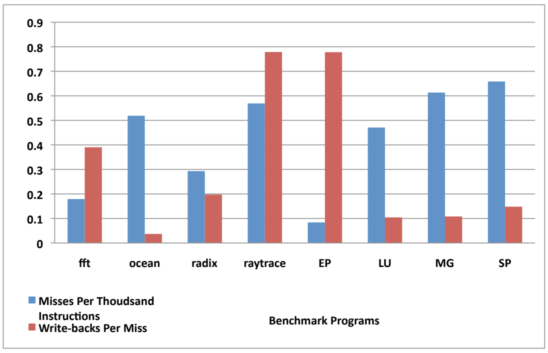


Fig. 7 Cache Performance Characteristics

6. Related Work

In this section, we present a short summary of previous work related to this paper. Data compression for NoCs, as an efficient on-chip optimization, has been extensively studied for 2D design[10)~12)]. In 10), the authors were the first to apply frequent pattern compression on a CMP with Network-on-Chip architecture. Their primary goal was to make a comparison between cache compression and network compression with the same algorithm, in terms of their effects on performance and energy consumption. Both 11) and 12) were about compressing data on NoCs with another candidate algorithm, frequent value compression. Although their results are showing positive feedback, we believe that for any architecture having multiple communicating nodes, frequent value compression can be inefficient because of its overheads of area and synchronization make it hardly scale. In 11), the authors also propose a solution to the area overhead and an adaptive compression control mechanism taking into account the network congestion. It was shown by these three papers that data compression on NoCs can result in performance boost of up to 32% and energy saving of as much as 36%.

Before the study of data compression on NoCs was carried out, there were already many efforts of applying data compression on bus and cache[6)~9)]. Proposed for the purpose of enlarging L2 cache capacity, 6)~8) had proved the chip scale data locality and the efficiency of frequent pattern compression algorithm. It was recorded that for various workloads in evaluation, with FPC on L2 cache, the compressibility can be as much as 60% and the performance improvement can be up to 18%. And in 9), the memory bandwidth demand can be reduced by up to 70% for integer and media workloads. Moreover, a study carried out in 18) had proved that both cache and bus compression are highly efficient in terms of further scaling CMP designs.

7. Conclusions

In this work we have evaluated how data compression affects the network and system performance for CMPs with 3D NoCs. We also presented what difference on performance is made with an adaptive scheme of data compression proposed in the paper. We find that in a bandwidth limited situation like a CMP with 3D NoCs, data compression is always promising in terms of relaxing the bandwidth limitation. We have found that both the static and adaptive compression schemes can improve the performance of the 3D CMP model by up to 13%; and for latency, both approaches imply a reduction of up to 12%. A positive consequence we found is for nearly all cases we have tested, the adaptive compression scheme incurs more latency reduction than the static one; but for performance improvement, the results are very application-specific and implementation-aware. We found that write-back traffic and the physical location of memory controllers are important factors affecting the performance improvement of our proposed adaptive approach. Another observation is, with both compression schemes, performance boost is almost always positive.

References

1) B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. P. Shen, and C. Webb. Die Stacking (3D) Microarchitecture. *Proceedings of the International Symposium on Microarchitecture (MICRO6)*, pp. 469479 (2006).

2) K. Kumagai, C. Yang, S. Goto, T. Ikenaga, Y. Mabuchi, and K. Yoshida. System-in-Silicon Architecture and its Application to H.264/AVC Motion Estimation for 1080HDTV. *Proceedings of the International Solid-State Circuits Conference (ISSCC6)*, pp. 430431 (2006).

3) J. Burns, L. McIlrath, C. Keast, C. Lewis,

A. Loomis, K. Warner, and P. Wyatt. Three-Dimensional Integrated Circuits for Low-Power High-Bandwidth Systems on a Chip. *Proceedings of the International Solid-State Circuits Conference (ISSCC1)*, pp. 268-69 (2001).

4) W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon. Demystifying 3D ICs: The Pros and Cons of Going Vertical. *IEEE Design and Test of Computers*, 22(6):498510 (2005).

5) D. H. Kim, K. Athikulwongse, and S. K. Lim. A Study of Through-Silicon-Via Impact on the 3D Stacked IC Layout. *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD9)*, pp. 674–680 (2009).

6) A. R. Alameldeen, and D. A. Wood. Frequent Pattern Compression: A Significance-Based Compression Scheme for L2 Caches. Technical Report 1500, Computer Sciences Department, University of Wisconsin-Madison (2004).

7) A. R. Alameldeen. Using Compression to Improve Chip Multiprocessor Performance. PhD Thesis, University of Wisconsin at Madison (2006).

8) A. R. Alameldeen, and D. A. Wood. Adaptive Cache Compression for High-Performance Processors. *ACM SIGARCH Computer Architecture News*, 32(2):212–223 (2004).

9) M. Thuresson, L. Spracklen, and P. Stenstrom. Memory-Link Compression Schemes: A Value Locality Perspective. *IEEE Transactions on Computers*, 57(7):916–927 (2008).

10) R. Das, A. K. Mishra, C. Nicopoulos, D. Park, V. Narayanan, R. Iyer, M. S. Yousif, and C. R. Das. Performance and Power Optimization through Data Compression in Network-on-Chip Architectures. *Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA8)*, pp. 215–225 (2008).

11) Y. Jin, K. H. Yum, and E. J. Kim. Adaptive Data Compression for High-Performance Low-Power On-Chip Networks. *Proceedings of the IEEE/ACM International Symposium on Microarchitecture (MICRO8)*, pp. 354–363 (2008).

12) P. Zhou, B. Zhao, Y. Du, Y. Xu, Y. Zhang, J. Yang, and L. Zhao. Frequent Value Compression in Packet-based NoC Architectures. *Proceedings of the Asia and South Pacific Design Automation Conference (ASPDAC9)*, pp. 13–18 (2009).

13) M. M. K. Martin, D. J. Sorin, B. M. Beckmann, M. R. Marty, M. Xu, A. R. Alameldeen, K. E. Moore, M. D. Hill, and D. A. Wood. Multifacets General Execution-driven Multiprocessor Simulator (GEMS) Toolset. *ACM SIGARCH Computer Architecture News*, 33(4):92–99 (2005).

14) P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, and B. Werner. Simics: A Full System Simulation Platform. *IEEE Computer*, 35(2):50-8 (2002).

15) N. Agarwal, L.-S. Peh, and N. Jha. Garnet: A Detailed Interconnection Network Model inside a Full-system Simulation Framework. Technical Report CE-P08-001, Princeton University (2008).

16) J. P. Singh, W. Weber, and A. Gupta. SPLASH: Stanford Parallel Applications for Shared-Memory. *ACM SIGARCH Computer Architecture News*, 20(1):5-4 (1992).

17) H. Jin, M. Frumkin, and J. Yan. The OpenMP Implementation of NAS Parallel Benchmarks and Its Performane. NAS Technical Report NAS-99-011, NASA Advanced Supercomputing (NAS) Division (1999).

18) B. Rogers, A. Krishna, G. Bell, K. Vu, X. Jiang, and Y. Solihin. Scaling the Bandwidth Wall: Challenges in and Avenues for CMP Scaling. *Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA9)*, pp. 371–382 (2009).

19) D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. F. Brown III, and A. Agarwal. On-Chip Interconnection Architecture of the Tile Processor. *IEEE Micro*, 27(5):15–31 (2007).

20) S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar. An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS. *IEEE Journal of Solid-State Circuits*, 43(1):29–41 (2008).

21) L. Benini, and G. De Micheli. *Networks on Chips: Technology And Tools*. Morgan Kaufmann (2006).

22) A. Sheibanyrad, F. Petrot, and A. Janstch. *3D Integration for NoC-Based SoC Architectures*. Springer (2010).

23) V. F. Pavlidis, and E. G. Friedman. 3-D Topologies for Networks-on-Chip. *IEEE Transactions on Very Large Scale Integration Systems*, 15(10):1081–1090 (2007).

24) J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, N. Vijaykrishnan, M. Yousif, and C. Das. A Novel Dimensionally-Decomposed Router for On-Chip Communication in 3D Architectures. *Proceedings of the International Symposium on Computer Architecture (ISCA'07)*, pp. 138–149 (2007).

25) D. Park, S. Eachempati, R. Das, A. K. Mishra, V. Narayanan, Y. Xie, and C. R. Das. MIRA: A Multi-layered On-Chip Interconnect Router Architecture. *Proceedings of the International Symposium on Computer Architecture (ISCA'08)*, pp. 251–261 (2008).

26) R. S. Ramanujam, and B. Lin. Randomized Partially-Minimal Routing on Three-Dimensional Mesh Networks. *IEEE Computer Architecture Letters*, 7(2):37–40 (2008).

27) A Communication Characterisation of SPLASH-2 and PARSEC. N. Barrow-Williams, C. Fensch, and S. Moore. *IEEE International Symposium on Workload Characterization (IISWC'09)*, pp. 86–97 (2009).