Towards Ideal Hop Counts in Interconnection Networks with Arbitrary Size

Michihiro Koibuchi*†,
Ikki Fujiwara*
*National Institute of Informatics, Tokyo, Japan
† The Graduate University for Advanced Studies (SOKENDAI)
{koibuchi,ikki}@nii.ac.jp

Fabien Chaix‡
‡ Institute of Computer Science, Foundation for Research and Technology - Hellas, Greece
fabien.chaix@gmail.com

Henri Casanova§
§University of Hawai‘i at Manoa, Honolulu
henric@hawaii.edu

***Abstract*—Designing low-latency network topologies of switches is a key objective for next-generation parallel computing platforms. Low latency is preconditioned on low hop counts, but existing network topologies have hop counts much larger than theoretical lower bounds. The degree diameter problem (DDP) has been studied for decades and consists in generating the largest possible graph given degree and diameter constraints, striving to approach theoretical upper bounds. To generate network topologies with low hop counts we propose using best known DDP solutions as starting points for generating topologies of arbitrary size. Using discrete-event simulation, we quantify the performance of representative parallel applications when executed on our proposed topologies, on previously proposed fully random topologies, and on a classical non-random topology.**

***Index Terms*—interconnection networks; network topology; diameter; random topology**

I. Introduction

A goal for upcoming high performance computing (HPC) systems with possibly millions of cores is to achieve low network latency, e.g., $1\mu s$ across system, as well as high bisection bandwidth [12]. Switch delays, e.g., around 100 or 200 nsec for recent InfiniBand switches, are large compared to the typical 5ns/m cable delay. To achieve low latency, a topology of switches should thus have low diameter and low average shortest path length (ASPL), both measured in numbers of switch hops. Defined by graph theoreticians, the degree diameter problem (DDP) consists in finding the largest graph for given degree and diameter constraints. This problem has been studied both for graphs and best known solutions are publicly available [7]. Alternatively, the order degree problem (ODP) that finds out the best graph in terms of diameter and ASPL for given size and degree constraints is recently discussed in [1]. At present the graphs for a limited number of their pairs are publicly available.

However, currently available ODP or DDP solutions may not be directly usable for network topologies in HPC systems because they are for particular number of switches, whereas the number of switches in a real system is determined based on practical considerations, e.g., budget.

A recent approach for generating topologies with low hop counts is to use randomness. Topologies have been proposed that are generated from fully random graphs [21] or by adding random links to non-random topologies [15]. These works show that randomization leads to dramatically lower hop counts than same-degree non-random topologies traditionally used in parallel platforms, especially for low-degree topologies. A key advantage of random topologies is that they can be generated for arbitrary network sizes, which is not the case for best known DDP solutions or even for traditional structured non-random topologies such as k-ary n-cubes. Random topologies, however, lead to increased cable length, and cannot benefit from custom routing schemes. These issues have been investigated in previous work [14].

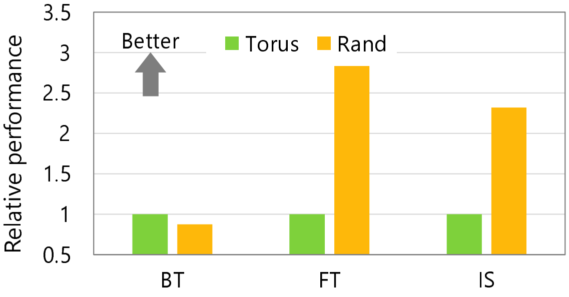

Figure 1. Performance of the BT, FT and IS NAS Parallel Benchmarks on a non-random torus topology and a fully random topology, relative to the performance on the non-random torus topology.

Decades of parallel algorithm research have gone into developing efficient mappings of parallel scientific applications with “regular” workloads onto structured topologies, e.g., parallel numerical linear algebra applications on high-diameter, high-ASPL k-ary n-cubes. Low diameter and low ASPL are desirable for “irregular” workloads, i.e., those parallel applications in which communication patterns are not necessarily structured and in which any process may need to communicate with any other process throughout application execution. Many emerging big data processing applications have such irregular workloads. A well-known example is graph analysis applications [23], which can benefit from large HPC systems with low-latency interconnects [16]. For such applications, rack-scale systems are envisioned that allow for

2379-1896/16 $31.00 © 2016 IEEE
DOI 10.1109/CANDAR.2016.5

a few hundreds nanoseconds latency across 100,000 cores in a single cabinet [8]. Finally, some traditional compute-intensive applications (e.g., in the field of weather prediction [16]) have also become data-intensive and exhibit irregular or all-to-all communication patterns.

One interesting question is: how do different HPC applications fare on interconnects with random network topologies? Figure 1 shows simulation results for the BT (Block Tridiagonal), FT (Fourier Transform) and IS (Integer Sort) NAS Parallel Benchmarks (NPB) [22] for a non-random torus topology and a fully random topology, both with degree d=8 and n=256 switches. These results are obtained with the SimGrid discrete-event simulator [20] (details of the experimental methodology are given in Section IV). The BT benchmark experiences a 13% performance decrease when comparing its execution on the random topology to that on the non-random torus. This is because BT relies mostly on neighbor-to-neighbor communications. By contrast, a random topology can provide significant advantage to applications that use (all-to-all) collective communications. This is the case for FT and IS, for which the results in Figure 1 show a 231% and 132% performance increase on the random topology, respectively. A random topology also provides performance advantages for application with irregular communication patterns.

Although randomly generated topology graphs achieve low hop counts, there is still room for improvement. For a degree d and a network size n, a lower bound for the diameter for a graph is computed as the lowest value k such that $1 + d\sum_{i=0}^{k-1}(d-1)^i \geq n$. A lower bound on the ASPL is computed in a similar manner, but weighting each term by path length and averaging over n. Let us consider an example with n=1,024 and d=35. Results presented later in this paper show that a fully random graph has a diameter of 3 and an ASPL of 2.25, while the diameter lower bound is 2 and the ASPL lower bound is 1.97. It turns out that the best known DDP solution with diameter 2 for this degree is a network with $n = 1,058$ vertices. This graph has an ASPL of 1.97.

Our objective is to generate network topologies of arbitrary size with hop counts close to the theoretical bounds, or at least closer than that of previously proposed fully random topologies. Our approach is to use best known DDP solutions as starting points to which vertices and edges are added/removed to achieve a desired topology size. More specifically, our contributions are:

- We propose a simple heuristic to build graphs with arbitrary numbers of vertices based on best known DDP solutions. We demonstrate that the topologies generated by this approach lead to lower hop counts than random topologies generated using previously proposed methods. (Section III)
- We evaluate our proposed topologies and competing topologies (both random and non-random) for actual HPC applications using a discrete-event simulator. We find that topologies with low diameter and ASPL, i.e., the topologies proposed in this work lead to significant application performance gains when applications use collective all-to-all or irregular communications. (Section IV)
- We discuss practical concerns for topology deployment, demonstrating that the topologies generated by our approach do not exhibit significant increases in average cable length, cost and power consumption when compared to same-degree previously proposed fully random topologies (Section V).
- We demonstrate that our heuristic, albeit simple, provides graph with low hop counts and that is it likely difficult to design better heuristics. (Section VI)

Other sections in this paper are as follows. Section II reviews background information and related work. Section VII concludes with a summary of our findings.

II. Background and Related Work

Low-degree torus topologies are used commonly in supercomputers: 6 of the top 10 systems on the June 2015 Top500 list use torus networks. Supercomputers historically use low-degree k-ary n-cube network topologies. 2 of the top 10 systems use a high-degree fat-tree topology, and the remaining 2 use the high-degree Dragonfly topology [13].

The topologies proposed in this work are competitors to both these low-degree and high-degree topologies. Our topologies are “flat” and “irregular” so as to achieve low hop counts and high bisection even at low degree. An advantage of existing topologies with deterministic and regular structures is that the routing scheme can exploit the structure. Instead, our proposed topologies lack structure due to randomness. As a result, they require the use of routing/forwarding tables at each switch or source routing. Note that this is the approach used in InfiniBand and Ethernet, two technologies that are routinely used today in large-scale systems (e.g., in 81.2% of systems on the June 2015 Top500 list).

Because random graphs are known to achieve low diameters [5], several authors have recently proposed the use of random network topologies to achieve lower hop counts without necessarily using a high degree [15], [21]. Besides low hop counts, two related advantages of random topologies are that they can be generated for arbitrary numbers of switches and can be easily expanded to larger numbers of switches because they are unstructured [21]. The first advantage is important because there is an increasing trend for supercomputers to be designed for various network sizes. Recently proposed cable-geometric and floorplan designs can be combined to deploy network topologies with arbitrary numbers of switches [10], [13]. The second advantage is important because many data-centers and supercomputers are incrementally expanded year after year (depending on evolving resource demands and budget considerations) [21]. In this work we also advocate for using randomness, but we propose a novel topology generation method that achieves hop counts even lower that than of previously proposed fully random topologies.

On high-radix networks, DDP solutions can be used for reducing diameter, such as the SlimFly topology [4]. The SlimFly topology uses MMS graphs [17], which have diameter $k = 2$ and are known good solutions to the DDP. As a result, the SlimFly topology can only be constructed for particular numbers of switches, e.g., n=98, 242, 338, 578, 722, 1058, 1682, 1922, 2738 and 5618, for particular degrees d=11, 17, 19, 25, 29, 35, 43, 47, 55 and 79. Instead, in this work we propose generating topologies with arbitrary numbers of switches, using a good DDP solution as a starting point. Our work thus can be complementary to that in [4] since we can use SlimFly topologies as starting points.

III. Building Random Graphs From Known Low-diameter Graphs

Given a graph with degree d and diameter k, the number of vertices in the graph is at most $1 + d\sum_{i=0}^{k-1}(d-1)^i$ and $\sum_{i=0}^{k} d^i$ for graphs, respectively, which is termed the Moore bound. Researchers have attempted to find solutions to the DDP with numbers of vertices that approach the Moore bound. Only a few graphs are known that achieve this bound and are thus optimal solutions to the DDP [18]. Our goal is to generate graphs for arbitrary numbers of vertices while using known DDP solutions as starting points.

A. Graph Generation Heuristic

We wish to construct a graph with n' vertices where each vertex has degree d. Let the graph $G = (V, E)$ be a solution of the DDP with maximum degree d, diameter k, and n vertices. If $n = n'$ we simply return G. If $n' > n$ we *augment* G with $n' - n$ additional vertices. Conversely, if $n' < n$ we *reduce* G by $n - n'$ vertices. Algorithm 1 shows the pseudo-code for Augment and Reduce functions. Each function is called $|n - n'|$ times and uses a simple heuristic. Augment adds a vertex, new, to V (line 2). It then iterates over all vertices in V with degree $< d$ and at each iteration adds an edge between one such vertex and new. It returns at once when new has degree d (lines 4-7). Otherwise it removes a random edge from E (line 8) and repeats. Reduce simply removes a random vertex, ensuring that the graph is not partitioned.

Algorithm 1 Pseudo-code for Augment and Reduce

```
 1: procedure AUGMENT($V$, $E$, $d$)
 2:     $V \leftarrow V \cup \{new\}$
 3:     while true do
 4:         for $v \in V - \{new\}$ with $degree(v) < d$ do
 5:             $E \leftarrow E \cup (v, new)$
 6:             if $degree(new) == d$ then
 7:                 return $(V, E)$
 8:         $E \leftarrow E - \{randomElement(E)\}$

 9: procedure REDUCE($V$, $E$, $d$)
10:     while true do
11:         $v \leftarrow randomElement(V)$
12:         $V' \leftarrow V - \{v\}$
13:         $E' \leftarrow E - \{(x, y) | v \in \{x, y\}\}$
14:         if $connected(V', E')$ then
15:             return $(V', E')$
```

Once a graph has been augmented/reduced it is possible that some vertices have degree $< d$ (e.g., because the original graph G may have such vertices, because of edge and vertex removals). We post-process all our augmented/reduced graphs by randomly adding edges between vertices with degree $< d$ until no longer possible. The rationale is that HPC network topologies are typically regular graphs in which all vertices have the same degree. Furthermore, this postprocessing allows for fair comparisons between the graphs that we generate as they all have almost the same number of edges (and thus the networks would use almost the same number of links).

Given a desired number of vertices n', we select the best known DDP solutions with the smallest $n_{\text{high}} \geq n'$ and the largest $n_{\text{low}} \leq n'$ number of vertices. In other words, these are the two best known DDP solutions with numbers of vertices the closest to n' above and below. In some cases only a single such DDP solution exists (i.e., when n' is smaller or larger than the number of vertices in any best known DDP solution). We then augment/reduce each of these solutions with the above heuristics so as to obtain graphs with n' vertices, and we pick the graph that achieves the lowest diameter, breaking ties based on the ASPL. We term a graph generated by this approach a Modified DDP (MDDP) Solution. In this paper, all topology graphs that have a random component are generated with 10 trials, selecting the trial that has the lowest diameter, breaking ties based on the ASPL (the study in [15] shows that a small number of trials is sufficient).

B. Evaluation for Graphs

1) Hop count: A key question we want to answer is whether our approach produces graphs with low hop counts, and in particular hop counts lower than that of previously proposed fully random graphs. In particular, we wish to determine how the hop count gap between best known DDP solutions and our augmented/reduced graphs evolves as their size differences ($|n - n'|$) increase. The two relevant metrics for hop counts are diameter and ASPL.

In terms of diameter, expectedly we find that our approach does not provide improvements over best known DDP solutions. In other terms, augmenting a best known DDP solution always raises the diameter. Otherwise, our approach would have generated new best known DDP solutions. Comparing our approach to fully random graphs, we find that our approach can provide lower diameter in a neighborhood of best known DDP solutions. However, these improvements occur mostly for very low degree $d = 3$. For degree $d = 6$, these improvements occur only rarely, and for higher degrees they are simply never observed in our experiments. We conclude that our approach is likely not effective for improving topology diameter in practical settings.

While the diameter is an important metric, reducing the ASPL can have a large impact on application performance even if the diameter is not reduced (see results in Section IV). Figure 2 shows ASPL (in hops) vs. network size for (i) the Moore bound (Ideal); (ii) best known DDP solutions (DDP); (iii) fully random graphs (Random); and (iv) graphs generated using our proposed approach (MDDP). Only a few points are shown for DDP since these graphs are designed to have the largest size rather than to have the lowest diameter for a given

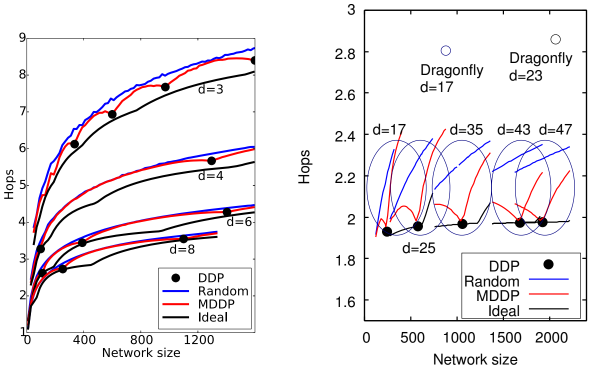

Figure 2. ASPL vs. network size for fully random graphs (Random), best known DDP solutions (DDP Solution), our proposed graphs (MDDP), and Moore bounds (Ideal). Results shown for various d values.

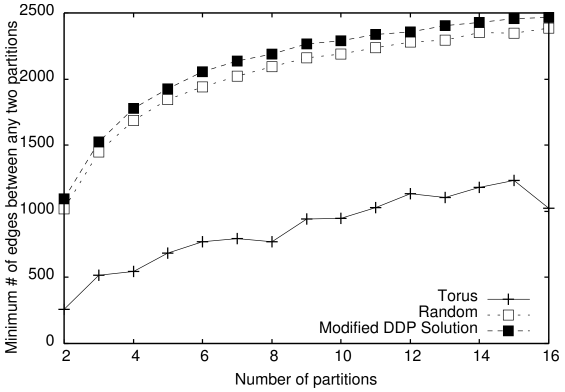

Figure 3. Minimum number of edges between any two same-size subgraphs vs. number of subgraphs for 1,024-vertex graphs.

graph size. Graphs with similar diameter but with smaller size exist but are not best known DDP solutions.

The left part of Figure 2 shows results for low degrees ($d = 3, 4, 6, 8$). As expected, DDP is relatively close to Ideal and MDDP and Random have higher ASPL. MDDP achieves lower ASPL than Random in the neighborhoods of DDP points. In other words, if one wishes to generate a graph with a size close to that of a best known DDP solution, then our approach is preferable to fully random graphs. When the desired graph size if far from that of best known DDP solutions, our approach leads to ASPL never larger than that of Random.

The right part of Figure 2 shows results for high degrees ($d = 17, 23, 25, 35, 43, 47$). For these results we pick best known DDP solutions with diameter 2, i.e., the MMS graphs used by the recently proposed SlimFly topology [4]. For comparison purposes, we also show data points for two Dragonfly topology configurations [13] ($d = 17, 23$). The observations are the same as for the low-degree results. Because MMS graphs are very close to the Moore bound, our approach leads to better relative improvement over Random. For instance, consider degree $d = 25$ for which the MMS graph has 578 vertices. Considering graphs with sizes within 10% of that size, MDDP leads to a 11% average relative improvement over Random. The Dragonfly topology has markedly higher ASPL than our other considered topologies, but a direct comparison is difficult since its graph is larger for a given degree. For instance, Dragonfly with degree $d = 17$ has 876 vertices and an ASPL of 2.81, while for the same number of vertices our approach has degree $d = 25$ and ASPL of 2.43, i.e., 13% lower. Whether this degree increase is worth the ASPL decrease depends on practical (e.g., budget) considerations. In this work we aim for low hop counts, closer to the Moore bound than that of previously proposed random topologies for a given degree.

We conclude that although our approach does not typically retain the low diameter of best known DDP solutions, it is able to produce graphs with ASPL lower than that of fully random graphs for both low and high degrees. ASPL degrades gracefully as the difference in network size with a best known DDP solution ($|n - n'|$) increases, eventually leading to ASPL similar to that of fully random graphs. Although these reductions in ASPL may seem modest (a few percent or on the order a few tenth of a hop), discrete event simulation experiments show that they can lead to substantial application performance increases (see Section IV).

2) Bisection: An important characteristic of a network topology is its bisection. The bisection is the minimum number of edges that connect two same-size subgraphs, which together form a partition of the graph. It is an important metric because it correlates to the achievable bisection bandwidth that would affect the performance of collective communication through the entire network. Bisection is typically computed using $p = 2$ subgraphs. For generality we compute the minimum number of inter-subgraph edges for all subgraph pairs among $p \geq 2$ same-size subgraphs. We perform this computation using METIS, which uses an efficient multi-level iterative approach [2]. We present results for low-degree scenarios. The topologies used as starting points by our approach in high-degree scenarios, i.e., MMS graphs, already achieve high network bisection [4].

Figure 3(a) shows the minimum number of inter-subgraph edges vs. p, for graphs with $n = 1,024$ vertices and degree $d = 8$. Results are presented for fully random graphs, our proposed graphs, and non-random 4-D torus graphs. As expected, random graphs achieve much higher minimum number of edges than the 4-D torus. We also see that our approach leads to a small improvement over fully random graphs. Note that in our experiments we have never seen a decrease in bisection due to random shortcutting, which likely indicates that such reduction happens with low probability. As a result, in practice one can generate a graph with desirable high bisection by invoking our graph augmenting or reducing procedure once, or perhaps just a few times.

IV. Parallel Application Performance Evaluation

While hop count and bisection, which relate to network latency and throughput, are important for evaluating and comparing network topologies, the ultimate metric is application performance. In this section we use discrete-event simulation to evaluate the performance of parallel applications and benchmarks when executed on our proposed topologies and previously proposed topologies.

A. Methodology

We use the SIMGRID simulation framework (v3.12) [20]. SIMGRID implements validated simulation models, is scalable, and makes it possible to simulate the execution of unmodified parallel applications that use the Message Passing Interface (MPI) [6]. We consider 2 case studies for different degrees, as described in Table 1 (which indicates ASPL and bisection values for the topologies in each case study).

Table 1. Case studies for simulation experiments (*d*: switch degree, 256 switches). For each graph we indicate its ASPL (in hops) and bisection (in number of edges).

	Case 1 (*d*=8)		Case 2 (*d*=17)	
	ASPL	Bis	ASPL	Bis
Torus	4.02	128	—	—
Rand	2.89	264	2.23	753
MDDP	2.75	310	2.03	726

For all MDDP topologies we use a shortest path routing scheme, i.e. Dijkstra's algorithm. To avoid deadlocks between paths, topology-agnostic virtual-channel transition routing is a well-studied approach for irregular bi-directional networks [9]. Each switch has a 100 nsec delay. Switches and hosts are interconnected together via links with 40 Gbps bandwidth. Each host has 100 GFlops. We configure SIMGRID to utilize its built-in version of the MVAPICH2 implementation of MPI collective communications [3].

We simulate the execution of the NAS Parallel Benchmarks (version 3.3.1, MPI versions) [22] (Class B for BT, CG, DT, LU, MG and SP, and Class A for FT and IS benchmarks), the matrix multiplication example provided in the SIMGRID distribution (MM), and the Graph500 benchmark (version 2.1.4) [23].

B. Evaluation Results

Figures 4 and 5 show normalized performance results for each benchmark for fully random topologies and our proposed topologies, normalized to the performance of the torus topology.

As expected, the network topology can have a large influence on application performance. Random and MDDP outperform the torus topology for those applications that rely on collective communication patterns (FT, IS and MM). On average, considering all experiments in the case study 1, when MDDP, resp. Random, improves over the torus topology, the improvement is 77%, resp. 79%. In case study 2 (Figure 5), for which the torus is not considered, MDDP is never outperformed by Random and outperforms it by 22% on average.

Overall, we find that our approach leads to higher application performance than the previously proposed fully random topology in almost all our experiments. When a non-random topology (i.e., torus) is preferred, then our proposed topology leads to a lower performance degradation than the previously proposed fully random topology. Conversely, when a random topology is preferred, then our proposed topology leads to better performance than the previously proposed fully random topology.

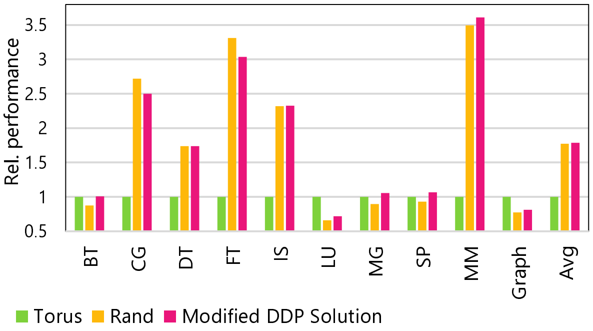

Figure 4. Case 1 (*d*=8) results.

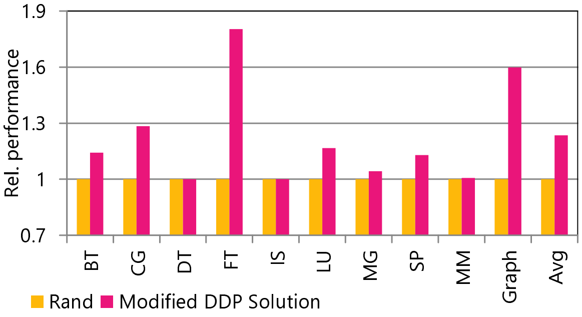

Figure 5. Case 2 (*d*=17) results.

V. TOPOLOGY CABLING

Practical concerns for a (large) network topology include cabling complexity and cable length. The 1st-generation Earth simulator supercomputer implemented a full-crossbar network using about 100,000 "fat" electric cables whose aggregate length reached 2,400km. This instance demonstrates that it is possible to deploy production systems with high complexity and high aggregate cable length. However, in general low complexity and length are desirable. Two other important practical concerns are topology cost and network power consumption. In this section we compare various topologies quantitatively from the cabling point of view when deployed in cabinets in a machine room. We first discuss cabling complexity qualitatively. We then estimate cable length, topology cost and network power consumption using the technology model used in [4].

A. Cabling Complexity

One difficulty with our proposed topologies is that the cabling complexity increases because of the lack of topology

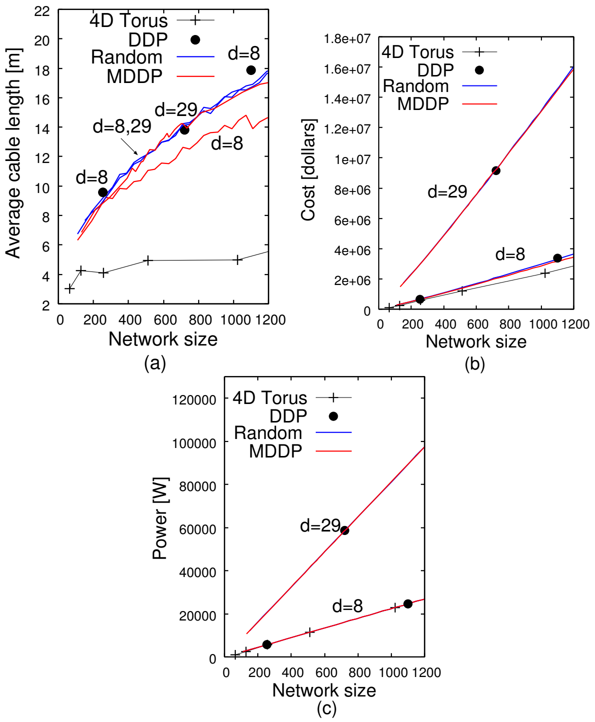

Figure 6. Average cable length (a), cost (b) and power (c) vs. network size for fully random graphs (Random), best known DDP solutions (DDP Solution), and our proposed graphs (MDDP).

structure. Note that the inter-cabinet cabling medium is optical, meaning that cables are "thin", while the intra-cabinet cabling medium may be electric, meaning that cables are "fat" as their bandwidth becomes large. Thinner optical cables can enable higher cabling density and can relax cable bending constraints. These two factors could mitigate the impact of our proposed network topology on cabling complexity.

One advantage of our proposed topology in terms of cabling complexity is that in the case of mis-cabling (human error) or even link failure, a topology-agnostic routing scheme can easily recover connectivity via path updates [9], [15]. This is because in random topologies there are many paths with low hop counts between each source-destination pair. Allowing approximate cabling may drastically shortens the installation time of supercomputers and datacenters.

B. Cable Length, Cost and Power

We assume a physical floorplan sufficiently large to align all cabinets on a 2-D grid. Formally, assuming m cabinets, the number of cabinet rows is $q = \lceil \sqrt{m}\, \rceil$ and the number of cabinets per row is $p = \lceil m/q \rceil$. We assume that each cabinet is 0.6m wide and 2.1m deep including space for the aisle. The distance between the cabinets is computed using the Manhattan distance. We estimate cable length, cost and power as in [4].

Figure 6(a) shows the average cable length of inter-switch cables. The DDP solution for $d = 29$ is the MMS graph used by the SlimFly topology. As expected, the torus topology has cable length several factors lower than that of all the other topologies. For a given degree all these other topologies have roughly the same average cable length. We conclude that although cable length is an important factor for choosing a topology, it is not a discriminating criterion for the topologies considered in this work.

Figure 6(b) shows network cost vs. network size for for degree $d = 8$ and $d = 29$. Switch degree strongly impacts the cost, whereas the cable length, whose medium is electric or optical, is not a dominant cost factor. Similar trends are obtained when using other cost models [19]. Figures 6(c) shows network power consumption vs. network size for degree $d = 8$ and $d = 29$. Overall, these results show that, for a given degree, all our considered topologies have similar cost and power consumption. Note that Dragonfly would have a similar curve to that of the MDDP and 4-D torus, as long as the degree and network sizes are the same.

Given the results in this section, we conclude that our proposed topology design has similar cable length, cost and power consumption when compared to previously proposed fully random graphs. This is perhaps not surprising, although it was conceivable that using best known DDP solutions as a starting point could have unexpected effects on these metrics when compared to fully random topologies.

VI. Toward Better Augmented/Reduced Graphs

The graph generation heuristics described in Section III are very simple, and in this section we explore whether better heuristics could be designed.

To assess the quality of our MDDP graphs, we compute all possible graphs for given degree and number of vertices using Graphillion [11]. For d=4 and n=10, there are 66,462,480 possible graphs and the lowest ASPL among these graphs is 1.56 hops. Our approach produces this same graph. For d=4 and n=14, the ASPL of the best graph is 1.69 hops, whereas our approach produces a graph with an ASPL of 1.71 hops. In both cases, the fully random graphs that we have generated have higher ASPL. This exploratory study, which can only be conducted for low n, shows that our heuristics, albeit simple, can produce graphs with low hop counts.

For large n vales, for all MDDP graphs considered in this work we perform a brute-force random local search to improve hop counts. Considering two edges (a, b) and (c, d), where a, b, c, and d are distinct vertices, we delete these edges and add two new edges: (a, d) and (c, b). We repeat this operation for batches of edges until no more improvements in hop counts are observed. For best known DDP solutions, improvements with this method are unlikely and in fact do not occur in our experiments. Occasional ASPL improvements can be achieved for some graphs, e.g., fully random graphs. However, we essentially see no improvements for our MDDP graphs. Although no strong conclusions can be drawn, these results indicate that it may not be straightforward to design better heuristics to augment or reduce best known DDP solutions.

The DDP is formulated as a maximization of the number of vertices for given degree and diameter bounds. As such, its solutions cannot be used directly to generate network topologies with arbitrary numbers of vertices, which is why we have proposed to augment/reduce best known DDP solutions. However, our approach would benefit from starting point graphs with size smaller than that of the best known DDP solution. Even though the largest known graph with degree d and diameter k has, say, n vertices, our method would likely benefit from using a graph with degree d and diameter k with αn vertices ($\alpha < 1$). In fact, such a smaller graph could be a previously best known DDP solution that has been replaced by a better solution. As of today, only few such smaller graphs are easily identified. In other words, it is better to start from a good (but not best known) DDP solution with a number of vertices closer to our target rather than to start from a best known DDP solution that has a number of vertices further away from our target.

VII. Conclusions

Random network topologies have recently been proposed for HPC systems because they achieve drastically better tradeoffs between degree and hop counts (diameter, ASPL) than traditionally used non-random topologies, while achieving comparable or higher bisection. Random topologies can lead to performance slowdowns for parallel applications with regular workloads, and in fact could preclude many performance tuning optimizations due to the lack of topology structure. Conversely, they can boost the performance of parallel applications with irregular communication patterns and/or with heavy use of collective communications. Furthermore, topology-aware performance tuning for such applications is notoriously difficult. Although previous work has proposed the use of fully random topologies, in this paper we have instead explored extreme low-latency network designs that use randomness but also attempt to approach the Moore bound. More specifically, we propose using best known solutions to the Degree Diameter Problem (DDP) as starting points, using simple heuristics to augment/reduce these solutions so as to generate topologies for arbitrary numbers of vertices.

Our results show that our proposed topologies achieve lower hop counts than previously proposed fully random topologies. In addition, like fully random topologies, our proposed topologies achieve high bisection due to the use of unstructured random shortcutting. Discrete-event simulations of parallel application benchmarks show that our proposed topologies outperform previously proposed fully random topologies and, in many cases, state-of-the-art torus topologies. Because some of these applications are latency-sensitive, even small ASPL improvements in the network topology can lead to significant application performance improvements.

Acknowledgment

This work was partially supported by MIC/SCOPE(152103004) and JST CREST.

References

[1] "Graph Golf: the degree-order problem competition." [Online]. Available: http://research.nii.ac.jp/graphgolf/

[2] "METIS - Serial Graph Partitioning and Fill-reducing Matrix Ordering." [Online]. Available: http://glaros.dtc.umn.edu/gkhome/metis/metis/overview

[3] "MVAPICH: MPI over InfiniBand, 10GigE/iWARP and RoCE." [Online]. Available: http://mvapich.cse.ohio-state.edu/

[4] M. Besta and T. Hoefler, "Slim Fly: A Cost Effective Low-diameter Network Topology," in *Proc. of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014, pp. 348–359.

[5] B. Bollobás and F. R. K. Chung, "The Diameter of a Cycle Plus a Random Matching," *SIAM J. Discrete Math.*, vol. 1, no. 3, pp. 328–333, 1988.

[6] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, "Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2899–2917, 2014.

[7] Combinatorics Wiki, "The Degree Diameter Problem for General Graphs," http://combinatoricswiki.org/wiki/The_Degree_Diameter_Problem_for_General_Graphs.

[8] P. Costa, H. Ballani, and D. Narayanan, "Rethinking the Network Stack for Rack-scale Computers," in *6th USENIX Workshop on Hot Topics in Cloud Computing, HotCloud '14*, 2014.

[9] J. Flich, T. Skeie, A. Mejia, O. Lysne, P. Lopez, A. Robles, J. Duato, M. Koibuchi, T. Rokicki, and J. C. Sancho, "A Survey and Evaluation of Topology Agnostic Deterministic Routing Algorithms," *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 405–425, 2012.

[10] I. Fujiwara, M. Koibuchi, H. Matsutani, and H. Casanova, "Skywalk: a Topology for HPC Networks with Low-delay Switches," in *IEEE International Symposium on Parallel and Distributed Processing (IPDPS)*, May 2014, pp. 263–272.

[11] T. Inoue, H. Iwashita, J. Kawahara, and S. ichi Minato, "Graphillion: Software Library Designed for Very Large Sets of Labeled Graphs," *International Journal on Software Tools for Technology Transfer, Springer*, Oct. 2014.

[12] K. Scott Hemmert and Jeffrey S. Vetter and Keren Bergman and Chita Das and Azita Emami and Curtis Janssen and Dhabaleswar K. Panda and Craig Stunkel and Keith Underwood and Sudhakar Yalamanchili, "Report on Institute for Advanced Architectures and Algorithms, Interconnection Networks Workshop 2008."

[13] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-Driven, Highly-Scalable Dragonfly Topology," in *ISCA*, 2008, pp. 77–88.

[14] M. Koibuchi, I. Fujiwara, H. Matsutani, and H. Casanova, "Layout-conscious Random Topologies for HPC Off-chip Interconnects," in *HPCA*, 2013, pp. 484–495.

[15] M. Koibuchi, H. Matsutani, H. Amano, D. F. Hsu, and H. Casanova, "A Case for Random Shortcut Topologies for HPC Interconnects," in *ISCA*, 2012, pp. 177–188.

[16] S. Matsuoka, H. Sato, O. Tatebe, F. Takatsu, M. A. Jabri, M. Koibuchi, I. Fujiwara, S. Suzuki, M. Kakuta, T. Ishida, Y. Akiyama, T. Suzumura, K. Ueno, H. Kanezashi, and T. Miyoshi, "Extreme Big Data (EBD): Next Generation Big Data Infrastructure Technologies Towards Yottabyte/Year," *Supercomputing Frontiers and Innovations*, vol. 1, no. 2, pp. 89–107, 2014.

[17] B. McKay, M. Miller, and J. Širán, "A note on large graphs of diameter two and given maximum degree," *Journal of Combinatorial Theory*, vol. 74, no. 1, pp. 110–118, 1998.

[18] M. Miller and J. Siran, "Moore graphs and beyond: A survey of the degree/diameter problem," *Electronic Journal of Combinatorics*, 2013, Dynamic Surveys.

[19] J. Mudigonda, P. Yalagandula, and J. Mogul, "Taming the flying cable monster: A topology design and optimization framework for data-center networks," *USENIX ATC*, pp. 1–14, 2011. [Online]. Available: http://static.usenix.org/events/atc11/tech/final_files/Mudigonda.pdf

[20] SimGrid: Versatile Simulation of Distributed Systems, http://simgrid.gforge.inria.fr/.

[21] A. Singla, C.-Y. Hong, L. Popa, and P. B. h. Godfrey, "Jellyfish: Networking Data Centers Randomly," in *NSDI*, 2012, pp. 225–238.

[22] The NAS Parallel Benchmarks, http://www.nas.nasa.gov/Software/NPB/.

[23] Top 500 Sites, http://www.graph500.org/.