High-Performance and Energy-Efficient 3D Manycore GPU Architecture for Accelerating Graph Analytics

DWAIPAYAN CHOUDHURY, ARAVIND SUKUMARAN RAJAM, ANANTH KALYANARAMAN, and PARTHA PRATIM PANDE, Washington State University, Pullman, WA

Recent advances in GPU-based manycore accelerators provide the opportunity to efficiently process large-scale graphs on chip. However, real world graphs have a diverse range of topology and connectivity patterns (e.g., degree distributions) that make the design of input-agnostic hardware architectures a challenge. **Network-on-Chip (NoC)**-based architectures provide a way to overcome this challenge as the architectural topology can be used to approximately model the expected traffic patterns that emerge from graph application workloads. In this paper, we first study the mix of long- and short-range traffic patterns generated on-chip using graph workloads, and subsequently use the findings to adapt the design of an optimal NoC-based architecture. In particular, by leveraging emerging **three-dimensional (3D)** integration technology, we propose design of a **small-world NoC (SWNoC)**-enabled manycore GPU architecture, where the placement of the links connecting the **streaming multiprocessors (SM)** and the **memory controllers (MC)** follow a power-law distribution. The proposed 3D manycore GPU architecture outperforms the traditional planar (2D) counterparts in both performance and energy consumption. Moreover, by adopting a joint performance-thermal optimization strategy, we address the thermal concerns in a 3D design without noticeably compromising the achievable performance. The 3D integration technology is also leveraged to incorporate **Near Data Processing (NDP)** to complement the performance benefits introduced by the SWNoC architecture. As graph applications are inherently memory intensive, off-chip data movement gives rise to latency and energy overheads in the presence of external DRAM. In conventional GPU architectures, as the main memory layer is not integrated with the logic, off-chip data movement negatively impacts overall performance and energy consumption. We demonstrate that NDP significantly reduces the overheads associated with such frequent and irregular memory accesses in graph-based applications. The proposed SWNoC-enabled NDP framework that integrates 3D memory (like Micron's HMC) with a massive number of GPU cores achieves 29.5% performance improvement and 30.03% less energy consumption on average compared to a conventional planar Mesh-based design with external DRAM.

CCS Concepts: • **Computer systems organization → Architectures**; **Other architectures**; **Special purpose systems**;

Additional Key Words and Phrases: GPU manycores, small world NoC, near data processing, graph analytics, irregular applications

This work was supported by the US National Science Foundation (NSF) grant CCF-1815467.

Authors' address: D. Choudhury, A. S. Rajam, A. Kalyanaraman, and P. P. Pande, School of Electrical Engineering and Computer Science, Washington State University PO BOX 642752, Pullman, WA 99164-2752; emails: dwaipayan.choudhury@wsu.edu, a.sukumaranrajam@wsu.edu, ananth@wsu.edu, pande@wsu.edu.

1550-4832/2021/10-ART18 $15.00
https://doi.org/10.1145/3482880

ACM Reference format:
Dwaipayan Choudhury, Aravind Sukumaran Rajam, Ananth Kalyanaraman, and Partha Pratim Pande. 2021. High-Performance and Energy-Efficient 3D Manycore GPU Architecture for Accelerating Graph Analytics. *J. Emerg. Technol. Comput. Syst.* 18, 1, Article 18 (October 2021), 19 pages.
https://doi.org/10.1145/3482880

1 INTRODUCTION

Graph analytics has become a mainstay in many scientific and industrial applications. With advances in high-throughput data generation techniques in various domains, it has now become possible to not just collect raw data but also observe interactions between them at scale. A natural way to represent such relational data is as graphs, where vertices represent the individual entities and edges represent pairwise interactions (or relationships). For instance, in social networks, users are represented as vertices and their interactions represented as edges. Despite their broad application base and the concomitant rise of manycore computing platforms, scalable processing of large-scale graphs on emerging manycore platforms remains a challenge. The large sizes of real-world graphs (with millions of vertices and billions of edges) coupled with irregular graph topologies (scale-free characteristics [1]) create significant challenges associated with data movement that impacts performance. In particular, the power law distributions of vertex degrees (i.e., number of neighbors of a vertex) that characterize scale-free graphs, make it nearly impossible to guarantee locality preservation while storing these graphs in memory. Therefore, even a single update of a vertex that has a large degree could trigger significant on-chip traffic at each step of a graph computation. Furthermore, the mixture of high and low degree vertices could generate an on-chip traffic pattern that is comprised of long- and short-range data movements. GPU-based manycore computing offers a promising direction toward accelerating graph applications. However, such platforms have deep memory hierarchies which could exacerbate the costs in moving data for graph applications [2]. Hence, it is essential to design efficient interconnection networks for manycore chips performing graph analytics. The on-chip memory access patterns associated with graph algorithms give rise to two specific challenges: a) handling the high volume of the memory access requests arise from the data migration, and b) efficiently transferring the long-range on-chip memory access requests that can require multiple hops in a traditional **network-on-chip (NoC).**

In conventional GPU architectures, the main memory is not integrated with the processing (logic) layer and off-chip data movement degrade overall performance and increases energy consumption. In such cases, **Processing-In Memory (PIM)** or **Near Data Processing (NDP)** can present an effective paradigm to reduce data movement overheads by moving the computations closer to the data stored in memory [3, 4]. It enables faster transfer of the data to/from memory to the logic layer. Hence, it reduces both latency and energy consumption. NDP has the ability to take advantage of the emerging 3D-stacked memory and logic devices (such as Micron's **Hybrid Memory Cube** or **HMC**), to enable high-bandwidth, low latency, and low energy memory accesses. In data-intensive applications, a GPU-based manycore architecture with NDP is capable of breaking the barrier between memory access and computational efficiency, but its potential is yet to be adequately demonstrated through careful design and performance evaluation.

The NDP-enabled 3D stacked manycore design needs to be complemented with a suitable NoC architecture that can efficiently handle the long-range and irregular traffic pattern. In this regard, we envision a **small-world (SW) network-enabled 3D NoC (SWNoC)** to interconnect the **streaming multiprocessor (SM)** and **memory controllers (MC)** of a GPU-based manycore platform will be suitable. Given that irregular data movement is the primary factor that limits

graph applications' performance, we posit that reducing the cost and/or the volume of on-chip data movement is critical in scaling up graph applications. Towards this goal, in this work, we present the design of a 3D NoC enabled GPU-based manycore architectures with NDP for accelerating graph analytics. The main contributions of this work are:

(1) We present the design of a new 3D SWNoC architecture that follows a power-law based link distribution as a communication backbone for a GPU-based manycore system. This SWNoC architecture achieves high-performance and energy efficiency for real world graphs with different degree distributions.
(2) The SWNoC is complemented with NDP to further enhance the performance and energy efficiency. The NDP architecture reduces the memory access latency in presence of inherently high cache miss rate.
(3) By adopting a joint performance-thermal optimization, we achieve high performance without creating a thermal bottleneck.

The rest of the paper is organized as follows. Section 2 presents the related work. In Section 3, we discuss the overall architecture designed for graph analytics. Section 4 presents our experimental results and evaluation. Finally, in Section 5, we conclude the paper by summarizing the salient features of this work.

2 RELATED WORK

There is a growing body of work in the implementation of graph algorithms on GPUs. Pannotia [5] is one such collection that implements a diverse range of graph applications for GPU platforms. In CuSha [6] the advantage of **parallel-sliding-window (PSW)** graph representation has been utilized on the GPU to avoid non-coalesced memory access. Mapgraph [7] presents a high-performance parallel graph programming framework where it dynamically selects the strategy to reduce the architectural limits of the GPU. Gunrock [8] allows programmers to implement graph primitives using a high level of abstraction while delivering high performance. As these graph application suites are widely used, designing suitable manycore architectures accelerating these applications is extremely important and that is the focus of this work.

While these works have sufficiently demonstrated the utility of GPU architectures for accelerating graph applications, conventional GPU architectures are predominantly planar (2D) and hence are ill-equipped to efficiently handle long-range and irregular traffic patterns inherent in graph workloads. 3D integration enables performance enhancement of conventional GPU architectures. It is possible to modify the organization of caches and partition these caches into multiple planar layers in a 3D structure to improve the cache hit rate while maintaining low access time [9]. Hardware architectures can also be customized for different vertex-centric applications by inserting application-level data structures and functions [10]. However, due to poor data locality and high memory bandwidth requirement, significant amount of data movements degrade the performance and energy consumption for graph-based applications in conventional architectures with external DRAM. Hence, current graph accelerators choose to focus mainly on optimizing memory accesses. For example, Graphicionado [11] improves memory throughput by replacing random accesses with sequential accesses to scratchpad memory, whereas Tunao [12] devotes a dedicated on-chip buffer to store high degree vertices. The re-configurable architecture in [10] allows us to customize memory to improve temporal and spatial locality. Graph accelerators [14, 15] are designed using the DRAM-based **Hybrid Memory Cube (HMC). Resistive Random-Access Memory (ReRAM)** can be used as memory and also to perform in situ **MAC (multiply-and-accumulate)** operations [16] . ReRAM based graph accelerators have been shown to significantly outperform CPU- or GPU-based systems both in terms of execution time and energy [17–19]. However, it should

be noted that ReRAMs have an unreliable substrate and therefore placing the burden on the algorithms to be error tolerant. It should be noted that existing ReRAM-based graph accelerators assume ideal (error-free) ReRAM behavior, which is not realistic yet.

In contrast, manycore architectures with NDP where memory is closely integrated with logic layers, achieve significant improvement in performance of graph-based applications *without* reliability issues. 3D-stacked memory and logic devices (such as Micron's HMC) enable high-bandwidth, low latency, and low energy memory access [20]. However, conventional 3D architectures are restricted by thermal constraints as temperature impacts both memory retention and overall performance [21, 22]. In [23] the authors demonstrated that proper core placement reduces the peak temperature in a 3D architecture by preventing high power consuming cores from being placed on top of each other. Suitable floor planning [24] and temperature-aware task scheduling [25] are some of the popular techniques for temperature reduction in 3D ICs.

In this paper, we propose a GPU-based manycore architecture where power law-based link distribution in 3D SWNoC enables high performance and energy efficiency for different real-world graphs with varying degree distribution. As graph-based applications are inherently memory intensive, NDP complements the 3D SWNoC in reducing memory access latency and further improves the achievable performance and energy efficiency.

3 OVERALL ARCHITECTURE

3.1 Traffic Characteristics of Graph Workloads

The degree of a vertex in an undirected graph is the number of edges incident on that vertex. The degree distribution of a graph is the distribution of the degrees of all the vertices in the graph. Real-world graphs are known to have a wide range of degree distributions – for instance, the vertices of planar graphs (such as a road network) may have a degree-bounding property thereby leading to a more uniform (or normal) distribution; whereas social networks could have a skewed distribution where only a small fraction of vertices have high degrees. The latter class is typically referred to as "scale-free", and such graphs are expected to follow a power law distribution – where the probability of finding a vertex with degree k is given by $P(k) \propto k^{-\lambda}$, where λ is a constant. The scale-free property of real-world networks was discovered in the late '90s [26] while recent empirical studies [27, 28] have pointed to a wide variety and rich diversity of such scale-free networks.

As examples, Figure 1(a), (b), (c), (d) and (e) show the degree distribution of Web, Facebook, GitHub, Deezer and Road map dataset, respectively. We can observe that the degree distributions for Web, Facebook, GitHub, and Deezer follow power law distribution whereas the Road map follows a normal distribution. For graphs following power law distribution, a small percentage of vertices contribute to the majority of edges. In contrast, most of the vertices in Road map have connectivity close to the average value of the degree. The degree distribution of graph datasets influences the overall on-chip traffic patterns when inputs are mapped on manycore hardware architecture.

In a manycore GPU architecture, data is exchanged between **streaming multi processors (SMs)** and **memory controllers (MCs).** Also, the number of SMs is generally larger than that of MCs. Hence, irregular application workloads such as graph operations are expected to predominantly give rise to many-to-few and/or few-to-many traffic patterns. Henceforth for convenience, we simply refer to this pattern as "many-to-few". This is owing to fact that the SMs process vertices in parallel. Consequently, SMs assigned to process high degree vertices will generate more traffic to MCs than the others. The many-to-few type of data exchange gives rise to long-range traffic patterns as it will be impractical to assume all communicating SMs and MCs can be placed in close proximity on-chip.

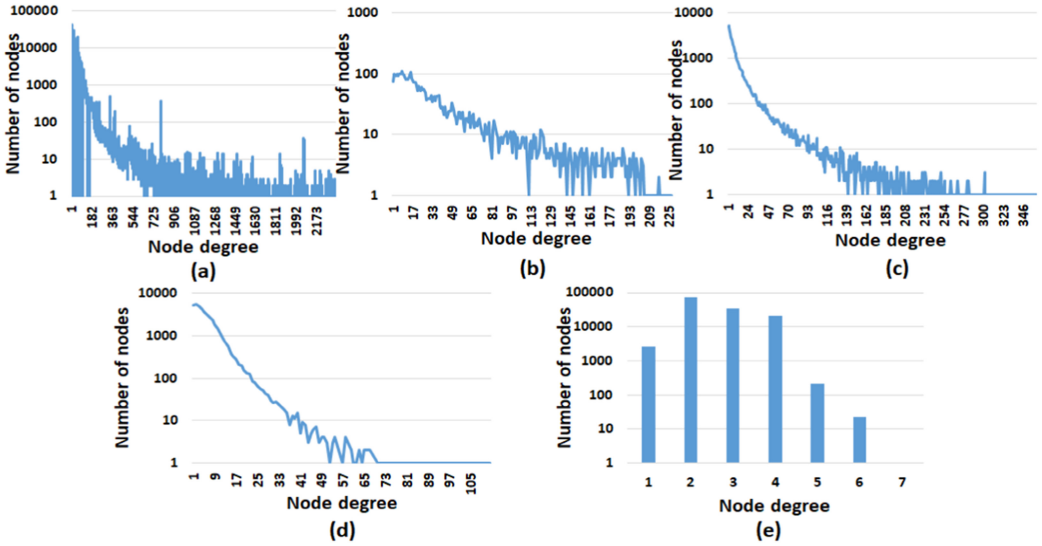

Fig. 1. Degree distribution of (a) Web, (b) Facebook, (c) GitHub, (d) Deezer and (e) Road map datasets.

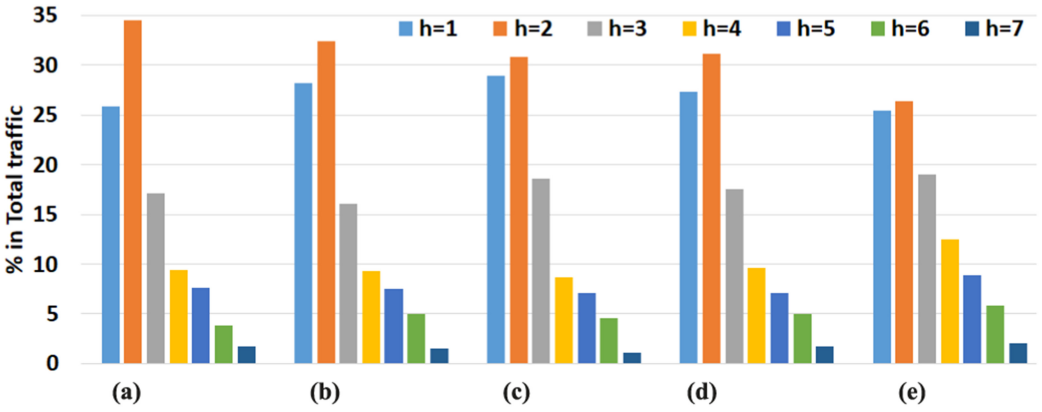

Fig. 2. Hop count distribution in 2D Mesh NoC for PageRank with (a) Web, (b) Facebook, (c) GitHub, (d) Deezer, and (e) Road map (hop counts are given by *h* in the legend).

To analyze the on-chip traffic patterns, we study the hop count distributions of three graph applications: PageRank, graph coloring ("Color"), and **single source shortest path (SSSP)** taken from the Pannotia suite. We consider three different graph datasets, viz., Web [13], Facebook [41], GitHub [43], Deezer [44] and Road map [42], respectively and map the graph applications on to a traditional 2D Mesh NoC architecture. To create the best baseline for 2D Mesh NoC, we optimized the placements of the SMs and MCs to maximize the achievable throughput. Figure 2 shows the traffic distribution using the PageRank application as an example; similar patterns (not shown) were observed with the other two graph applications. It is evident from Figure 2 that for all the datasets, there is a significant amount of traffic exchange between SMs and MCs separated by more than two hops. We refer to the traffic with more than two hops as a "long-range" traffic. The long-range traffic for Web, Facebook, GitHub, Deezer and Road Map account for 37.58%, 38.66%, 40.08%, 41.26%, and 48.96% of their total traffic, respectively. In other words, all the graph inputs

considered here generated a significant amount of long-range traffic. For graphs following power law degree distribution like Web and Facebook the hop count distribution is more skewed (short-range traffic is more than long-range counterpart). For the Road map, which has a normal degree distribution, the traffic is almost evenly distributed between short- and long-range. Analyzing these traffic distributions, we conclude that designing **small world NoC (SWNoC)** is suitable for the graph applications under consideration. It has been already shown that either by inserting long-range shortcuts in a regular Mesh to induce small-world effects or by adopting power-law based small-world connectivity, we can achieve significant performance gain and lower energy dissipation compared to traditional multi-hop Mesh networks [29]. Hence, we advocate that this concept of small-worldness should be adopted while designing the NoC architecture tailor made for graph analytics.

3.2 NoC Optimization

Our objective is to design an NoC based on the hop count distributions in graph application workloads. Hence, in the proposed NoC architecture, SMs and MCs are connected using a small-world interconnection network, where the links are established following the power law distribution. More precisely, if Euclidean distance between core (SM/MC) i and j is d_{ij}, the probability $P(i, j)$ of establishing a link between these two cores is proportional to the distance d_{ij} raised to a finite power. We can represent the probability $P(i, j)$ as follows:

$$P(i,j) = \frac{d_{ij}^{-\alpha}}{\sum_{\forall k} \sum_{\forall l} d_{kl}^{-\alpha}} \tag{1}$$

The parameter α governs the nature of connectivity [32]. A larger α means a locally connected network with a few or even no long-range links. On the other hand, a zero value of α generates an ideal small-world network following the Watts-Strogatz model [30] - one with long-range shortcuts that are virtually independent of the distance between the cores. It has been shown that the average hop count is minimum with a fixed wiring cost for the value α of 1.8 [31]. It should be noted that when a small-world network is implemented in a planar (2D) structure, there will be multiple physically long wires connecting the largely separated cores. Ultimately, this will give rise to high timing and energy overheads. However, when a small-world NoC is implemented using 3D integration, the largely separated cores in a 2D structure can be placed in different planar dies and connected using vertical links. As the vertical links are much smaller in length, the 3D SWNoC reduces the timing and energy costs [32]. Therefore, in this work, we use the manycore GPU architecture interconnected via the 3D SWNoC designed using the power law model of (1) as the computing substrate for the graph applications under consideration. In the 3D SWNoC architecture, we need to consider the placements of the SMs, MCs and the links (both vertical and horizontal) to optimize the overall performance. Due to massive data parallelism, GPU-based manycore architectures are typically optimized to achieve high throughput (*Tput*) [33]. Reducing the average hop count reduces latency, making cores available for more computation and thereby improving the throughput of computation. Additionally, load balancing across the NoC is used to further enhance throughput [34]. Minimizing the standard deviation of hop count will achieve load balancing by reducing the congestion along various paths. Hence, we compare designs (θ) with different core and link placements via the degree of achievable load balancing in the NoC, i.e., using mean $M(\theta)$ and standard deviation $SD(\theta)$ of the hop count, as given by:

$$M(\theta) = \frac{1}{L} * \sum_{i=1}^{C} \sum_{j=1}^{C} h_{ij} \tag{2}$$

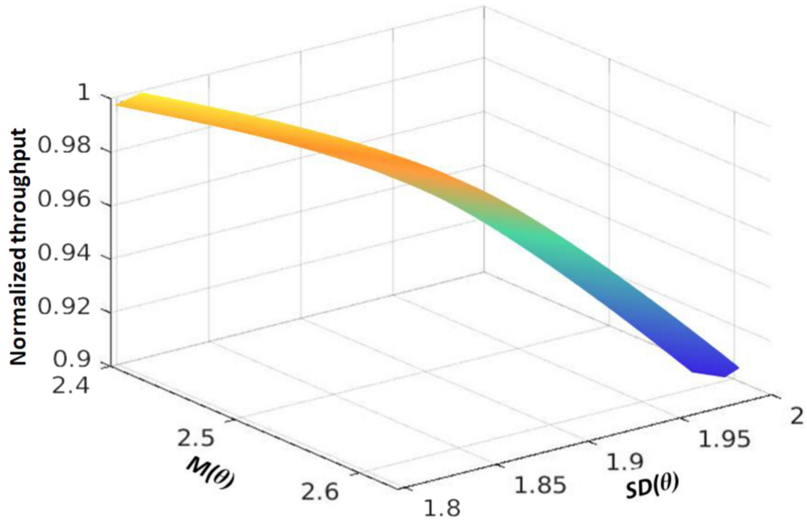

Fig. 3. Normalized throughput with respect to mean ($M(\theta)$) and standard deviation ($SD(\theta)$) for PageRank with Facebook dataset where maximum throughput is considered as one.

$$SD\left(\theta\right) = \sqrt{\frac{1}{L}\sum_{i=1}^{C}\sum_{j=1}^{C}\left(h_{ij} - M\left(\theta\right)\right)^2} \tag{3}$$

where C and L represents the number of cores (SMs+MCs) and the number of links respectively, in the overall architecture, h_{ij} is the number of hops from core i to core j.

Model Validation: In this work, we have modeled maximizing throughput as minimizing $M(\theta)$ and $SD(\theta)$ [34]. We validate this throughput model using detailed cycle accurate NoC simulations. Figure 3 shows the trend of normalized throughput with respect to $M(\theta)$ and $SD(\theta)$ for PageRank with Facebook dataset. A similar trend is observed for all other applications and datasets considered here. It is clear from Figure 3 that throughput is inversely proportional with $M(\theta)$ and $SD(\theta)$. Hence, reducing $M(\theta)$ and $SD(\theta)$ simultaneously leads to increasing the throughput. Therefore, increasing throughput can alternatively be expressed as minimizing $M(\theta)$ and $SD(\theta)$, validating our throughput model.

Overall Multi objective Optimization (MOO) Formulation: As the weighted sum is one of many ways to combine the two parameters: $M(\theta)$ and $SD(\theta)$, we estimate throughput ($Tput$) by combining the effects of both (2) and (3) using weighted-sum function (F). Overall, the throughput of a design θ can be expressed as:

$$Tput\left(\theta\right) = F\left(M\left(\theta\right),\ SD\left(\theta\right)\right) \tag{4}$$

In addition to performance, temperature considerations also become an important design consideration. Consequently, we consider the peak temperature ($Tpeak$) during the optimization of the SWNoC architecture and predict the maximum on-chip temperature of a design θ using the following equation [33].

$$Tpeak\left(\theta\right) = \max_{\mathrm{s,k}}\left\{\sum_{n=1}^{k}\left(P_{s,n}\sum_{y=1}^{n}T_y\right) + T_b\sum_{n=1}^{k}P_{s,n}\right\} * T_H \tag{5}$$

Here, $P_{s,n}$ is the power consumption of a core n planar layers away from the sink in a vertical stack s, T_y is the vertical thermal resistance, T_b is the thermal resistance of the base layer on which the dies are placed, and k represents the k^{th} planar layer where the core is located. We also consider the effects of lateral heat flow T_H, which represents the maximum temperature difference among all layers, to make the prediction accurate. The values of T_y and T_b depend on the material properties. We obtain these values from [34] and calibrate using 3D-ICE simulations [35].

Therefore, the designing of the optimized SWNoC boils down to a **multi-objective** (*Tput* and *Tpeak*) **optimization (MOO)** problem. The multi-objective optimization is aimed to mitigate thermal hotspots while optimizing for throughput. We can represent the MOO-formulation as follows:

$$P^* = \left\{ \theta^* | \ \theta^* \in \ arg \ \max_{\theta} f \left(\frac{Tput\ (\theta)}{Tpeak\ (\theta)} \right) \right\} \tag{6}$$

where, P^* is the set of Pareto optimal designs. We solve this optimization problem by using the popular **simulated annealing (SA)**-based multi-objective optimization heuristic, AMOSA [36] as it is capable of finding out high quality solutions in a reasonable amount of time. We focus on both core and link perturbation in the 3D SWNoC architecture and utilize two moves for solution perturbation namely the swapping of cores ($C(i, j)$) and the new link placement ($L(l)$) as discussed below:

(1) Swapping cores: $C(i, j)$ selects two cores i and j randomly, and swaps their locations. By swapping the placement of two cores, the communication pattern in the network changes and hence, the NoC configuration is changed as well. Consequently, it results in a different numerical value for the parameters mentioned in Equations (4) and (5) before and after the "swapping" function is invoked. It is to be noted that the swapping perturbation function ensures that similar network constraints are maintained before and after the function is invoked.
(2) New Link Placement: $L(l)$ Selects and removes a link of length l between two randomly selected routers. If the link is a planar link, this function creates a new link of the same length between two other routers in the same layer so that overall link distribution remains the same.

The $L(l)$ move preserves the number of length l links, which is required to maintain the power law and the small-worldness of the 3D SWNoC architecture during and after the optimization. Alternating between these two perturbations, we avoid the possibility of getting stuck at a local minimum in the solution space and thereby reaching a (near-) global optimum [47]. In this work we design SWNoC architectures following the two optimization strategies: (a) **Performance optimized (PO):** These architectures have the highest throughput, and (b) **Performance-Thermal joint optimized (PT):** Architectures where throughput is maximized while minimizing the peak temperature. For PO, we maximize the throughput ($Tput(\theta)$) by minimizing the mean $M(\theta)$ and standard deviation $SD(\theta)$ of the hop count simultaneously. Therefore, we choose the design (θ^*_{PO}) for which $Tput(\theta)$ is maximum. Hence, θ^*_{PO} can be expressed as:

$$\theta^*_{PO} = arg \ \max_{\theta} (Tput\ (\theta)) \tag{7}$$

For PT, throughput ($Tput(\theta)$) is maximized while minimizing the peak temperature ($Tpeak(\theta)$). Therefore, we have selected the design (θ^*_{PT}) where, the function of the ratio between $Tput(\theta)$ and $Tpeak(\theta)$ is maximized. We can represent θ^*_{PT} as follows:

$$\theta^*_{PT} = arg \ \max_{\theta} f \left(\frac{Tput\ (\theta)}{Tpeak\ (\theta)} \right) \tag{8}$$

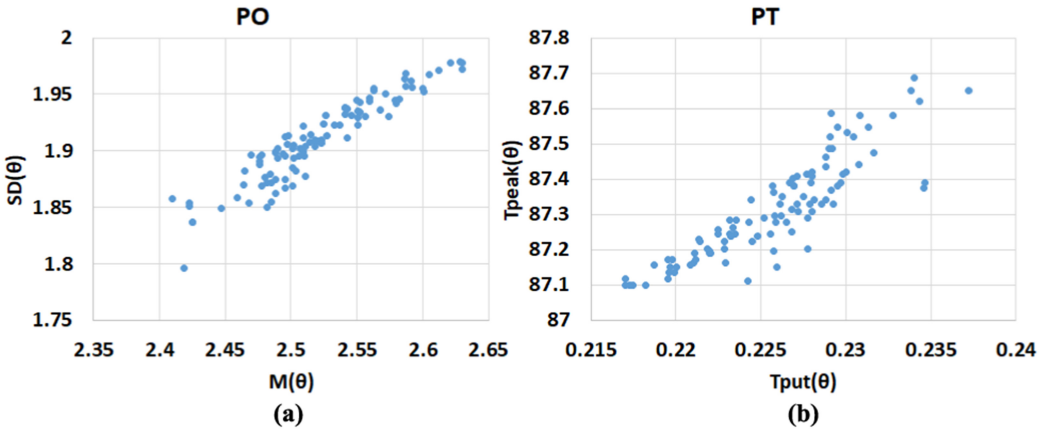

Fig. 4. Illustration of Pareto fronts for (a) PO and (b) PT.

Figure 4(a) and (b) illustrate the Pareto fronts for PO- and PT-based designs. We choose the design (θ^*_{PO}) for PO from the set of Pareto optimal designs (as shown in Figure 4(a)) by following Equation (7). It implies that, for any other design θ, the throughput is lower than that of θ^*_{PO}. Similarly, θ^*_{PT} is selected for PT by following Equation (8) where the function of the ratio between $Tput(\theta)$ and $Tpeak(\theta)$ is maximum among all Pareto optimal designs. Therefore, throughput of θ^*_{PT} is maximum while minimizing the peak temperature.

We have followed the same optimization process for placement of cores (SM/MC) in a 2D Mesh NoC to maximize the achievable throughput. We utilize AMOSA for joint optimization of $M(\theta)$ and $SD(\theta)$. However, we have not included $Tpeak(\theta)$ in the optimization for 2D Mesh as temperature is not a concern in a planar system.

3.3 Necessity for NDP

As graph operations are inherently memory intensive, there is significant amount of off-chip data movement between logic layer and external DRAM. We observe that there is high L2 cache miss rate for the graph applications (86% for PageRank, 83% for Color and 88% for SSSP) considered here. In the presence of high cache miss rate, external DRAM access can quickly become a performance bottleneck, as high data movement increases energy and latency overheads. In this context, 3D **near data processing (NDP)** becomes more desirable as the memory is mounted on top of the 3D stacked logic layers. The NDP paradigm is better suited to take advantage of emerging 3D-stacked memory and logic devices to enable high-bandwidth, low latency and low energy memory access. As the memory is closely integrated with the logic layer, main memory access latency and energy consumption for data movement can be significantly reduced. The NDP-based manycore GPU architecture consists of multiple NDP blocks. Each NDP block has a portion of logic layers and the memory bank connected through TSV-based vertical links. Memory unit in each NDP block is connected to one MC. Here, each block is analogous to an HMC vault [46]. It uses a packet-based interface to transfer data over TSV links. Packet-based memories utilize the NoC by connecting their internal structural elements to enable scalability. HMC interface employs high-speed **serialization/deserialization (SerDes)** circuits which achieve higher raw link bandwidths than traditional, synchronous, bus-based interfaces [46]. However, unlike conventional HMC vaults, we consider GPUs as part of the logic layer in NDP blocks. Each block is connected to one MC, as shown in Figure 5. All the layers are virtually divided into multiple equal 3D cubes and each cube contains equal number of resources (two cores per logic layer and the portion of memory directly

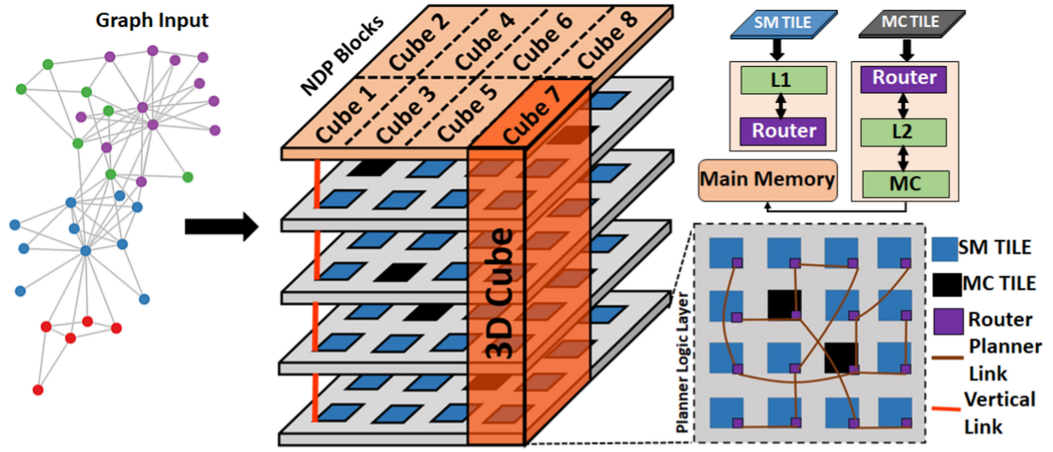

Fig. 5. Proposed Near Data Processing (NDP)-based manycore GPU architecture.

above it). The SWNoC connects different NDP blocks in such a way that the overall throughput of the system is maximized.

4 EXPERIMENTAL RESULTS

4.1 Experimental Setup

Experimental platform: We used GPGPU-Sim [37] to simulate the performance of the proposed NDP based 3D manycore GPU architecture. We use Booksim [38] for implementing the interconnection network to connect the SMs and MCs via different NoC architectures considered in this work. The GPUs are based on the NVIDIA Volta architecture and here we have considered 56 SMs and 8 MCs to give rise to a system with 64 cores where SMs and MCs, both are described as cores. In this work, 2D Mesh is considered as the baseline architecture where 64 cores are arranged in an 8×8 grid pattern. Considering a 20mmx20mm die, the length of each inter-router link is 2.5mm. The overall system runs at the clock frequency of 1.2 GHz. Considering this clock frequency, a 2.5 mm link can be traversed in one cycle. In our proposed 3D SWNoC architecture, 64 cores are equally partitioned into four planar layers. Each layer is of size 10 mm $\times$ 10 mm (considering same area as the 2D system). Within each layer, 16 cores are placed in a 4×4 grid pattern. In the SWNoC architecture, there are planar links longer than 2.5 mm. The longer links are divided into multiple pipelined stages where each stage is of length 2.5 mm. Hence, multiple cycles are necessary to traverse these links. All the vertical links connecting the planar layers are traversed in one cycle. We use the length of each link (in term of cycles) along with core, router and memory characteristics in GPUWattch [39] to determine the overall energy consumption. The width of all links is equal to the flit width of 32 bits. Each core (SM/MC) is connected to one router. Following our previous work, the number of intra-router stages is considered to be three [45]. Dijkstra's algorithm is adopted as the routing algorithm for finding the shortest paths between the cores. Each GPU/SM has a SIMD width of 8 and 64KB of L1 cache. Threads are scheduled to the SIMD pipeline in a group of 32 threads called a warp where all threads in a warp execute the same instruction [37]. For all experiments, we have considered DRAM-based memory. For simulating DRAM, we have used GPGPU-Sim [37]. GPGPU-Sim models GDDR3 DRAM where it employs a **FIFO (First in First out)**-based scheduler. In this work, DRAM is equally divided into eight memory banks and each of them is connected to one MC. The area of an MC is same as that of an SM. Following prior works, the memory is modeled as a multi-layer stack in the proposed NDP design. All layers

Table 1. Characteristics of the Graph Datasets used in our Experiments

Graph	Number of vertices	Number of edges
Facebook [41]	4,039	88,234
GitHub [43]	37,699	289,003
Deezer [44]	41,773	125,826
Road map [42]	129,164	165,435
Web [13]	862,664	38,470,280

in the same vertical stack have equal area i.e., the memory die area is assumed to be same as the logic die area in the proposed 3D NDP architecture. For thermal evaluation, we model the overall architecture in Hotspot based on various parameters e.g., layer thickness, thermal conductivity, etc. as listed in [40].

Graph applications and input datasets: For full-system performance evaluation in this work, we consider Pannotia benchmark [5] as it offers a diverse set of graph applications. For this study, we considered three of these applications, viz. PageRank, Color, and SSSP. These three were chosen due to their varying algorithmic properties. PageRank is a classical case of iterative vertex-centric graph computation where each vertex' state is affected by its neighbors iteratively until convergence. "Color" implements graph coloring, which is also vertex-centric albeit an exemplar of a greedy class of approaches that are typical of independent set problems. **SSSP** is **single source shortest path**, which performs a sweep of the full graph (much like a BFS). Five different datasets as shown in Table 1 were considered:

4.2 Hop Count Distribution for Different NoC Architectures

In this section, we analyze the hop count distribution of 3D SWNoC and compare that with a 2D Mesh-based design when the graph inputs are mapped on to the 64-core system. We obtained the traffic characteristics of graph applications considered here, using full-system simulations on GPGPU-Sim. Due to many-to-few-to-many type of data exchange between cores, the communication between two distantly placed SM and MC gives rise to long range traffic. As mentioned above, the SWNoC was designed and optimized following a power-law based link distribution. Figure 6 shows the hop count distribution of both the 2D Mesh and 3D SWNoC for the PageRank application with all the datasets considered in this work. It is evident that for the 2D Mesh, all the datasets have significant amount of traffic beyond two hops. Noticeably, Mesh has traffic up to seven hops.

We have observed similar characteristics for the other graph applications (Color, SSSP) considered here. Even after placing the cores optimally in 2D Mesh, we have high amount of multi-hop communication from SM to MC or vice-versa. We can see the contrasting situation with the 3D SWNoC architecture, where SWNoC is able to move the hop peaks to the left in Figure 6. As mentioned earlier the SMs and MCs that are separated by long distance on a 2D Mesh can be placed in different planar layers and connected through vertical links. As vertical links are smaller in length compared to their planar counterparts, they can establish single-hop data exchange. Hence, it is evident from Figure 6 that the traffic within two hops has been increased and the amount of long-range traffic has been reduced. We see from Figure 6(a), (b), (c), (d) and (e) that the long-range traffic for SWNoC is 19%, 19.77%, 20.7%, 21.5% and 27.66% of total traffic for Web, Facebook, GitHub, Deezer and Road map respectively – which are significantly less than the long-range traffic observed in 2D Mesh.

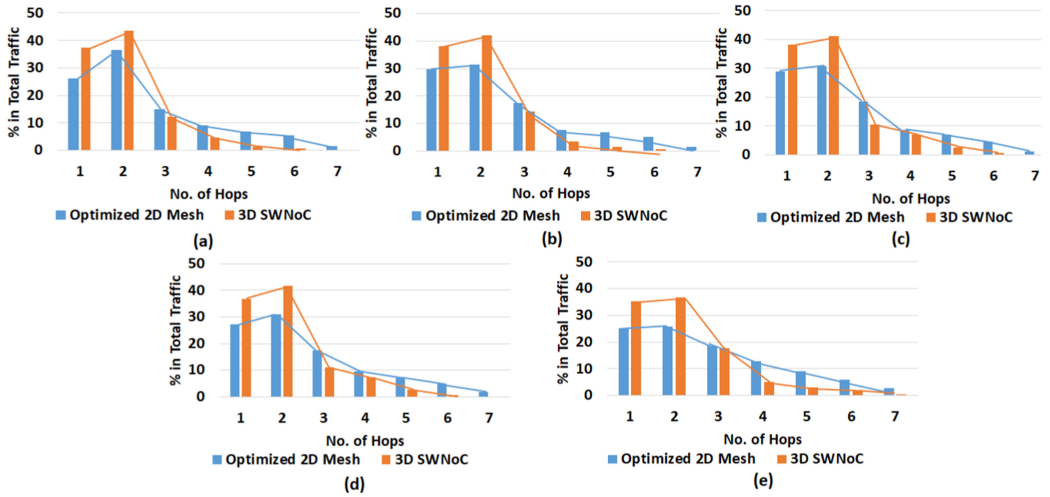

Fig. 6. Hop count distribution of PageRank with (a) Web, (b) Facebook, (c) GitHub, (d) Deezer and (e) Road map for 2D Mesh and 3D SWNoC.

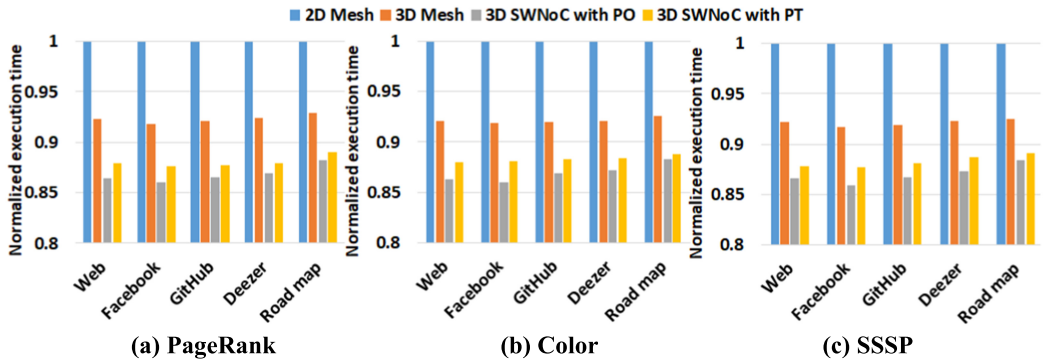

Fig. 7. Normalized execution time of 3D Mesh, 3D SWNoC with PO and PT compared to 2D Mesh for the three test graph applications.

4.3 Performance Evaluation

In Section 4.2, we showed that 3D SWNoC is able to reduce long-range traffic significantly by leveraging vertical links to enhance the overall performance. However, as thermal feasibility is a major concern for any 3D architecture, we need to consider both performance and thermal aspects while designing the 3D SWNoC. Therefore, we design SWNoC architectures following the two optimization strategies mentioned in Equations (4) and (6) above: (a) **Performance optimized (PO):** These architectures have the highest throughput, and (b) **Performance-Thermal joint optimized (PT):** Architectures where throughput is maximized while minimizing the peak temperature. We evaluate the performance of PO and PT using full-system simulations on GPGPU-Sim. Figures 7(a), (b) and (c) illustrate the normalized execution time of PO and PT for PageRank, Color and SSSP, respectively, compared to a 2D Mesh. For completeness, we also show the execution time of a 3D Mesh. We can see from Figure 7 that 3D Mesh achieves 7.8% performance improvement over a 2D Mesh. However, 3D SWNoC improves the overall performance by 13.1% and 12% for PO and PT, respectively, compared to 2D Mesh. This also clearly demonstrates the

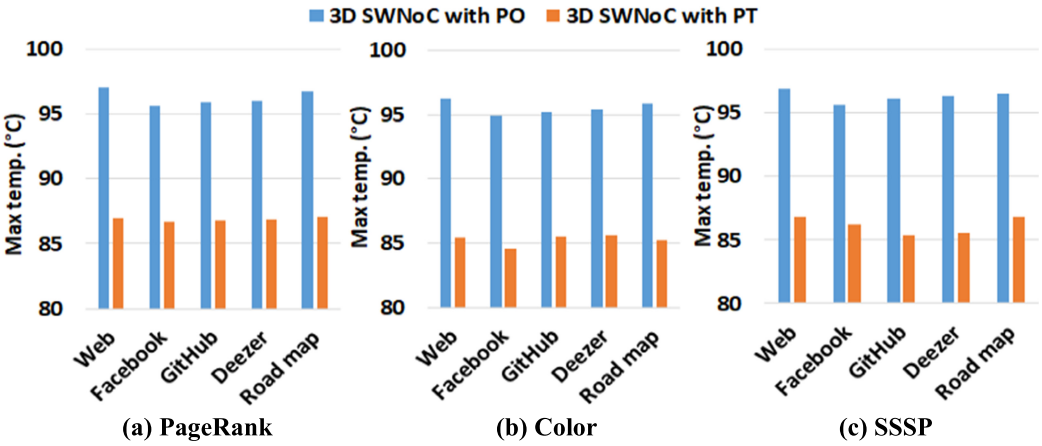

Fig. 8. Maximum temperature of 3D SWNoC with PO and PT for the three test graph applications.

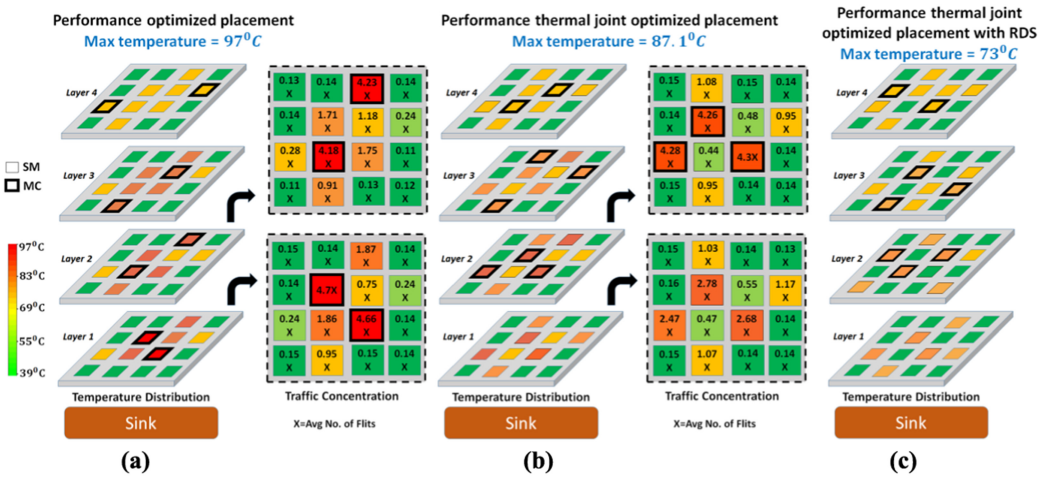

Fig. 9. Temperature distribution of proposed 3D SWNoC with (a) PO, (b) PT and (c) PT+RDS for PageRank with Web dataset.

superiority of 3D SWNoC over a 3D Mesh. Hence, we use the 3D SWNoC as the architecture of choice in this work.

We can see that the performance loss in PT compared to PO is insignificant. However, a detailed thermal analysis will bring out the difference between these two configurations. For thermal evaluation of PO and PT, each logic layer has been modeled following [40] in HotSpot. In the considered 3D architecture, the bottom layer is the closest to the heat sink while the topmost layer is the furthest. Figures 8(a), (b) and (c) show the maximum temperature of PO and PT for PageRank, Color and SSSP, respectively. We can see from Figure 8 that peak temperature of PO for different applications considered here varies from 94.9°C to 97°C. In contrast, the range of peak temperature for PT is 84.6°C to 87.1°C. To investigate the thermal distribution over 3D SWNoC, Figures 9(a) and (b) illustrate the temperature of all the cores for PO and PT, respectively. The temperature profile in Figure 9 is demonstrated for PageRank with Web dataset as an example. We observe the same trend for other applications as well. Figures 9(a) and (b) show that for both PO and PT, the bottom

two layers (layer 1 and layer 2) are generating more temperature than other layers. We can also observe from Figure 9(a) that, for PO, the highest temperatures (97°C and 96.8°C) are generated by two MCs in layer 1. It is well known that, in manycore architectures, thermal hotspots are resulting from high-power densities [40]. On the other hand, the traffic associated with each core contributes to the power consumption. Therefore, temperature is correlated with traffic concentration on the core. Hence, we show the traffic concentrations in the bottom two layers in Figures 9(a) and (b) along with their temperature distributions. The bottom two layers are mainly responsible for creating thermal bottleneck. Due to many-to-few traffic pattern in a GPU manycore architecture, the traffic distribution is skewed. Hence, a few of the cores handle substantially more traffic than the rest. We can see from Figure 9(a) that the MCs generating peak temperature (97°C and 96.8°C) have highest traffic concentration (4.7 times and 4.66 times of the average traffic concentration respectively). In contrast, traffic associated with SMs is varying from 0.13 times to 1.87 times of average traffic concentration, which is significantly smaller than the traffic associated with MCs due to many-to-few traffic. Therefore, placement of core should be done in such a way that traffic associated with MCs is reduced to lower the peak temperature. In Figure 9(b), we see the thermal behavior of PT configuration where both performance and thermal aspects were considered. We see from the comparative study of Figures 9(a) and (b) that traffic associated with MCs is reduced in PT where the highest traffic associated with SMs has been increased from 1.87 to 2.78 times of average traffic in PT compared to PO. However, the highest traffic concentration in SMs is significantly less than MCs for both the cases. Therefore, SMs are not creating temperature bottleneck even after increment of traffic associated with SMs from PO to PT. Moreover, traffic associated with MCs in PT is distributed more evenly than PO so that very high traffic associated with only a few MC is reduced from PO to PT. We can see from Figure 9(b) that the peak temperature is reduced to 87°C. Therefore, PT is able to reduce the peak temperature with insignificant degradation in performance compared to PO due to joint performance-thermal optimization. Hence, we consider PT as more suitable architecture for graph-based applications than PO by considering both thermal and performance aspects.

To evaluate the characteristics of the PT-based design compared to a state-of-the-art technique, we consider the **Reciprocal Design Symmetry (RDS)**-based counterpart [45]. Stacking high power consuming cores directly on top of each other increases the power density and consequently creates thermal hotspots. Therefore, in RDS-based NoC design, two cores those are prone to generate thermal hotspot cannot be placed on top each other. Therefore, we present the performance-thermal trade-off of PT-based design with respect to RDS [45]. For comparative performance evaluation, we incorporate RDS-based design in our PT configuration. We can see from Figure 9(b) that the cores with higher temperature are indicated by red and orange colors. Therefore, along with optimizing the SWNoC based on $Tput(\theta)$ and $Tpeak(\theta)$ (as mentioned in Equations (4) and (5)) in PT, we include RDS by avoiding the placement of red and orange cores on top of each other. We avoid the red-red, orange-orange and red-orange and orange-red configurations. We can see from Figure 9 that RDS is able to reduce the maximum temperature by 14.1°C from PT. To analyze the performance of PT with RDS, Figures 10(a), (b) and (c) illustrate the normalized execution time of PT and PT with RDS for PageRank, Color and SSSP, respectively, compared to a 2D Mesh. We can see from Figure 10 that the performance improvement for PT with RDS is 9.9% on average whereas performance improvement for PT is 12%. Hence, we can conclude that incorporating RDS in PT reduces the maximum temperature significantly with relatively modest degradation in performance. Therefore, PT based design can be further enhanced by incorporating RDS as it achieves better trade-off between performance and temperature.

The other important parameter to consider while designing the 3D SWNoC architecture is the significant amount of off-chip DRAM transaction as the graph applications are inherently memory

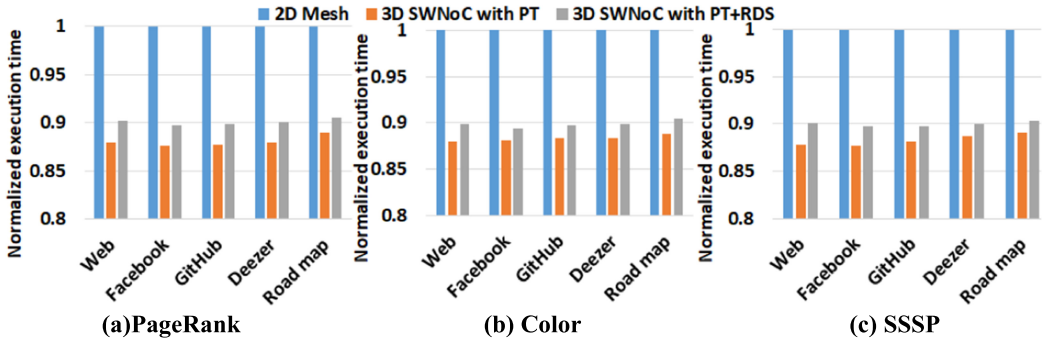

Fig. 10. Normalized execution time of 3D SWNoC with PT and PT+RDS compared to 2D Mesh for the three test graph applications.

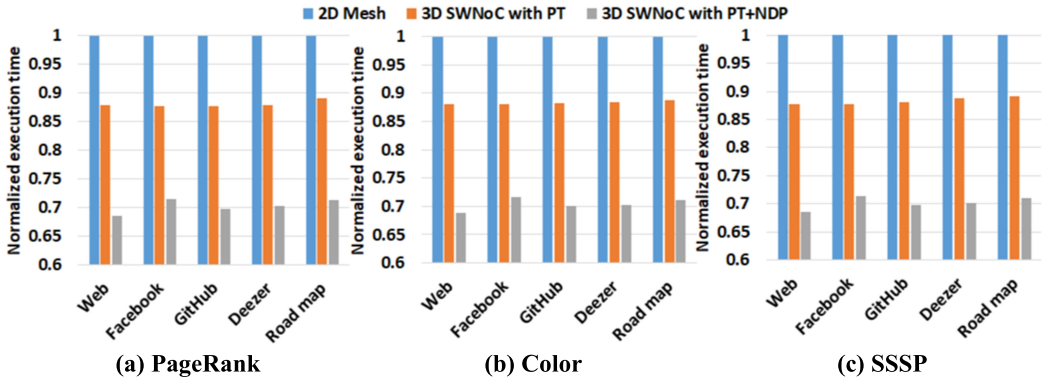

Fig. 11 Normalized execution time of 3D SWNoC with PT and PT+NDP compared to 2D Mesh for the three test graph applications.

intensive. Due to high L2 cache miss rate for the graph applications (86% for PageRank, 83% for Color and 88% for SSSP), accessing external DRAM gives rise to latency and energy overheads. In contrast, by bringing memory unit physically closer to the logic layers, NDP unit reduces memory access latency and energy during L2 cache misses. Figures 11(a), (b) and (c) illustrate the normalized full-system execution time of 3D SWNoC for PT and PT with NDP compared to the baseline (2D Mesh) for PageRank, Color and SSSP respectively. We see from Figure 11 that PT with NDP achieves 29.5% performance improvement compared to the baseline. Along with the performance evaluation, energy consumption of the overall architecture needs to be analyzed. As 3D SWNoC is able to reduce long-range traffic compared to 2D Mesh, energy associated with multi-hop communication is reduced. However, due to the high cache miss rates which are typical in graph based applications, overall energy consumption is dominated by off-chip data movement for external DRAM. Therefore, NDP reduces the energy overhead associated with off-chip data movement by bringing the memory unit physically closer to the 3D logic layers. Figures 12(a), (b) and (c) illustrate the normalized full-system energy consumption of 3D SWNoC for PT & PT with NDP compared to 2D Mesh for PageRank, Color and SSSP, respectively. Careful observation from Figure 12 tells that 3D SWNoC for PT reduces 13.5% whereas PT with NDP reduces 33.03% of overall energy consumption compared to 2D Mesh. Therefore, the savings in energy and execution time prove the effectiveness of NDP for graph-based applications.

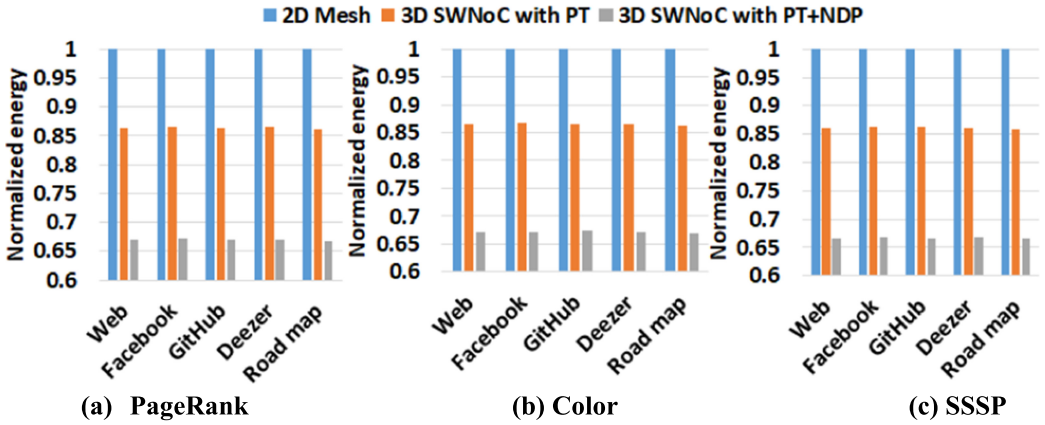

Fig. 12. Normalized energy of 3D SWNoC with PT and PT+NDP compared to 2D Mesh for the three test graph applications.

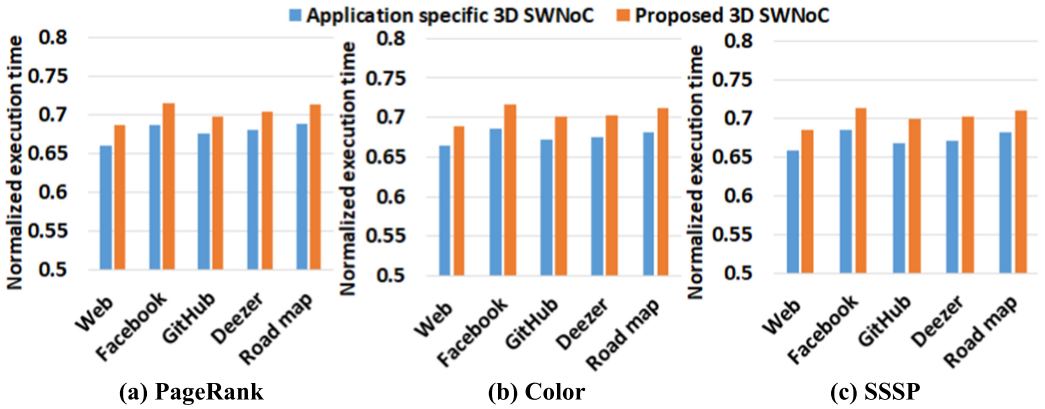

Fig. 13. Normalized execution time of application specific 3D SWNoC and our proposed 3D SWNoC compared to 2D Mesh for the three test graph applications.

It should be noted that, our proposed 3D SWNoC is an application agnostic design where the links are established following the power law distribution. However, we could have designed application specific 3D SWNoC by incorporating individual traffic pattern of different graph applications. Therefore, we design application specific SWNoC architectures by taking into account the frequency of traffic interactions between cores (f_{ij}) in Equation (1). Hence, for application specific 3D SWNoC, the probability $P(i, j)$ of establishing a link between these two cores (i and j) is represented as follows:

$$P(i,j) = \frac{d_{ij}^{-\alpha} f_{ij}}{\sum_{\forall k} \sum_{\forall l} d_{kl}^{-\alpha} f_{kl}} \tag{7}$$

We analyzed the performance of proposed architecture along with the application specific design compared to 2D Mesh. Figures 13(a), (b) and (c) illustrate the normalized execution time of application specific architecture and our proposed architecture compared to 2D Mesh for PageRank, Color and SSSP respectively. Careful observation from Figures 13 shows that there is 2.63% performance degradation on average from application specific architecture to the proposed architecture. Therefore, our proposed 3D architecture is robust to different graph applications.

5 CONCLUSION

In this paper, we propose a 3D manycore GPU architecture design targeted for graph analytics applications. We leverage the benefits introduced by emerging 3D integration to design the proposed manycore architecture that incorporates a **small-world network-based NoC (SWNoC)** to interconnect the SMs and MCs. The 3D SWNoC reduces the long-range traffic by 43.5%–49.7% depending on the data sets considered in this work compared to a 2D Mesh-based design. We complement the advantages introduced by the SWNoC with a joint performance-thermal optimization strategy to place the cores (SMs and MCs) and an NDP framework that integrates 3D memory (like Micron's HMC) with the massive number of GPU cores. The joint optimization strategy preserves the performance benefit introduced by the 3D SWNoC architecture without introducing unnecessary thermal bottlenecks. On the other hand, the NDP framework significantly reduces the performance and energy overheads associated with external DRAM access. Due to high L2 cache miss rate for the graph applications, accessing external DRAM gives rise to latency and energy overhead. In contrast, the NDP unit reduces memory access latency and energy in the presence of high L2 cache miss rate by bringing the memory unit closer to the computation. Putting everything together, the 3D SWNoC-enabled manycore GPU architecture with NDP achieves 29.5% performance improvement and 30.03% less energy consumption compared to the 2D Mesh-based architecture with external DRAM.

REFERENCES

[1] A. L. Barabási and E. Bonabeau. 2003. Scale-free networks. *Scientific American* 288, 5 (2003), 60–69.

[2] K. Duraisamy, H. Lu, P. Pande, and A. Kalyanaraman. 2016. High-performance and energy-efficient network-on-chip architectures for graph analytics. *ACM Transactions on Embedded Computing Systems* 66. DOI: https://doi.org/10.1145/2961027

[3] R. Balasubramonian, J. Chang, T. Manning, J. Moreno, R. Murphy, R. Nair, and S. Swanson. 2014. Near-data processing: Insights from a MICRO-46 workshop. *IEEE Micro.* 34 (2014), 36–42. DOI: https://doi.org/10.1109/MM.2014.55

[4] J. Ahn, S. Yoo, O. Mutlu, and K. Choi. 2015. PIM-enabled instructions: A low-overhead, locality-aware processing-in-memory architecture. In *ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA), Portland, OR.* 336–348. DOI: https://doi.org/10.1145/2749469.2750385

[5] S. Che, B. M. Beckmann, S. K. Reinhardt, and K. Skadron. 2013. Pannotia: Understanding irregular GPGPU graph applications. In *IEEE International Symposium on Workload Characterization (IISWC), Portland, OR.* 185–195. DOI: https://doi.org/10.1109/IISWC.2013.6704684

[6] F. Khorasani, K. Vora, R. Gupta, and L. N. Bhuyan. 2014. CuSha: Vertex-centric graph processing on GPUs. In *Proceedings of the 23rd International Symposium on High-Performance Parallel and Distributed Computing.* 239–252. DOI: https://doi.org/10.1145/2600212.2600227

[7] Z. Fu, M. Personick, and B. Thompson. 2014. MapGraph: A high level API for fast development of high performance graph analytics on GPUs. DOI: https://doi.org/10.1145/2621934.2621936

[8] Y. Wang, A. Davidson, Y. Pan, Y. Wu, A. Riffel, and J. D. Owens. 2016. Gunrock: A high-performance graph processing library on the GPU. *SIGPLAN Not.* 51, 8 (2016), Article 11. DOI: https://doi.org/10.1145/3016078.2851145

[9] A. A. Maashri, G. Sun, X. Dong, V. Narayanan, and Y. Xie. 2009. 3D GPU architecture using cache stacking: Performance, cost, power and thermal analysis. In *IEEE International Conference on Computer Design, Lake Tahoe, CA.* 254–259. DOI: https://doi.org/10.1109/ICCD.2009.5413147

[10] M. M. Ozdal, S. Yesil, T. Kim, A. Ayupov, J. Greth, S. Burns, and O. Ozturk. 2016. Energy efficient architecture for graph analytics accelerators. In *ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA'16), Seoul.* 166–177. DOI: https://doi.org/10.1109/ISCA.2016.24

[11] T. J. Ham, L. Wu, N. Sundaram, N. Satish, and M. Martonosi. 2016. Graphicionado: A high-performance and energy-efficient accelerator for graph analytics. In *49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'16).* DOI: https://doi.org/10.1109/micro.2016.7783759

[12] J. Zhou et al. 2017. TuNao: A high-performance and energy-efficient reconfigurable accelerator for graph processing. In *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID'17).* DOI: https://doi.org/10.1109/ccgrid.2017.114

[13] http://www.sommer.jp/graphs/.

[14] L. Nai, R. Hadidi, J. Sim, H. Kim, P. Kumar, and H. Kim. 2017. GraphPIM: Enabling instruction-level PIM offloading in graph computing frameworks. In *IEEE International Symposium on High Performance Computer Architecture (HPCA'17)*. DOI: https://doi.org/10.1109/hpca.2017.54

[15] M. Zhang et al. 2018. GraphP: Reducing communication for PIM-Based graph processing with efficient data partition. In *IEEE International Symposium on High Performance Computer Architecture (HPCA'18)*. DOI: https://doi.org/10.1109/hpca.2018.00053

[16] D. Fujiki, S. Mahlke, and R. Das. 2017. In-memory data flow processor. In *26th International Conference on Parallel Architectures and Compilation Techniques (PACT'16)*. DOI: https://doi.org/10.1109/pact.2017.53

[17] L. Song, Y. Zhuo, X. Qian, H. Li, and Y. Chen. 2018. GraphR: Accelerating graph processing using ReRAM. In *IEEE International Symposium on High Performance Computer Architecture (HPCA'16)*. DOI: https://doi.org/10.1109/hpca.2018.00052

[18] G. Dai, T. Huang, Y. Wang, H. Yang, and J. Wawrzynek. 2019. GraphSAR. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*. DOI: https://doi.org/10.1145/3287624.3287637

[19] L. Zheng et al. 2020. Spara: An energy-efficient ReRAM-based accelerator for sparse graph analytics applications. In *IEEE International Parallel and Distributed Processing Symposium (IPDPS'20)*. DOI: https://doi.org/10.1109/ipdps47924.2020.00077

[20] E. Azarkhish, D. Rossi, I. Loi, and L. Benini. 2016. Design and evaluation of a processing-in-memory architecture for the smart memory cube. In *Architecture of Computing Systems (ARCS'16)*. 19–31. DOI: https://doi.org/10.1007/978-3-319-30695-7_2

[21] J. Liu, B. Jaiyen, Y. Kim, C. Wilkerson, and O. Mutlu. 2013. An experimental study of data retention behavior in modern DRAM devices. *ACM SIGARCH Computer Architecture News* 41, 3 (2013), 60–71. DOI: https://doi.org/10.1145/2508148.2485928

[22] Y. Zhu, B. Wang, D. Li, and J. Zhao. 2016. Integrated thermal analysis for processing in die-stacking memory. In *Proceedings of the Second International Symposium on Memory Systems*. DOI: https://doi.org/10.1145/2989081.2989093

[23] B. K. Joardar et al. 2017. 3D NoC-enabled heterogeneous manycore architectures for accelerating CNN training: Performance and thermal trade-offs. In *Eleventh IEEE/ACM International Symposium on Networks-on-Chip (NOCS'17), Seoul*. DOI: https://doi.org/10.1145/3130218.3130219

[24] S. M. Alam, R. E. Jones, S. Pozder, and A. Jain. 2009. Die/wafer stacking with reciprocal design symmetry (RDS) for mask reuse in three dimensional (3D) integration technology. In *10th International Symposium on Quality Electronic Design, San Jose, CA*. 569–575. DOI: https://doi.org/10.1109/ISQED.2009.4810357

[25] X. Zhou, Y. Xu, Y. Du, Y. Zhang, and J. Yang. 2008. Thermal management for 3D processors via task scheduling. In *2008 37th International Conference on Parallel Processing, Portland, OR*. 115–122. DOI: https://doi.org/10.1109/ICPP.2008.51

[26] A. Barabási and R. Albert. 1999. Emergence of scaling in random networks. *Disordered Systems and Neural Networks Science* 286 (1999), 509–512. DOI: https://doi.org/10.1126/science.286.5439.509

[27] A. D. Broido and A. Clauset. 2019. Scale-free networks are rare. *Nat Commun* 10 (2019), 1017. DOI: https://doi.org/10.1038/s41467-019-08746-5

[28] P. Holme. 2019. Rare and everywhere: Perspectives on scale-free networks. *Nat Commun* 10 (2019), 1016. DOI: https://doi.org/10.1038/s41467-019-09038-8

[29] U. Ogras and R. Marculescu. 2006. It's a small world after all: NoC performance optimization via long-range link insertion. In *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* Vol. 14. 693–706. DOI: https://doi.org/10.1109/TVLSI.2006.878263

[30] D. J. Watts and S. H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393, 6684 (1998), 440–442. DOI: https://doi.org/10.1038/30918

[31] T. Petermann and P. De Los Rios. 2005. Spatial small-world networks: A wiring cost perspective. arXiv: Cond-mat/0501420v2

[32] S. Das, J. R. Doppa, P. P. Pande, and K. Chakrabarty. 2016. Design-space exploration and optimization of an energy-efficient and reliable 3D small-world network-on-chip. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD'16)*. DOI: https://doi.org/10.1109/TCAD.2016.2604288

[33] A. Arka, B. Joardar, R. Kim, D. Kim, J. Doppa, and P. Pande. 2020. HeM3D: Heterogeneous manycore architecture based on monolithic 3D vertical integration. *ACM Transactions on Design Automation of Electronic Systems*. DOI: https://doi.org/10.1145/3424239

[34] B. K. Joardar et al. 2019. Learning-based application-agnostic 3D NoC design for heterogeneous manycore systems. *IEEE Transactions on Computers* 68, 6 (2019), 852–866. DOI: https://doi.org/10.1109/TC.2018.2889053

[35] J. Cong, J. Wei, and Y. Zhang, 2004. A thermal-driven floorplanning algorithm for 3D ICs. In *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers (ICCAD'04)*. 306–313. DOI: https://doi.org/10.1109/ICCAD.2004.1382591

[36] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb. 2008. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation* 12, 3 (2008), 269–283. DOI: https://doi.org/10.1109/TEVC.2007.900837

[37] A. Bakhoda, G. L. Yuan, W. W. L. Fung, H. Wong, and T. M. Aamodt. 2009. Analyzing CUDA workloads using a detailed GPU simulator. In *IEEE International Symposium on Performance Analysis of Systems and Software, Boston, MA*. 163–174. DOI: https://doi.org/10.1109/ISPASS.2009.4919648

[38] Nan Jiang et al. 2013. A detailed and flexible cycle-accurate network-on-chip simulator. In *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS'13) Austin, TX*. 86–96. DOI: https://doi.org/10.1109/ISPASS.2013.6557149

[39] J. Leng, T. Hetherington, A. ElTantawy, S. Gilani, N. S. Kim, T. M. Aamodt, and V. J. Reddi. 2013. GPUWattch: Enabling energy optimizations in GPGPUs. *SIGARCH Comput. Archit. News* 41, 3 (2013), 487–498. DOI: https://doi.org/10.1145/2508148.2485964

[40] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, and D. Atienza. 2010. 3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling. In *Proceedings of ICCAD, San Jose, CA*. 463–470.

[41] http://snap.stanford.edu/data/egonets-Facebook.html.

[42] http://networkrepository.com/road-usroads.php.

[43] http://snap.stanford.edu/data/github-social.html.

[44] http://snap.stanford.edu/data/gemsec-Deezer.html.

[45] D. Lee, S. Das and P. Pande. 2019. Analyzing power-thermal-performance trade-offs in a high-performance 3D NoC architecture. *Integration* 65 (2019), 282–292.

[46] R. Hadidi et al. 2018. Performance implications of NoCs on 3D-stacked memories: Insights from the hybrid memory cube. In *IEEE ISPASS, Belfast*. 99–108

[47] D. Lee, S. Das, J. R. Doppa, P. Pande, and K. Chakrabarty. 2018. Performance and thermal tradeoffs for energy-efficient monolithic 3D network-on-chip. *ACM Trans. Des. Autom. Electron. Syst.* 23, 5, Article 60 (2018), 25 pages.

[48] T. Bui, C. Heigham, C. Jones, and T. Leighton. 1989. Improving the performance of the Kernighan-Lin and simulated annealing graph bisection algorithms. In *Proceedings of the 26th ACM/IEEE Design Automation Conference (DAC'89)*. ACM, New York, 775–778.

Received April 2020; revised May 2021; accepted August 2021