The Journal of Supercomputing (2020) 76:9558–9584
https://doi.org/10.1007/s11227-020-03216-y

# Optimal low-latency network topologies for cluster performance enhancement

**Yuefan Deng[1] · Meng Guo[2] · Alexandre F. Ramos[3,4] · Xiaolong Huang[1] · Zhipeng Xu[1,5] · Weifeng Liu[6]**

Published online: 2 March 2020


**Abstract**
We propose that clusters interconnected with network topologies having minimal mean path length will increase their processing speeds. We approach our heuristic by constructing clusters of up to 32 nodes having torus, ring, Chvatal, Wagner, Bidiakis and optimal topology for minimal mean path length and by simulating the performance of 256 nodes clusters with the same network topologies. The optimal (or near-optimal) low-latency network topologies are found by minimizing the mean path length of regular graphs. The selected topologies are benchmarked using ping-pong messaging, the MPI collective communications and the standard parallel applications including effective bandwidth, FFTE, Graph 500 and NAS parallel benchmarks. We established strong correlations between the clusters' performances and the network topologies, especially the mean path lengths, for a wide range of applications. In communication-intensive benchmarks, optimal graphs enabled network topologies with multifold performance enhancement in comparison with mainstream graphs. It is striking that mere adjustment of the network topology suffices to reclaim performance from the same computing hardware.

**Keywords** Network topology · Graph theory · Latency · Benchmarks

## 1 Introduction

The ever increasing processing speeds of supercomputers—culminating at IBM Summit [6] with its peak speed of 201 PFlops and 2,414,592 cores—brings exascale era within reach by systems and applications developers. For achieving the milestone of exascale computing, the developers must reduce power consumption and

**Electronic supplementary material** The online version of this article (https://doi.org/10.1007/s11227-020-03216-y) contains supplementary material, which is available to authorized users.

✉ Alexandre F. Ramos
alex.ramos@usp.br

Extended author information available on the last page of the article

increase processing speeds by means of, e.g., design of power-efficient processors (and other components) capable of delivering higher local performance and design of networks capable of delivering low-latency and high-bandwidth communications. Those goals have been incrementally achieved, e.g., the ratio of performance to power consumption of IBM Summit is greater than that of TaihuLight; IBM Summit's faster processing speed is reached with a smaller number of cores; comparison of June 2018 and November 2018 Top 500 lists [6] shows Sierra machine surpassing TaihuLight with a new High-Performance Linpack (HPL) result. Performance increase, however, cannot rely only on raising individual processors clock speed because of the power wall of the Moore's law [60]. Consequently, the number of interconnected processors will keep increasing along with the impact of network topologies on the supercomputers' sustained (maintained average) processing speed, a deed raising the necessity of providing architects with consistent tools for the discovery and design of optimal networks. To attend that, theoretical insights [30, 64] for describing, designing, analyzing and optimizing the next-generation interconnection networks to increase global processing speeds of supercomputers may become a major tool for the HPC community.

In this manuscript, we approach the problem of enhancing a cluster's performance using symmetric minimal latency network topologies supported by a new framework for designing regular graphs of degree $k$ with rotational symmetry and minimal mean path length. The graphs support the network topologies of the directly connected clusters that we benchmarked. The optimal graphs enabled building a cluster which may outperforms a torus of the same degree by a factor of up to 3. Our graphs of degree 3 can achieve the same performance of the torus of degree 4—a clear reduction in hardware costs, engineering complexity, and power consumption. Our results showing the favorable impact of optimal graphs on a cluster's performance open a new avenue of theoretical and experimental research for supercomputer architects. Related work is discussed in Sect. 2, and Sect. 3 presents our algorithm for designing a network topology and the cluster that we used on our analysis. Section 4 presents and examines graph properties supporting different clusters designs and their benchmark results. Concluding remarks are presented in Sect. 5.

## 2 Related work

We present a discussion on the potential use of our approach and on how it complements existing technologies for network topologies for supercomputers and data centers. Despite active theoretical investigations on network design for clusters [52, 54], the use of advanced topologies in actual machines has not been a priority since the early days of parallel computing [43] because of potential engineering complications and lack of a measure of performance gains. Network topologies are the main elements affecting supercomputer interconnection network performance, and for decades, meshes [28], tori of 3D through 6D [8–10, 18, 25, 44, 66], hypercubes of various dimensions [32, 40, 42], fat trees [37, 51, 53] and off-the-shelf Ethernet or adapted InfiniBand [45] switched fabrics have been the mainstream network subsystems. Mesh topologies which are based on lattice graphs, tori resulting from graph product of rings and hypercubes as binary $n$-cubes represent the direct interconnection network [28], while fat trees (folded Clos) belong to multistage indirect networks which consist of multiple layers of switches [28, 51].

In general, the system architecture aims at providing maximal connectivity, scalable performance, minimal engineering complexity and least monetary cost [30]. An ideal network of a fixed node degree must satisfy performance requirements including small network diameter, broad bisection width, simple symmetric topology, engineering feasibility, and modular, expandable design [30]. For example, mesh topology has low node degree and engineering complexity, but its large network diameter and average distance dampen node-to-node communications; fat tree by its multi-level switches realizes the maximum bisection width but with large diameter; the torus and its derivative $k$-ary $n$-cube [26] have lower node degree, relatively smaller diameter and average distance. Hybrid 6D mesh/torus TOFU interconnect is incorporated in K computer [9], while modified 3D torus with combined 2-node is designed to form the Cray Gemini interconnect [10], upgrading from the traditional 3D torus topology as in Cray SeaStar [18, 66], IBM Blue Gene/L [8] and Blue Gene/P [44], and 5D torus is applied in IBM Blue Gene/Q [44]. Other variants of torus such as the SRT [46] and RDT [75] networks, variant of $k$-ary $n$-cube such as the Express Cubes [27] and interlaced bypass torus (iBT) [76, 77] use the technique of adding bypass links. Modifications of fat tree [38, 41] have also been carried out to reduce its complexity and cost. Recently, high-radix hierarchical topologies such as Dragonfly [48] on which Aries interconnect [33] is based have been studied and implemented. Slim Fly [16] among the high-radix topologies also proposed to minimize mean path length but is limited by the fixed combination of its radixes and sizes. However, a classification of the graphs enabling minimal mean path length is only on its infancy [7, 39]. To the best of our knowledge, there are only a few network topologies aiming at minimizing mean path length that have been thoroughly researched and even less have been deployed and benchmarked in supercomputers architecture.

On the other hand, use of data centers for cloud computing has been rapidly increasing and challenges architects to build machines of which the amounts of processing nodes, memory and switches grow steadily while keeping the machine operational. That poses scalability and fault detection, along with maximal bisection bandwidth, as key features of data center networks (DCNs). Instead of reaching that by addition of switch layers, recent advances propose the use of optical networks of switches to replace top-of-rack aggregation switches [55]. That approach may be complemented by ours by constructing optimal networks of switches with reduced latency. In that case, there will be two optimization procedures, for minimizing mean path length (MPL) and labeling pairs of communicating optical channels, which will enable the small network of switches to perform optimally under constraint of a finite numbers of ports. Symmetry of our optimal network topologies enables low levels of engineering complexity, as exemplified by our prototype machines.

## 3 Discovery of optimal network topologies and cluster description

We aim to investigate the increase in the processing speeds of a cluster by optimizing its average latency accordingly with its network topology. Hence, we propose a new algorithm to discover minimal MPL symmetric graphs to support optimal low-latency network topologies for clusters and test experimentally our proposition on a directly connected cluster.

### 3.1 Discovery of optimal network topologies

To obtain optimal network topologies, we search for $N$-vertex degree-$k$ regular graphs, denoted by ($N$, $k$), with minimal mean path length (MPL). Cerf et al. [23] first calculated the lower bound of MPL for any regular graph and discovered small degree-3 graphs with up to 24 vertices whose MPL is minimal [24]. Additionally, it was proved that the diameters of such optimal graphs are also minimal. The exhaustive computer search of an optimal graph of fixed size and degree is computationally expensive, e.g., the number of non-isomorphic 32-vertex degree-3 regular graphs, labeled as (32,3), is $\sim 10^{13}$ [21]. Thus, heuristic methods have been developed using greedy local search [49], simulated annealing [68], or theoretical graph product and construction [59] for reduced search duration.

For the graphs reported in this manuscript, we implemented the graph parallel exhaustive search using the enumeration algorithms *snarkhunter* [19, 20] and *genreg* [58], with built-in split option for parallelization and girth (the length of the smallest cycle in the graph) option as constraint. Optimal graphs having large girths [24] help reduce the search space, e.g., a reduction from $\sim 10^{13}$ non-isomorphic (32,3) regular graphs (with no girth constraint) to $\sim 10^{5}$ by a constraint of girth 7 [21]. This method was used for finding the (32,3)-Optimal graph. However, the exhaustive search of graphs with more vertices or higher degree has astronomical duration even under girth constraint.

To find larger optimal graphs with higher degree, we used random iteration of Hamiltonian graphs (i.e., graphs having a closed cycle that visits each node only once called Hamiltonian cycle) [17] with rotational symmetry. By this method, we have discovered the (32,4)-Optimal graph. It is worth mentioning that the final layout of the (32,3)-Optimal graph is also 90° rotationally symmetric after the MPL optimization search. For each optimal graph, we reorder the vertices on the ring according to its different Hamiltonian cycles and look for more rotational symmetries among these isomorphic layouts. The coloring of the edges helps to visualize this symmetric design. Fixing such symmetric structure is also one way to reduce the search space, which we also apply to the optimization of larger-scale topologies.

### 3.2 Cluster description

To perform our experiments, we constructed a switchless Beowulf cluster named “Taishan” that has up to 32 nodes (Fig. 1). Each node has eight communication ports, with two of them used for cluster management and storage. Hence, we can

**Fig. 1** Taishan Beowulf cluster

evaluate performances of clusters with network topologies supported by graphs of degrees 2 to 6 and benchmark the impact of network topology on processing speeds. Because of hardware homogeneity, we conclude that our results on the impact of network topology remain valid when cutting edge technology is used.

Because of budget limits, we use a low-end hardware to build a functional prototype suited for investigating the impact of the network topology on the cluster's processing speeds. Moreover, we use such a configuration to focus on the role of the network on the cluster's performance while expecting to minimize additional influences. Each node of Taishan has 1 Intel Celeron 1037U dual-core processor (1.80 GHz, 2M Cache), 1 × 8 GB DDR3 SODIMM (1600 MHz, 1.35V), 128 GB SSD and eight Intel 82583V Gigabit Ethernet controllers (PCIe v.1.1, 2.5 GT/s). We use CentOS Linux 6.7 (kernel 2.6.32) as operational system and NFS for sharing files through one of the ports that is connected to a 48-port Gigabit Ethernet switch. Processes communicate directly through node's ports interconnected accordingly with the supporting graph adjacency rules. We use GCC version 4.4.7 and MPICH 3.2 for compiling and running our parallel programs. Static routing is used accordingly with Floyd's algorithm [34] to ensure the shortest path length and lowest congestion.

## 4 Analysis of graph properties and cluster benchmarks

### 4.1 Comparative analysis of optimal network topologies

In order to evaluate the effects of the optimal network topologies on the cluster performance, we have designed several network topologies using regular graphs

$(N, k)$ with $N = 16, 32$ and $k = 2, 3, 4$. The topologies of the benchmarked clusters of 16 nodes are ring (R), Wagner (W) [17], Bidiakis (B) [72], $4 \times 4$ torus (T) (4D hypercube) and two optimal graphs (O) re-discovered by our parallel exhaustive search. The 32 vertices clusters used the ring, Wagner, Bidiakis, $4 \times 8$ torus, Chvatal (C) [17] and the two optimal graphs obtained by our parallel exhaustive and random search. The adjacency matrices of all the benchmarked topologies are included in additional Online Resource. We also compute the bisection width (BW) of each topology using the KaHIP program, which efficiently achieves a balanced partition of a graph [65]. We refer to each cluster as $(N, k) - X$, where $X$ is the 1st letter of, or the name of the supporting graph. The evaluated network topologies and respective graph properties are presented in Table 1, while Fig. 2 shows the graphs (left) and their corresponding latency versus hop distance plots (right) obtained by actual ping-pong messaging tests. In all graphs of Fig. 2, the solid black disks denote average values for the latency and hop distance, while the error bars' lengths are obtained from the standard deviation. The dashed black line indicates the fit of the ping-pong latency, denoted by $T$, as a linear function of the hop distance $h$, namely $T = T_0 + \alpha \cdot h$ where $T_0$ is the network initiating time and $\alpha$ is the slope. We denote the Pearson correlation coefficient [36] between the ping-pong latency and the hop distance by $\rho$ and compute it as

$$\rho = \frac{\sum_{i,j=1}^{N} (T_{i,j} - \langle T \rangle)(h_{i,j} - \langle h \rangle)}{N(N-1)\, \sigma_T\, \sigma_h}, \quad i \neq j,$$

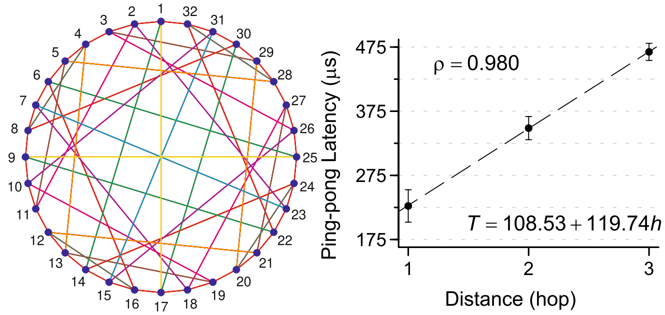
where $T_{i,j}$ and $h_{i,j}$ are the ping-pong latency and the hop distance between nodes $i$ and $j$. The average ping-pong latency and average hop distance (MPL) are given by, respectively,

$$\langle T \rangle = \frac{\sum_{i,j=1}^{N} T_{i,j}}{N(N-1)}, \quad \text{and} \quad \langle h \rangle = \frac{\sum_{i,j=1}^{N} h_{i,j}}{N(N-1)}, \quad i \neq j,$$

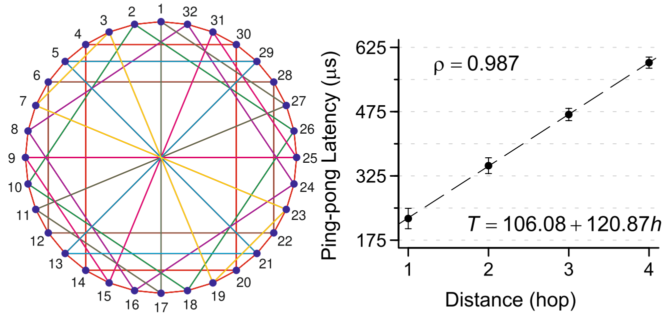
while their corresponding standard deviations are given by

$$\sigma_T = \sqrt{\frac{\sum_{i,j=1}^{N} (T_{i,j} - \langle T \rangle)^2}{N(N-1)}}, \quad \text{and} \quad \sigma_h = \sqrt{\frac{\sum_{i,j=1}^{N} (h_{i,j} - \langle h \rangle)^2}{N(N-1)}}, \quad i \neq j.$$

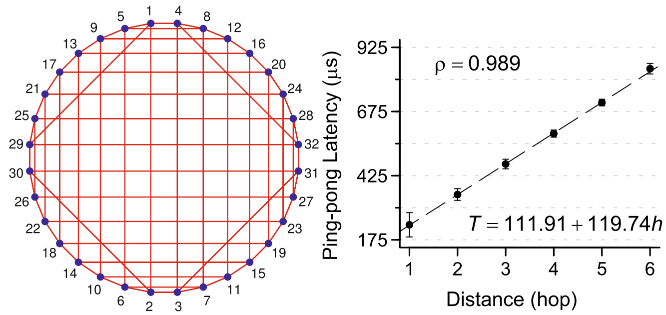
Table 1 shows the diameters ($D$), mean path length (MPL) and bisection width (BW) of the graphs supporting the benchmarked networks. Properties of optimal graphs are emphasized with bold fonts. For all $(N, k)$ graphs, the optimal topology has minimal MPL and D and maximal BW. Hence, we expect that the optimal graphs will support a network topology of low latency, because of shorter MPL and D (see ping-pong test results in Fig. 2), and high throughput, because of larger BW [28]. Indeed, results present in the next section lead to similar conclusions despite the influence of communication patterns, internal algorithms, message sizes, memory access, and routing.



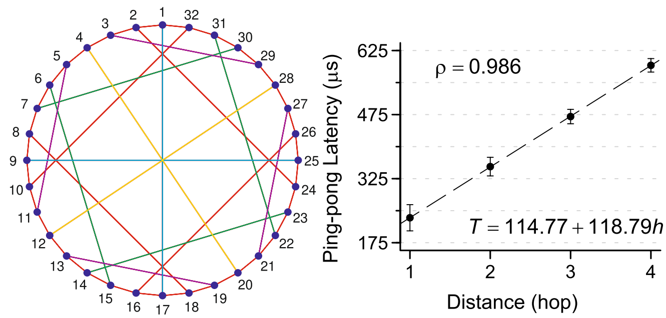

**(a)** (32,4)-Optimal




**(b)** (32,4)-Chvatal

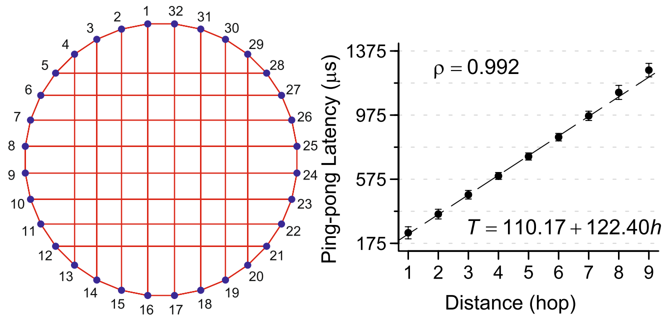



**(c)** (32,4)-Torus

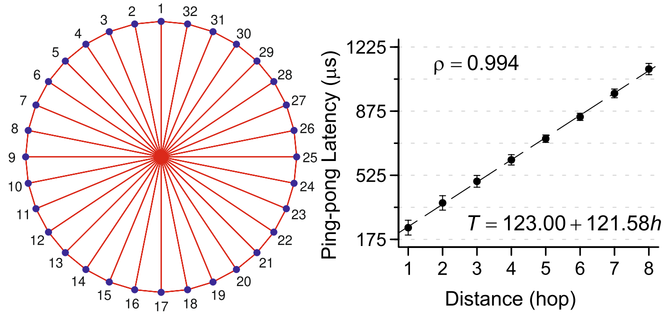



**(d)** (32,3)-Optimal

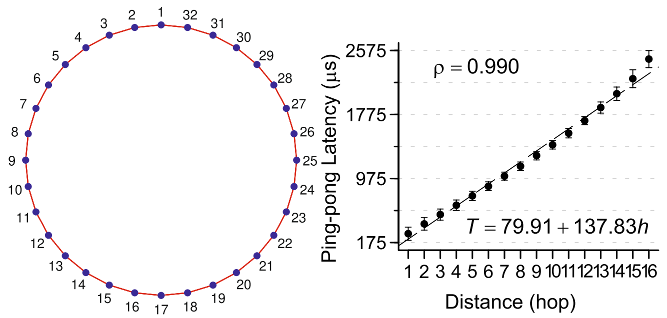
**Fig. 2** Benchmarked topologies (left) and their node-to-node ping-pong latency versus hop distance (right)



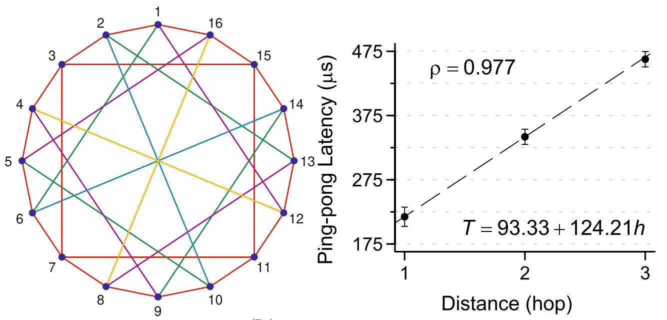

**(e)** (32,3)-Bidiakis




**(f)** (32,3)-Wagner

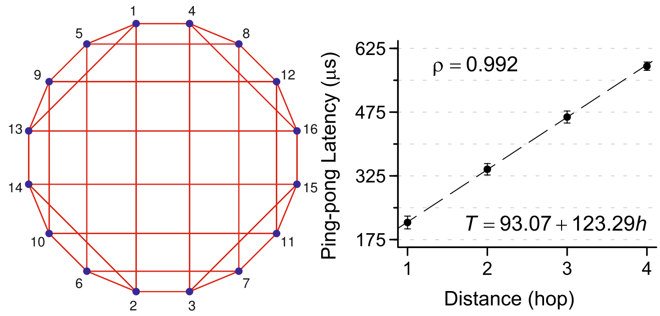



**(g)** (32,2)-Ring

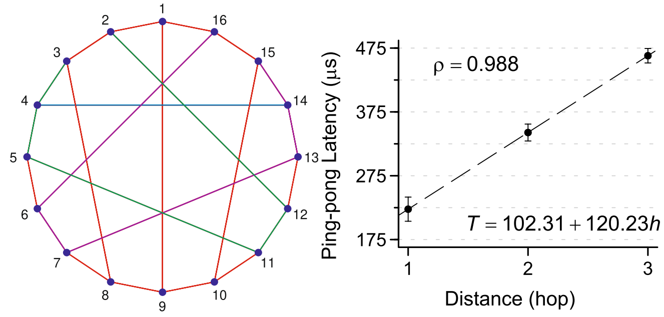



**(h)** (16,4)-Optimal

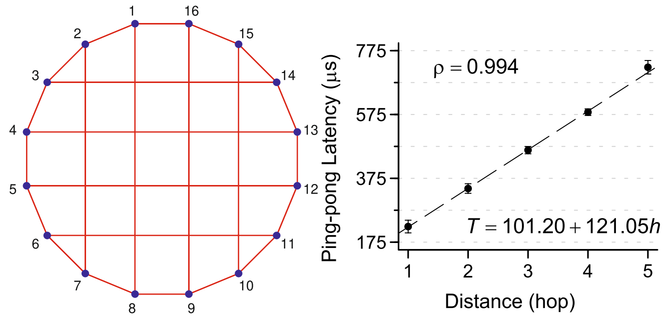
**Fig. 2** (continued)



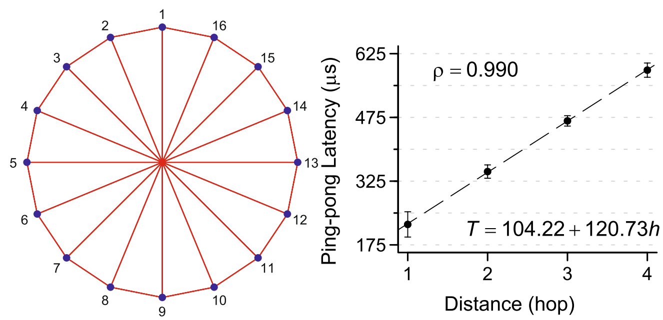

**(i)** (16,4)-Torus




**(j)** (16,3)-Optimal

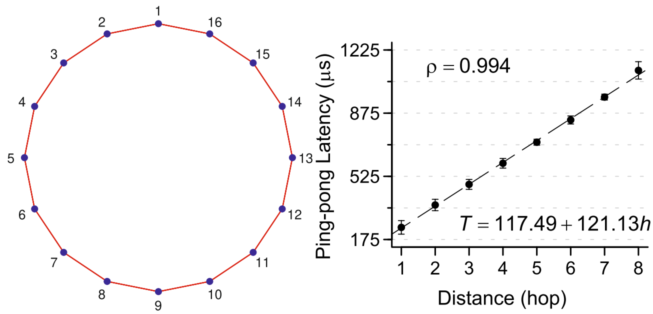



**(k)** (16,3)-Bidiakis




**(l)** (16,3)-Wagner

**Fig. 2** (continued)




**(m)** (16,2)-Ring

**Fig. 2** (continued)

**Table 1** Graph properties of benchmarked topologies

| Topology | D | MPL | BW | Topology | D | MPL | BW |
|---|---|---|---|---|---|---|---|
| **(16,4)-Optimal** | **3** | **1.75** | **12** | **(32,4)-Optimal** | **3** | **2.35** | **16** |
| | | | | (32,4)-Chvatal | 4 | 2.55 | 8 |
| (16,4)-Torus | 4 | 2.13 | 8 | (32,4)-Torus | 6 | 3.10 | 8 |
| **(16,3)-Optimal** | **3** | **2.20** | **6** | **(32,3)-Optimal** | **4** | **2.94** | **10** |
| (16,3)-Bidiakis | 5 | 2.53 | 4 | (32,3)-Bidiakis | 9 | 4.06 | 4 |
| (16,3)-Wagner | 4 | 2.60 | 4 | (32,3)-Wagner | 8 | 4.61 | 4 |
| (16,2)-Ring | 8 | 4.27 | 2 | (32,2)-Ring | 16 | 8.26 | 2 |

## 4.2 Benchmark results and analysis

The following representative benchmark programs were used to evaluate the cluster's performance: custom ping-pong and MPI collective communications; effective bandwidth (b_eff) [1, 50]; FFTE [2, 70]; Graph 500 [3, 61]; and the NAS Parallel Benchmarks (NPB) [5, 12]. Ping-pong tests report runtime and, for each topology, produce a node-to-node latency matrix used to show correlation with supporting graph's hop distances (Fig. 2). Here, benchmark runtime refers to the elapsed wall clock time for a benchmark to be completed. The evaluation of remaining benchmarks is done by means of the ratio of the sustained processing speed of a given topology to that of the ring of the same size. Since the effective bandwidth and Graph 500 benchmarks report average speed $S$ while the other benchmarks report average runtime $T$, the performance ratio of each topology to its corresponding ring is $S/S_{\text{ring}}$ or equivalently $T_{\text{ring}}/T$. The values of $S$ and $T$ are averages obtained after multiple executions of each benchmark. In particular, for ping-pong, MPI collective communications, effective bandwidth and Graph 500, the calculation method of average runtime or speed is specified in their respective sections. Our analysis generates scatter plots of the performance ratio at $y$-axis versus the topology's MPL at $x$-axis for each benchmark, as shown in Figs. 3, 4, 5, 6, 7, 8 and 9. Error bars are calculated by repeated experiments (except ping-pong and effective bandwidth). Red

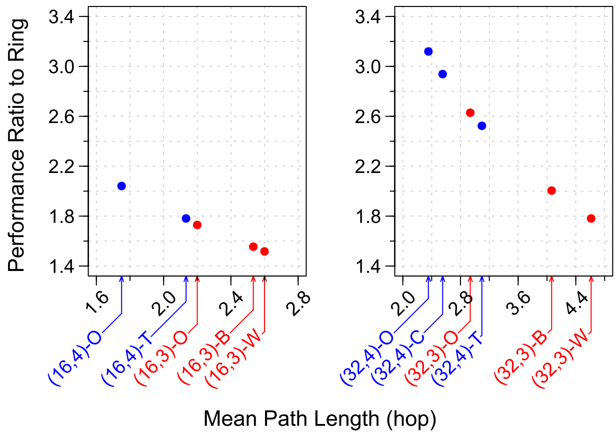


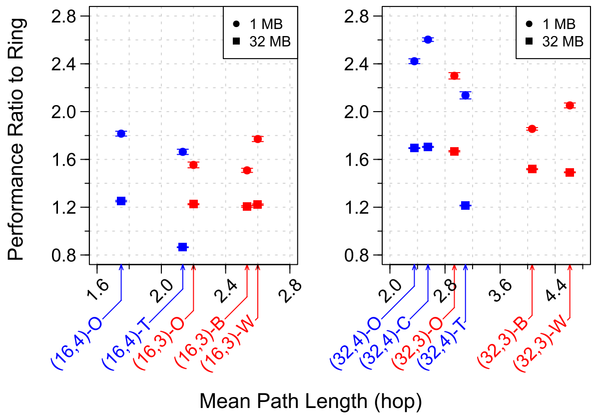

**Fig. 3** Performance ratios on ping-pong tests

(or blue) points indicate the data for degree-3 (or 4) clusters. Different data points' symbols represent different sub-tests of one application.

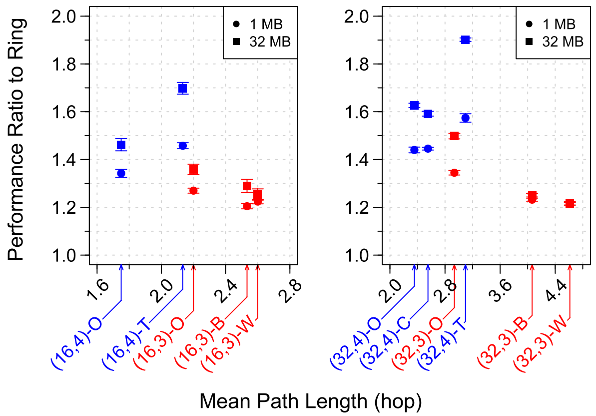
### 4.2.1 Ping-pong test

The routing algorithm and communication properties of the cluster in comparison with the supporting graph path lengths are evaluated by means of the ping-pong test designed using MPI_Send and MPI_Recv, with message sizes ranging from 1 byte to $2^{13}$ bytes (8 KB). Latency is measured as the average round-trip time for a message to travel between source and destination over multiple runs. We select 1 KB as the message size to output the corresponding node-to-node latency in the form of a matrix. The Pearson correlation and linear regression between node-to-node latency and hop distance were calculated for each topology as in Fig. 2, while performance ratios of average latency between all pairs of nodes for each topology are plotted in Fig. 3.

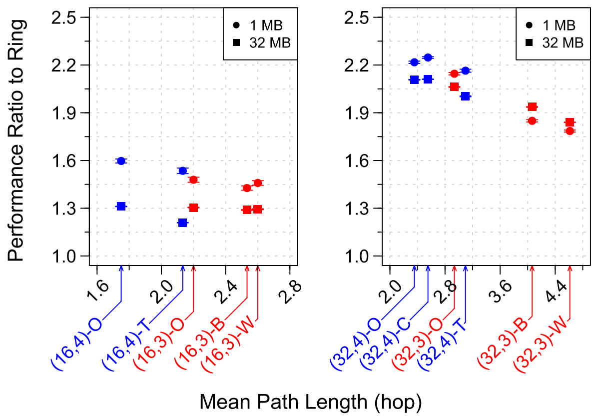
Figure 2 shows that the Pearson correlation coefficients ($\rho$) between ping-pong latency and hop distance under the shortest-path routing are all greater than 0.977. Such a strong correlation is reflected on the approximately linear dependence between node-to-node latency in the network and graph's distance (hop) as indicated by the dashed line. Notice that besides (32,2)-Ring the fitting equations describing the linear relation are very similar, independently of the cluster's sizes and topologies, the average of which being $T = 107.17 + 121.15h$. (Because of the high diameter of (32,2)-Ring, message traverse and serialization start to affect the latency for long-distance transfer.) Moreover, performance of ping-pong for different topologies is strongly inversely proportional to their MPL as shown in Fig. 3. Those results also hold for larger messages of sizes up to 8 KB.




**(a)** MPI_Bcast

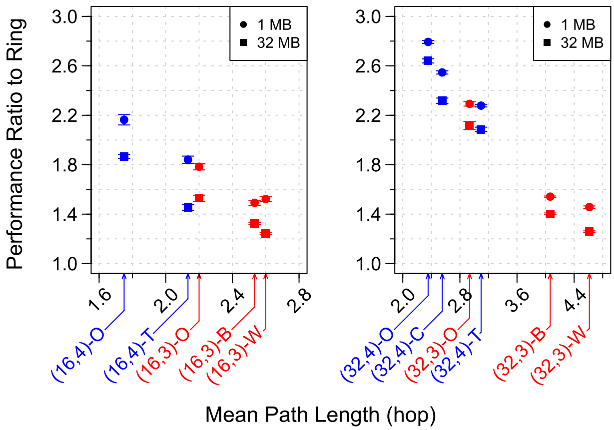



**(b)** MPI_Reduce




**(c)** MPI_Scatter

**Fig. 4** Performance ratios on collective communications




**(d)** MPI_Alltoall

**Fig. 4** (continued)

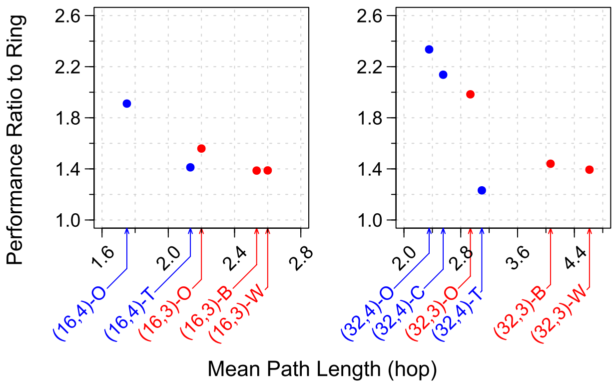



**Fig. 5** Performance ratios on effective bandwidth

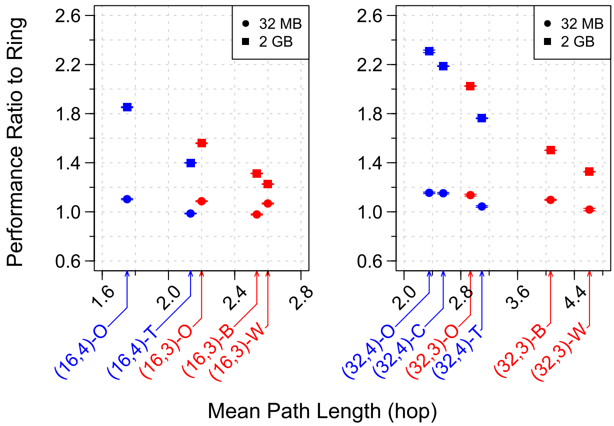



**Fig. 6** Performance ratios on 1D FFTE

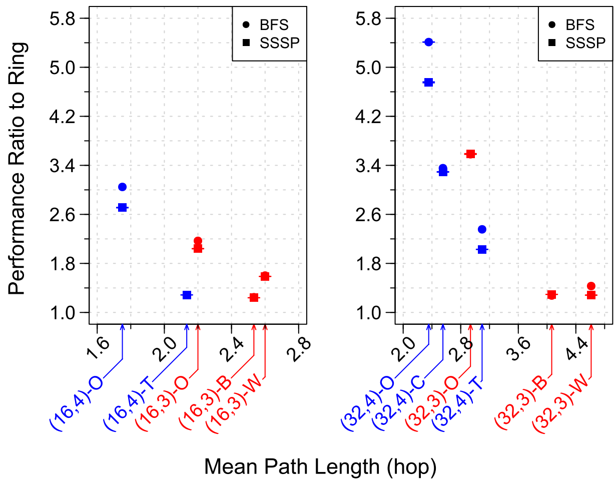


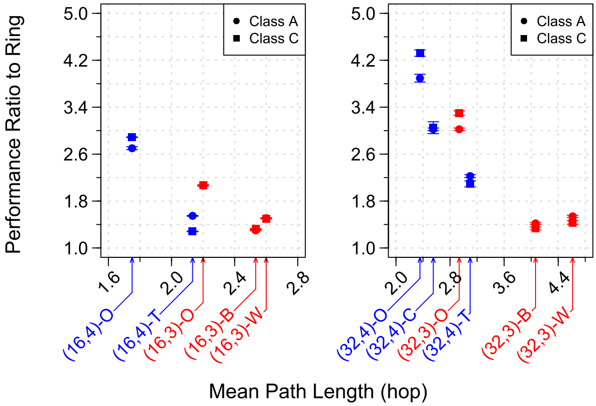

**Fig. 7** Performance ratios on Graph 500

### 4.2.2 Collective communications

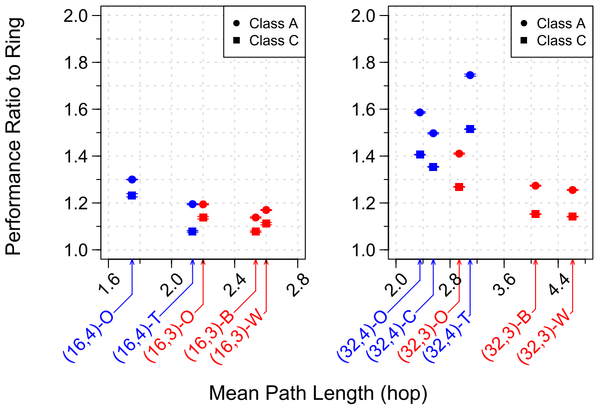
Collective communications benchmarks test the performance of MPI_Bcast, MPI_Reduce (with reduce operation MPI_SUM), MPI_Scatter and MPI_Alltoall. We choose unit messages of 1 MB and 32 MB under the constraint of 8 GB RAM available per node. On each node, the transfer message sizes are either equal to the unit message sizes or the unit sizes multiplied by the number of nodes, depending on whether it is the root node and on the MPI collective function.

MPI_Bcast, MPI_Reduce and MPI_Scatter were run multiple times with all nodes being root multiple times. Then, we average the runtime over all root nodes and then over all tests. The runtime of each test is the maximum elapsed wall clock time on all nodes. For MPI_Alltoall, we conduct the test multiple times and average the runtime over all tests. The runtime of each test is the average elapsed wall clock time on all nodes.

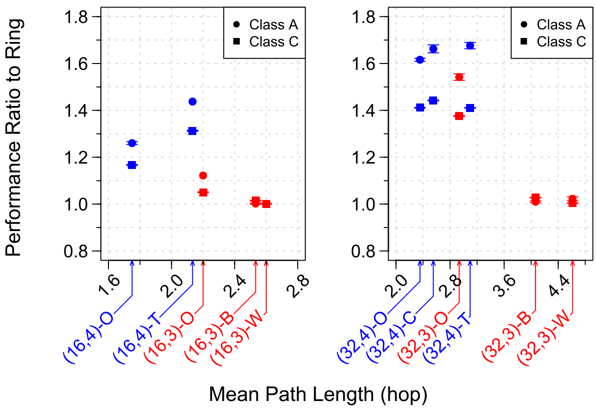
The performance ratios to ring are plotted in Fig. 4. Collective communications are influenced by MPL, BW, traffic pattern, MPI internal algorithm, message size and memory access. For example, Wagner topology has greater MPL but shorter diameter than Bidiakis, while they have the same bisection width (Table 1). The shorter diameter of Wagner graph is especially pronounced in the 1 MB message MPI_Bcast (Fig. 4a) which leads to a 17% and 11% performance gain, respectively, for (16,3)- and (32,3)-Wagner over Bidiakis. However, for larger messages and other MPI collective functions with similar traffic pattern such as MPI_Scatter (Fig. 4c), MPL becomes a more dominant factor and Bidiakis outperforms or at least performs equally as Wagner with slight fluctuation. Static shortest-path routing also affects the performance of collective communications. For example, torus has relatively low performance in MPI collective functions with large message, except MPI_Reduce (Fig. 4b). The low performance when transferring large message may




**(a)** NPB IS

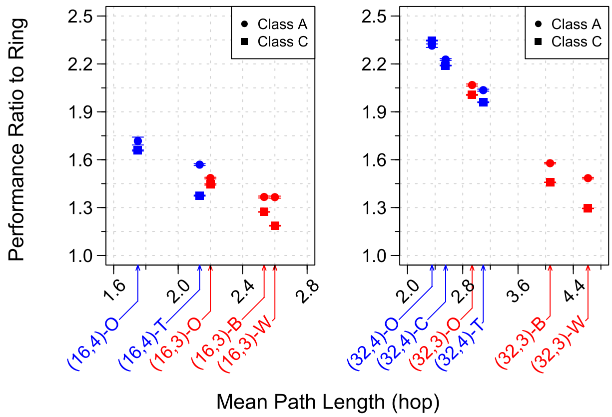



**(b)** NPB CG

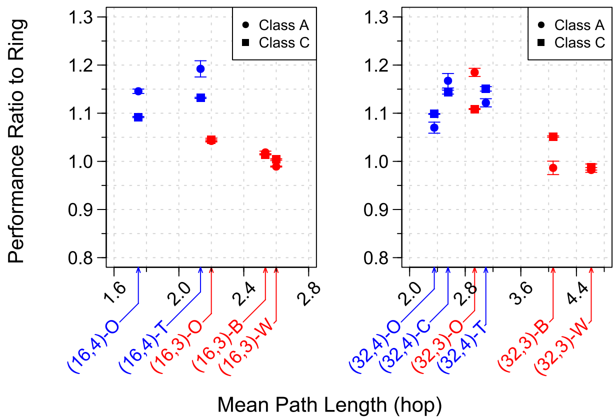



**(c)** NPB MG

**Fig. 8** Performance ratios on NPB




**(d)** NPB FT




**(e)** NPB LU

**Fig. 8** (continued)

be caused by network congestion due to static routing, especially for torus, while the internal algorithm of MPI_Reduce overcomes such congestion.

#### 4.2.3 Effective bandwidth

Effective bandwidth (b_eff, version 3.6.0.1) [1] measures the accumulated network bandwidth by means of multiple communication patterns (ordered naturally and randomly) with messages of 21 sizes ranging from 1 byte to 1/128 of memory per processor, 64 MB in Taishan. It uses MPI_Sendrecv, MPI_Alltoallv and non-blocking MPI_Irecv and MPI_Isend with MPI_Waitall. The output is the average bandwidth over ring and random patterns and 21 message sizes after taking the maximum bandwidth of the three MPI methods in each measurement [50].

The performance ratios to ring are plotted in Fig. 5. A strong impact of MPL on b_eff benchmark is shown, though traffic patterns, message sizes and MPI methods may also affect performance. Indeed, (16,4)- and (32,4)-Optimal have the highest effective bandwidths, 686.51 MB/s (and 1066.80 MB/s), a performance gain of 38% (and 68%) over (16,3)- and (32,3)-Wagner. Indeed, we can consider that performance of b_eff has an inversely proportional relation to MPL if we neglect the torus because the static shortest-path routing causes congestion in collective MPI functions.

#### 4.2.4 FFTE

We benchmarked the version 6.0 of the parallel FFTE [2, 70] from the HPC Challenge [4, 56], which in cache-based processors [69] has data transpositions as its main bottleneck because of all-to-all communications. We perform the parallel 1D FFTE routine with transform array lengths ranging from $2^{10}$ to $2^{27}$, limited by local 8 GB RAM. Then, we select $2^{21}$ and $2^{27}$ as the transform array lengths (equal to 32 MB and 2 GB in total transform array sizes).

Figure 6 shows the performance plots of 1D FFTE. Transforming larger arrays stresses the network such that 1D FFTE performs with almost linear dependence of MPL. When transforming 2 GB array in 1D FFTE, (16,4)- and (32,4)-Optimal topologies have top performance ratios of 1.85 and 2.31 to ring, a gain of 51% and 74% over (16,3)- and (32,3)-Wagner. For arrays < 32 MB, the performances are almost uniform for all network topologies.

#### 4.2.5 Graph 500

The Graph 500 (version 3.0.0) [3, 61] tests large-scale graph algorithms, where multiple breadth-first search (BFS) and single-source shortest path (SSSP) computations are performed on an extremely large undirected graph generated and distributed in the beginning of the test. Graph 500 evaluates data-intensive performance in supercomputers reporting the mean TEPS (traversed edges per second). The best choice for test scale limited by local RAM was 27, generating an initial unweighted graph of 24 GB for BFS and an initial weighted graph of 40 GB for SSSP.

Figure 7 shows the performance of Graph 500 benchmark. A strong inversely proportional relation to MPL is exhibited, despite fluctuations on torus (because of congestion), Bidiakis and (32,4)-Chvatal. The relatively high diameter of Bidiakis compared with Wagner and relatively low bisection width of (32,4)-Chvatal compared with (32,3)-Optimal topology (Table 1) weaken their performances as well. However, MPL keeps playing a major role on Graph 500 with (16,4)- and (32,4)-Optimal having top performances of, respectively, 3.05/2.71 and 5.41/4.75 for BFS/SSSP, a gain of 90%/71% and 278%/271% over (16,3)- and (32,3)-Wagner.

#### 4.2.6 NAS parallel benchmarks (NPB)

The NAS Parallel Benchmarks (NPB version 3.3.1 on MPI) [5, 12] contain a set of programs derived from computational fluid dynamics (CFD) applications, with built-in runtime reporting. We run integer sort (IS), conjugate gradient method (CG) for approximating the smallest eigenvalue, multi-grid solver (MG) for 3D Poisson PDE, FFT solver (FT) for 3D PDE NPB kernels, and lower–upper (LU) Gauss–Seidel solver pseudo-application [57]. IS uses intensive data communication, while also testing random memory access and integer computation speed; CG tests unstructured long-distance communication and irregular memory access; MG tests highly structured short- and long-distance communication with intensive memory access; FT tests long-distance all-to-all communication [5, 12, 13]. For each benchmark, we choose the standard problem sizes: Class A, B and C because of local memory constraints.

The performance ratios to ring for Classes A and C are shown in Fig. 8. Note that traffic patterns, internal algorithms, problem sizes, memory access and static shortest-path routing, apart from MPL and BW, affect the performance of NPB. The performances of CG (Fig. 8b) and MG (Fig. 8c) are similar to MPI_Reduce (Fig. 4b), in which torus shows relatively high performance. In these benchmarks, the static routing for torus does not cause congestion with internal algorithms and memory access benefitting the torus. LU (Fig. 8e) shows a nearly uniform performance over all benchmarked topologies, a result attributable to its limited parallelism [12], i.e., low communication-to-computation ratio. However, NPB performance exhibits weak, or even strong, dependence on MPL as in IS (Fig. 8a) and FT (Fig. 8d) resembling, respectively, Graph 500 (Fig. 7) and 1D FFTE with 2 GB array size (Fig. 6), as expected for benchmarks requiring heavy global communication. IS and FT Class A/C problem sizes are $2^{23}/2^{27}$ resulting in, respectively, 32 MB/512 MB total integer array sizes and 128 MB/2 GB transform array sizes. In IS Cass A/C, (16,4)- and (32,4)-Optimal topologies have top performance ratios of 2.70/2.89 and 3.89/4.32, respectively, a gain of 79%/93% and 153%/202% over (16,3)- and (32,3)-Wagner. In FT Class A/C, (16,4)- and (32,4)-Optimal topologies have top performance ratios of 2.70/2.89 and 3.89/4.32, respectively, a gain of 79%/93% and 153%/202% over (16,3)- and (32,3)-Wagner. In FT Class A/C, the optimal graphs, 1.72/1.66 and 2.31/2.35, outperform both Wagner graphs with a gain of 26%/40% and 56%/81%, respectively.

### 4.3 Large-scale topology optimization and simulation analysis

#### 4.3.1 Comparative analysis of larger-scale near-optimal network topologies

We obtain the near-optimal topologies of 256 nodes and degrees 3, 4, 6, 8 using random iteration of Hamiltonian graphs with rotational symmetry. The near-optimal topologies are compared with topologies of the same size and degrees: ring, Wagner, Bidiakis, $16 \times 16$ torus (4D hypercube), $4 \times 8 \times 8$ torus and $4 \times 4 \times 4 \times 4$ torus (8D hypercube), as shown in Table 2. For the near-optimal topologies, we also calculate their gaps of diameter and MPL compared to the theoretical lower bounds, respectively. Figures of the near-optimal topologies are listed in the Appendix. The adjacency matrices of all the simulated topologies are included in additional Online Resource.

Table 2 shows that the near-optimal topologies have the smallest diameter (*D*), MPL and highest bisection width (BW) among the topologies of the same sizes and

**Table 2** Graph properties of simulated topologies

| Topology | D[a] | MPL[a] | BW |
|---|---|---|---|
| **(256,8)-Near-optimal** | **3 + 1** | **2.72 + 0.03** | **298** |
| (256,8)-Torus | 8 | 4.02 | 128 |
| **(256,6)-Near-optimal** | **4 + 0** | **3.11 + 0.06** | **192** |
| (256,6)-Torus | 10 | 5.02 | 64 |
| **(256,4)-Near-optimal** | **5 + 1** | **4.09 + 0.05** | **92** |
| (256,4)-Torus | 16 | 8.03 | 32 |
| **(256,3)-Near-optimal** | **7 + 1** | **5.59 + 0.08** | **46** |
| (256,3)-Bidiakis | 65 | 25.09 | 4 |
| (256,3)-Wanger | 64 | 32.62 | 4 |
| (256,2)-Ring | 128 | 64.25 | 2 |

[a] The D and MPL of near-optimal topologies are written as the sum of the theoretical lower bounds and the difference to final values

degrees. Properties of near-optimal graphs are emphasized with bold fonts. For the gaps of D and MPL of near-optimal topologies, the diameter gap is within 1 and MPL gap is within 2% compared to the theoretical lower bounds. This shows our optimization method is effective on the large scale. The current optimization runtime is 96 h, and one may further extend the runtime or improve the method to obtain better near-optimal topologies.

#### 4.3.2 Simulation results and analysis

We simulate larger-scale topologies on the platform SimGrid (version 3.21) [22]. SimGrid provides versatile, accurate and scalable simulation of distributed applications, especially with SMPI API that enables simulation of unmodified MPI applications [22]. We configure SimGrid to approximate the settings and ping-pong test results of Taishan cluster, with dual-core CPU per host, 8 Gflops processing speed per core, gigabit bandwidth and 30 μs latency per link. Static shortest-path routing is implemented with full routing table calculated using the same algorithm as for the benchmarking cluster. We run the simulations on the SeaWulf cluster at Stony Brook University.

We select the benchmarks that largely depend on global communication: MPI_Alltoall, effective bandwidth, 1D FFTE, Graph 500 and NPB IS and FT. Because of the limited 128 GB RAM of SeaWulf nodes and long simulation runtime for large-scale topologies, we reduce the problem sizes for some benchmarks, namely 64 KB and 512 KB as the unit message sizes for MPI_Alltoall, 1 MB maximum message size for effective bandwidth and Class S and A for NPB IS. For Graph 500, due to implementation issues with SimGrid, we use a previous version 2.1.4 that only contains BFS test and reduce the test scale to 12.

The simulation performance ratios to ring are plotted in Fig. 9 for topologies of 256 nodes, with log scale on MPL. The near-optimal topologies are labeled as $(N, k) - N$ and gold (or cyan) points indicate the data for degree-6 (or 8) clusters.

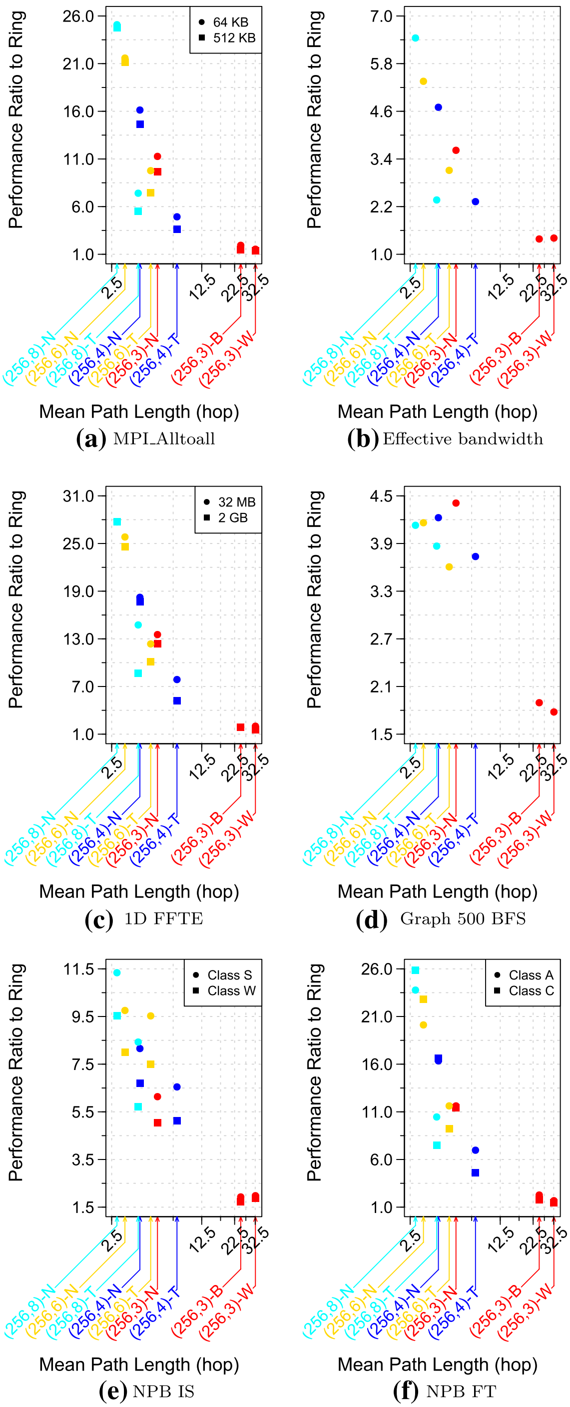



**Fig. 9** Performance ratios on simulated MPI_Alltoall, effective bandwidth, 1D FFTE, Graph 500 BFS and NPB

The simulation results reveal that for large-scale topologies, (256, $k$)-Near-optimal with low MPL has mostly prominent performance increase over other topologies with the same degree. Despite fluctuations in Graph 500 BFS (Fig. 9d) and NPB IS (Fig. 9e) due to limited problem sizes and thus less intensive communication, all the simulation performances show a strongly inversely proportional relation with respect to MPL. The performance gain of (256,8)-Near-optimal over (256,3)-Wagner is above 1000% in MPI_Alltoall (Fig. 9a), 1D FFTE (Fig. 9c) and NPB FT (Fig. 9f). Again, tori show low performance partially due to network congestion caused by static shortest-path routing.

## 5 Discussion and conclusion

In this manuscript, we examine our hypothesis on increasing a cluster's sustained processing speed by interconnecting its nodes with a minimal MPL network topology. That is done experimentally in small clusters supported by optimal symmetric regular graphs generating advanced network topologies. We build clusters of the same size with multiple topologies, namely torus, Wagner, Bidiakis, Chvatal and ring, to run a basic set of benchmarks. Our results show that the optimal network topologies, in general, deliver the highest performance. We also perform simulations of larger clusters that confirm our observations. Moreover, our results attest to the effectiveness and importance of the mathematically driven design of network topologies.

The minimum MPL graphs were constructed using our parallel enumeration algorithm with girth restrictions and random iteration on Hamiltonian graphs that generated a reduced search space by imposition of symmetry requirements. These methods are general, being applied well for the search of small and large (near) optimal network topologies. Hence, one may employ our algorithm for generating advanced network topologies for clusters of enhanced performance and provide parallel computers architects with an additional rationale to enhance those machine's performance.

Our results running high communication-to-computation ratio applications, namely MPI_Alltoall-based tests, effective bandwidth, 1D FFTE, Graph 500, and NPB IS and FT, indicate the strong influence of MPL on the clusters' performance. This proves the importance of network topologies with optimized MPL for speeding up processing and encourages designing clusters using (near) optimal symmetric regular graphs. Our results are also useful for architects designing switched networks, the communicating circuitry of multicore processors or DCN topologies. The (near) optimal graphs obtained with our algorithms can provide reduced communication times for any type of network since there is no assumption on the properties of the nodes. Architects interested on larger-scale clusters would still benefit from our methods as the (near) optimal graphs can be combined by graph product [29] or integrated as base graphs into hierarchical networks [47, 63, 67] to construct scalable network topologies of reduced latency and compete with other multistage networks like fat tree [31].

Optimal symmetric network topologies of minimal MPL are also important for ensuring engineering feasibility as demonstrated by the construction of our cluster Taishan. It enables optimal use of the available hardware while adding minimal costs: the time and energy for computational search of the optimal topology for a regular graph of a given size and node degree. Hence, further development of mathematical tools for minimizing the computer search time or, in an ideal scenario, finding optimal graphs by analytic calculations would be welcome. Currently, the parallel exhaustive search for (32,3)-Optimal graph without girth constraint goes through $\sim 10^{13}$ graphs and took about one week on thousands of Sunway BlueLight cores [74]. That amount of time is greatly reduced if we consider the symmetries and obtain near-optimal graphs as done for the 256-node graphs. Such improvement on the optimization method may lead to the discovery of larger-scale (near) optimal graphs in combination with graph product [59], hierarchical construction [14, 15] and other graph design and optimization techniques [62, 73].

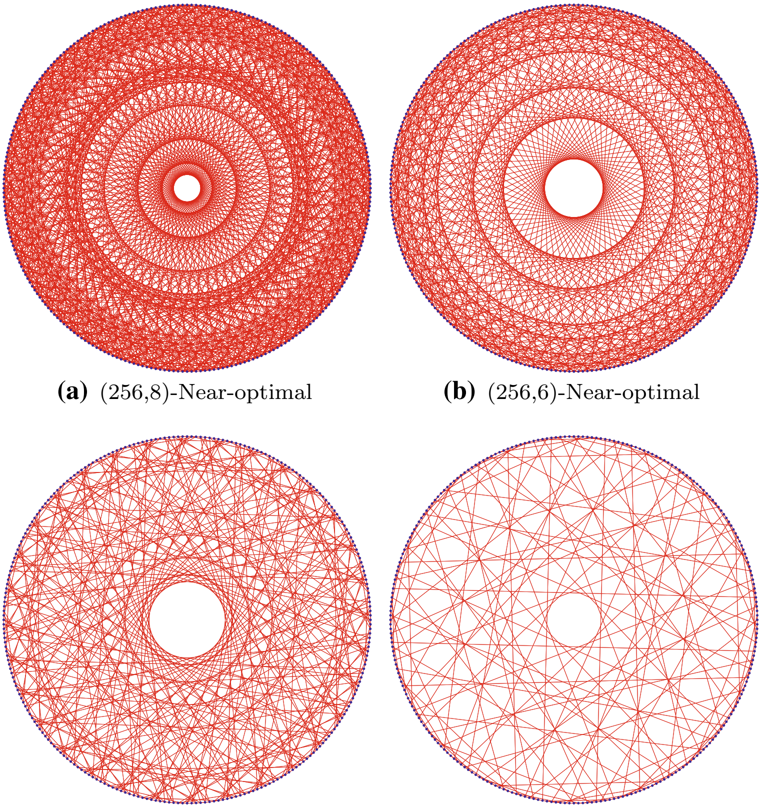
The linear relation between the distance and latency matrices for, respectively, the graph and the networks demonstrates the strength of our mathematically driven design as an additional layer for a supercomputer's optimization. Tables 1 and 2 show the properties of the networks that we have evaluated in our work, and also the symmetric (near) optimal graphs having minimized diameters and maximized bisection widths. Those two quantities also help in enhancing the cluster's performance as is widely known by supercomputer and DCN architects. Hence, our approach enables the concomitant optimization of three parameters.

A seminal model for latency in a computer's network considers its dependence on: its components technology determining both the time of message processing in a single router, $t_R$, and the velocity of package propagation through interconnects, $v$; the network topology determining the average hop distance, $H$, average cable distance, $\mu$, and bandwidth, $b$, that depends on node degree and packaging constraints (Section 3.3.2 of [28]). The latency of a message can be written as $T = Ht_R + \mu/v + L/b$, where $L$ is the message length. Since the performance of the components is a fixed parameter given by financial, energetic and technological constraints, latency reduction can be achieved by increasing node's degree and reducing average hop and cable distances. The linear relation between the latency and hop distance for the ping-pong test is contrast with the non-trivial dependence on MPL when more complex benchmarks are executed. Hence, a more complex theoretical work [11, 28, 35, 39, 57, 71] is necessary for understanding the dependence of a cluster's performance on its network topology and prevalent applications and to establish general principles to be used by supercomputer architects.

**Acknowledgements** This work is supported by Key Research and Development Program of Shandong Province of China (2015GGX101028), National Natural Science Foundation of China (11674264), Science Foundation for Youths of Shandong Academy of Sciences of China (No. 2014QN010) and Special Fund for Overseas High-level Talents of Shandong Province of China. AFR thanks CAPES Process n. 88881.062174/2014-0 and Tinker Visiting Professorship at CLAS-University of Chicago. Z. Xu thanks the Special Project on High-Performance Computing of the National Key R&D Program under No.2016YFB0200604. The authors thank the following individuals for their contributions to this project at different stages on various aspects: Y. Hu, X. Li, C. Chen, E. Raut, H. Fan, K. Wu, C. Han, Z. Ye, L. Zhang, D. McLoone, L. Pei, Z. Li, J. Han, H. Zheng, T. Mayer and Prof. J. Mitchell.

## Appendix: Simulated large-scale near-optimal topology figures

See Fig. 10.



**(a)** (256,8)-Near-optimal **(b)** (256,6)-Near-optimal **(c)** (256,4)-Near-optimal **(d)** (256,3)-Near-optimal

**Fig. 10** Simulated large-scale near-optimal topologies

## References


1. (2019) Effective bandwidth (b_eff) benchmark. https://fs.hlrs.de/projects/par/mpi/b_eff/
2. (2019) FFTE: a fast Fourier transform package. http://www.ffte.jp/
3. (2019) Graph 500. http://graph500.org/
4. (2019) HPC challenge benchmark. http://icl.utk.edu/hpcc/index.html
5. (2019) NPB: NAS parallel benchmarks. http://www.nas.nasa.gov/publications/npb.html
6. (2019) Top 500 supercomputer site (2019). http://www.top500.org
7. Abd-El-Barr M, Al-Somani TF (2011) Topological properties of hierarchical interconnection networks: a review and comparison. J Electr Comput Eng 2011:1–12. https://doi.org/10.1155/2011/189434

8. Adiga NR, Blumrich MA, Chen D, Coteus P, Gara A, Giampapa ME, Heidelberger P, Singh S, Steinmacher-Burow BD, Takken T, Tsao M, Vranas P (2005) Blue Gene/L torus interconnection network. IBM J Res Dev 49(2–3):265–276. https://doi.org/10.1147/rd.492.0265
9. Ajima Y, Sumimoto S, Shimizu T (2009) Tofu: a 6D mesh/torus interconnect for exascale computers. Computer 42(11):36–40. https://doi.org/10.1109/mc.2009.370
10. Alverson R, Roweth D, Kaplan L (2010) The gemini system interconnect. In: 2010 18th IEEE Symposium on High Performance Interconnects. IEEE. https://doi.org/10.1109/hoti.2010.23
11. Ardagna D, Barbierato E, Evangelinou A, Gianniti E, Gribaudo M, Pinto TB, Guimarães A, Couto da Silva AP, Almeida JM (2018) Performance prediction of cloud-based big data applications. In: Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering—ICPE'18. ACM Press, pp 192–199. https://doi.org/10.1145/3184407.3184420
12. Bailey D, Barszcz E, Barton J, Browning D, Carter R, Dagum L, Fatoohi R, Frederickson P, Lasinski T, Schreiber R, Simon H, Venkatakrishnan V, Weeratunga S (1991) The NAS parallel benchmarks. Int J Supercomput Appl 5(3):63–73. https://doi.org/10.1177/109434209100500306
13. Bailey D, Barszcz E, Dagum L, Simon H (1992) NAS parallel benchmark results. In: Proceedings Supercomputing '92. IEEE Computer Society Press. https://doi.org/10.1109/superc.1992.236665
14. Barriere L, Comellas F, Dalfó C, Fiol MA (2009) The hierarchical product of graphs. Discrete Appl Math 157(1):36–48. https://doi.org/10.1016/j.dam.2008.04.018
15. Barrière L, Dalfó C, Fiol MA, Mitjana M (2009) The generalized hierarchical product of graphs. Discrete Math 309(12):3871–3881. https://doi.org/10.1016/j.disc.2008.10.028
16. Besta M, Hoefler T (2014) Slim fly: a cost effective low-diameter network topology. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC'14. IEEE Press, Piscataway, NJ, USA, pp 348–359. https://doi.org/10.1109/SC.2014.34
17. Bondy JA, Murty U (1976) Graph theory with applications. Elsevier, Amsterdam
18. Brightwell R, Pedretti K, Underwood K, Hudson T (2006) SeaStar interconnect: balanced bandwidth for scalable performance. IEEE Micro 26(3):41–57. https://doi.org/10.1109/mm.2006.65
19. Brinkmann G, Goedgebeur J (2017) Generation of cubic graphs and snarks with large girth. J Graph Theory 86(2):255–272. https://doi.org/10.1002/jgt.22125
20. Brinkmann G, Goedgebeur J, McKay BD (2011) Generation of cubic graphs. Discrete Math Theor Comput Sci 13(2):69–79
21. Brinkmann G, Coolsaet K, Goedgebeur J, Mélot H (2013) House of graphs: a database of interesting graphs. Discrete Appl Math 161(1–2):311–314. https://doi.org/10.1016/j.dam.2012.07.018
22. Casanova H, Giersch A, Legrand A, Quinson M, Suter F (2014) Versatile, scalable, and accurate simulation of distributed applications and platforms. J Parallel Distrib Comput 74(10):2899–2917. https://doi.org/10.1016/j.jpdc.2014.06.008
23. Cerf VG, Cowan DD, Mullin RC, Stanton RG (1974) A lower bound on the average shortest path length in regular graphs. Networks 4(4):335–342. https://doi.org/10.1002/net.3230040405
24. Cerf VG, Cowan DD, Mullin RC, Stanton RG (1975) A partial census of trivalent generalized Moore networks. In: Combinatorial Mathematics III. Springer, Berlin, pp 1–27. https://doi.org/10.1007/bfb0069540
25. Chen D, Parker JJ, Eisley NA, Heidelberger P, Senger RM, Sugawara Y, Kumar S, Salapura V, Satterfield DL, Steinmacher-Burow B (2011) The IBM Blue Gene/Q interconnection network and message unit. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis on—SC'11. ACM Press. https://doi.org/10.1145/2063384.2063419
26. Dally W (1990) Performance analysis of k-ary n-cube interconnection networks. IEEE Trans Comput 39(6):775–785. https://doi.org/10.1109/12.53599
27. Dally W (1991) Express cubes: improving the performance of k-ary n-cube interconnection networks. IEEE Trans Comput 40(9):1016–1023. https://doi.org/10.1109/12.83652
28. Dally W, Towles B (2003) Principles and practices of interconnection networks. Elsevier, Amsterdam
29. Day K, Al-Ayyoub AE (1997) The cross product of interconnection networks. IEEE Trans Parallel Distrib Syst 8(2):109–118. https://doi.org/10.1109/71.577251
30. Deng Y, Ramos AF, Hornos JEM (2012) Symmetry insights for design of supercomputer network topologies: roots and weights lattices. Int J Mod Phys B 26(31):1250169. https://doi.org/10.1142/s021797921250169x
31. Domke J, Matsuoka S, Ivanov IR, Tsushima Y, Yuki T, Nomura A, Miura S, McDonald N, Floyd DL, Dubé N (2019) HyperX topology: first at-scale implementation and comparison to the fat-tree.

In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. ACM https://doi.org/10.1145/3295500.3356140
32. Efe K (1991) A variation on the hypercube with lower diameter. IEEE Trans Comput 40(11):1312–1316. https://doi.org/10.1109/12.102840
33. Faanes G, Bataineh A, Roweth D, Court T, Froese E, Alverson B, Johnson T, Kopnick J, Higgins M, Reinhard J (2012) Cray cascade: a scalable HPC system based on a dragonfly network. In: 2012 International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE. https://doi.org/10.1109/sc.2012.39
34. Floyd RW (1962) Algorithm 97: shortest path. Commun ACM 5(6):345. https://doi.org/10.1145/367766.368168
35. Foroutan S, Thonnart Y, Hersemeule R, Jerraya A (2010) An analytical method for evaluating network-on-chip performance. In: 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010). IEEE, pp 1629–1632. https://doi.org/10.1109/date.2010.5457072
36. Freund R, Wilson W, Mohr D (2010) Statistical methods, 3rd edn. Academic Press, Cambridge
37. Fu H, Liao J, Yang J, Wang L, Song Z, Huang X, Yang C, Xue W, Liu F, Qiao F, Zhao W, Yin X, Hou C, Zhang C, Ge W, Zhang J, Wang Y, Zhou C, Yang G (2016) The Sunway TaihuLight supercomputer: system and applications. Sci China Inf Sci. https://doi.org/10.1007/s11432-016-5588-7
38. Garzón DB, Gómez C, Gómez ME, López P, Duato J (2012) Towards an efficient fat–tree like topology. In: Euro-Par 2012 Parallel Processing. Springer, Berlin, pp 716–728. https://doi.org/10.1007/978-3-642-32820-6_71
39. Gupta AK, Dally WJ (2006) Topology optimization of interconnection networks. IEEE Comput Archit Lett 5(1):10–13. https://doi.org/10.1109/l-ca.2006.8
40. Harary F, Hayes JP, Wu HJ (1988) A survey of the theory of hypercube graphs. Comput Math Appl 15(4):277–289. https://doi.org/10.1016/0898-1221(88)90213-1
41. Harwood A, Shen H (1998) A low cost hybrid fat-tree interconnection network. In: Proceedings of International Conference on Parallel and Distributed Processing and Applications, pp 682–689
42. Hayes J, Mudge T (1989) Hypercube supercomputers. Proc IEEE 77(12):1829–1841. https://doi.org/10.1109/5.48826
43. Hill MD, Jouppi NP, Sohi GS (1999) Readings in computer architecture. Morgan Kaufmann, Burlington
44. IBM Blue Gene Team (2008) Overview of the IBM Blue Gene/P project. IBM J Res Dev 52(1–2):199–220. https://doi.org/10.1147/rd.521.0199
45. InfiniBand@ Trade Association (2016) InfiniBand architecture specification, release 1.3. http://www.infinibandtaorg
46. Inoguchi Y, Horiguchi S (1997) Shifted recursive torus interconnection for high performance computing. In: Proceedings High Performance Computing on the Information Superhighway. HPC Asia'97. IEEE Computer Society Press. https://doi.org/10.1109/hpc.1997.592123
47. Jan GE, Hwang Y, Lin M, Liang D (2004) Novel hierarchical interconnection networks for high-performance multicomputer systems. J Inf Sci Eng 20:1213–1229
48. Kim J, Dally WJ, Scott S, Abts D (2008) Technology-driven, highly-scalable dragonfly topology. In: 2008 International Symposium on Computer Architecture. IEEE. https://doi.org/10.1109/isca.2008.19
49. Kitasuka T, Iida M (2016) A heuristic method of generating diameter 3 graphs for order/degree problem (invited paper). In: 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS). IEEE. https://doi.org/10.1109/nocs.2016.7579334
50. Koniges A, Rabenseifner R, Solchenbach K (2001) Benchmark design for characterization of balanced high-performance architectures. In: Proceedings 15th International Parallel and Distributed Processing Symposium. IPDPS 2001. IEEE Computer Society Press. https://doi.org/10.1109/ipdps.2001.925208
51. Leiserson CE (1985) Fat-trees: universal networks for hardware-efficient supercomputing. IEEE Trans Comput C–34(10):892–901. https://doi.org/10.1109/tc.1985.6312192
52. Lenzen C, Wattenhofer R (2016) Clex: yet another supercomputer architecture? arXiv:1607.00298v1
53. Liao XK, Pang ZB, Wang KF, Lu YT, Xie M, Xia J, Dong DZ, Suo G (2015) High performance interconnect network for Tianhe system. J Comput Sci Technol 30(2):259–272. https://doi.org/10.1007/s11390-015-1520-7

54. Liu V, Halperin D, Krishnamurthy A, Anderson T (2013) F10: a fault-tolerant engineered network. In: Presented as Part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13). USENIX, Lombard, IL, pp 399–412
55. Liu YJ, Gao PX, Wong B, Keshav S (2014) Quartz: a new design element for low-latency DCNs. In: Proceedings of the 2014 ACM Conference on SIGCOMM—SIGCOMM'14. ACM Press. https://doi.org/10.1145/2619239.2626332
56. Luszczek PR, Bailey DH, Dongarra JJ, Kepner J, Lucas RF, Rabenseifner R, Takahashi D (2006) S12—the HPC challenge (HPCC) benchmark suite. In: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing—SC'06. ACM Press. https://doi.org/10.1145/1188455.1188677
57. Matsutani H, Koibuchi M, Amano H, Yoshinaga T (2009) Prediction router: yet another low latency on-chip router architecture. In: 2009 IEEE 15th International Symposium on High Performance Computer Architecture. IEEE, pp 367–378. https://doi.org/10.1109/hpca.2009.4798274
58. Meringer M (1999) Fast generation of regular graphs and construction of cages. J Graph Theory 30(2):137–146. https://doi.org/10.1002/(SICI)1097-0118(199902)30:2<137::AID-JGT7>3.0.CO;2-G
59. Mizuno R, Ishida Y (2016) Constructing large-scale low-latency network from small optimal networks. In: 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS). IEEE. https://doi.org/10.1109/nocs.2016.7579336
60. Moore GE (1965) Cramming more components onto integrated circuits. Electronics 38(8):114–117
61. Murphy RC, Wheeler KB, Barrett BW, Ang JA (2010) Introducing the graph 500. Cray Users Group (CUG) 19:45–74
62. Nakao M, Murai H, Sato M (2019) A method for order/degree problem based on graph symmetry and simulated annealing with MPI/OpenMP parallelization. In: Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region. ACM Press, pp 128–137. https://doi.org/10.1145/3293320.3293325
63. Rahman MMH, Nor RM, Sembok TMBT, Akhand MAH (2015) Architecture and network-on-chip implementation of a new hierarchical interconnection network. J Circuits Syst Comput 24(02):1540006. https://doi.org/10.1142/s021812661540006x
64. Sabino AU, Vasconcelos MFS, Deng Y, Ramos AF (2018) Symmetry-guided design of topologies for supercomputer networks. Int J Mod Phys C 29(07):1850048. https://doi.org/10.1142/s0129183118500481
65. Sanders P, Schulz C (2013) Think locally, act globally: highly balanced graph partitioning. In: Experimental Algorithms. Springer, Berlin, pp 164–175. https://doi.org/10.1007/978-3-642-38527-8_16
66. Scott SL et al (1996) The Cray T3E network: adaptive routing in a high performance 3D torus
67. Seo JH, Kim JS, Chang HJ, Lee HO (2017) The hierarchical Petersen network: a new interconnection network with fixed degree. J Supercomput 74(4):1636–1654. https://doi.org/10.1007/s11227-017-2186-4
68. Shimizu N, Mori R (2016) Average shortest path length of graphs of diameter 3. In: 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS). IEEE. https://doi.org/10.1109/nocs.2016.7579335
69. Takahashi D (2002) A blocking algorithm for parallel 1-D FFT on shared-memory parallel computers. In: Lecture Notes in Computer Science. Springer, Berlin, pp 380–389. https://doi.org/10.1007/3-540-48051-x_38
70. Takahashi D, Kanada Y (2000) High-performance radix-2, 3 and 5 parallel 1-D complex FFT algorithms for distributed-memory parallel computers. J Supercomput 15(2):207–228. https://doi.org/10.1023/a:1008160021085
71. Wang S, Li D, Geng J, Gu Y, Cheng Y (2019) Impact of network topology on the performance of DML: theoretical analysis and practical factors. In: IEEE INFOCOM 2019—IEEE Conference on Computer Communications. IEEE, pp 1729–1737. https://doi.org/10.1109/infocom.2019.8737595
72. Weisstein EW (2018) Bidiakis Cube. http://mathworld.wolfram.com/BidiakisCube.html
73. Xu J (2013) Topological structure and analysis of interconnection networks, vol 7. Springer, Berlin
74. Xu Z, Huang X, Jimenez F, Deng Y (2019) A new record of graph enumeration enabled by parallel processing. Mathematics 7(12):1214. https://doi.org/10.3390/math7121214
75. Yang Y, Funahashi A, Jouraku A, Nishi H, Amano H, Sueyoshi T (2001) Recursive diagonal torus: an interconnection network for massively parallel computers. IEEE Trans Parallel Distrib Syst 12(7):701–715. https://doi.org/10.1109/71.940745

76. Zhang P, Powell R, Deng Y (2011) Interlacing bypass rings to torus networks for more efficient networks. IEEE Trans Parallel Distrib Syst 22(2):287–295. https://doi.org/10.1109/tpds.2010.89
77. Zhang P, Deng Y, Feng R, Luo X, Wu J (2015) Evaluation of various networks configurated by adding bypass or torus links. IEEE Trans Parallel Distrib Syst 26(4):984–996. https://doi.org/10.1109/tpds.2014.2315201



## Affiliations

**Yuefan Deng[1] · Meng Guo[2] · Alexandre F. Ramos[3,4] · Xiaolong Huang[1] · Zhipeng Xu[1,5] · Weifeng Liu[6]**

Yuefan Deng
yuefan.deng@stonybrook.edu
https://you.stonybrook.edu/yuefandeng/

Meng Guo
guomeng@sdas.org

Xiaolong Huang
xiaolong.huang@stonybrook.edu

Zhipeng Xu
xuzhp9@mail2.sysu.edu.cn

Weifeng Liu
ise_liuwf@ujn.edu.cn

1 Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794, USA

2 Shandong Computer Science Center (National Supercomputer Center in Jinan) and Shandong Provincial Key Laboratory of Computer Networks, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250101, Shandong, People's Republic of China

3 Escola de Artes, Ciencias e Humanidades, Center for Translational Research in Oncology (LIM-24), Instituto do Cancer do Estado de Sao Paulo, Faculdade de Medicina, Universidade de Sao Paulo, Avenida Arlindo Bettio 1000, Sao Paulo CEP 03828-000, Brazil

4 Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan 250101, Shandong, People's Republic of China

5 School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, Guangdong, People's Republic of China

6 School of Information Science and Engineering, University of Jinan, Jinan 250022, Shandong, People's Republic of China