Comparing global link arrangements for Dragonfly networks

Emily Hastings, David Rincon-Cruz, Marc Spehlmann,
Sofia Meyers, Anda Xu, David P. Bunde
Department of Computer Science
Knox College
Galesburg, IL, USA
{*emhastings,drinconcruz,mspehlma,semeyers,axu,dbunde*}*@knox.edu*

Vitus J. Leung
Discrete Math & Optimization Department
Sandia National Laboratories
Albuquerque, NM, USA
vjleung@sandia.gov

***Abstract*—High-performance computing systems are shifting away from traditional interconnect topologies to exploit new technologies and to reduce interconnect power consumption. The Dragonfly topology is one promising candidate for new systems, with several variations already in production. It is hierarchical, with local links forming groups and global links joining the groups. At each level, the interconnect is a clique, with a link between each pair of switches in a group and a link between each pair of groups.**

This paper shows that the intergroup links can be made in meaningfully different ways. We evaluate three previously-proposed approaches for link organization (called global link arrangements) in two ways. First, we use bisection bandwidth, an important and commonly-used measure of the potential for communication bottlenecks. We show that the global link arrangements often give bisection bandwidths differing by 10s of percent, with the specific separation varying based on the relative bandwidths of local and global links. For the link bandwidths used in a current Dragonfly implementation, it is 33%. Second, we show that the choice of global link arrangement can greatly impact the regularity of task mappings for nearest neighbor stencil communication patterns, an important pattern in scientific applications.

***Keywords*-Dragonfly, parallel interconnect topology, bisection bandwidth, task mapping**

I. Introduction

There are several key challenges that must be addressed to continue improving the capabilities of high-performance computing (HPC) systems [1]. One of these challenges concerns the node interconnect; simply scaling up traditional interconnects will not work because such a network would require too much power and face severe bandwidth limitations as an increasing number of nodes attempt to split link bandwidth between more communicating pairs.

New interconnect topologies are being developed to address these challenges by exploiting two technology changes. The first change is the availability of *high-radix routers* [2], which use increasing bandwidth to increase the number of ports rather than the bandwidth per port. This large number of ports allows many destinations to be within a few hops. Reducing the number of hops saves energy since less equipment handles each message and reduces contention since each message consumes bandwidth on fewer links.

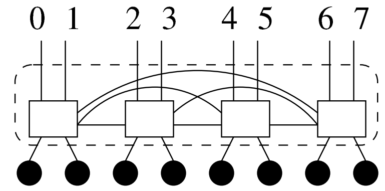

Figure 1. Group of switches forming a virtual switch

The second technology change is economical optical links. Optical links can be longer than electrical ones, enabling long distances to be covered in fewer hops, providing room for all the equipment in an HPC system. They also require less energy per unit distance that the message travels.

Several systems using high-radix routers have been proposed [3], [4], [5], [6]. Our focus here is the Dragonfly topology [6], which has been attracting significant academic interest and variations of which have recently been used to implement the Cray XC [7], [8] and PERCS [9].

A Dragonfly system is constructed from switches that are organized into groups. Each switch has nodes attached to it using traditional electrical links. Electrical links also connect every pair of switches in a group. In addition to these electrical links, each switch has outgoing optical links that connect to switches in other groups. This is done so that every group has a link to each other group. Effectively the switches of a group form a *virtual switch* of very high radix connecting all nodes associated with its group to the virtual switches associated with each other group. Figure 1 shows a virtual switch constructed from 4 physical switches. This connects 8 nodes (the circles) to 8 outgoing ports for optical links. Figures 2, 3, and 4 show three ways that 9 of these virtual switches can be connected into a Dragonfly system.[1] Each virtual switch is marked with a dashed line circling the physical switches in its group. To simplify the figures, they do not depict the electrical links or the nodes themselves.

Because of their different roles, the electrical links in a Dragonfly system are called *local links* while the optical

[1] The figures depict a $(p, 4, 2)$-Dragonfly, meaning each group has 4 switches, each with 2 optical links. The notation is defined in Section II.

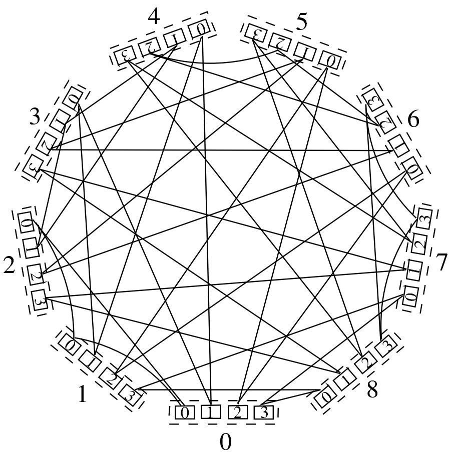

Figure 2. Absolute global link arrangement for $(p, 4, 2)$

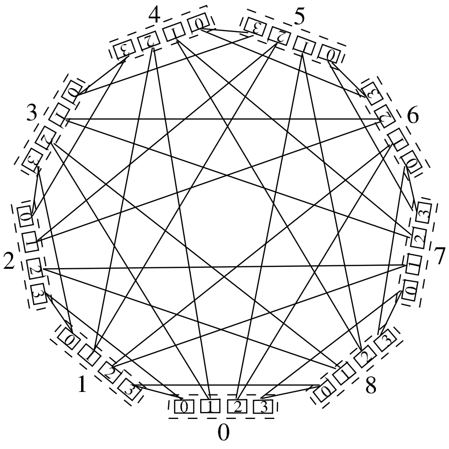

Figure 3. Relative global link arrangement for $(p, 4, 2)$

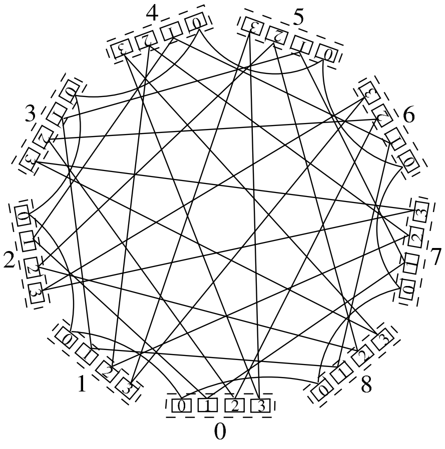

Figure 4. Circulant-based global link arrangement for $(p, 4, 2)$

ones are called *global links*. In a Dragonfly system, the shortest path between any pair of switches is at most three hops: one local link within the source group, one global link between groups, and one local link within the destination group. These shortest path routes are called *direct routes*. When direct routes are overly congested, an *indirect route* is used instead. These are found using Valiant's randomized routing algorithm [10], which first sends the message to a randomly-chosen intermediate group. Even with indirect routing, all messages traverse at most five hops.

Despite the interest in Dragonfly, the existing literature virtually ignores the issue of which switch in a group is connected to each of the other groups. In nearly all prior work, the analogs of Figures 2–4 uses ellipses that hide how the actual connections are made. The main exception is Camarero et al. [11], who define three specific ways to make these connections, called a *global link arrangement*. Even in this case, however, the global link arrangements are not compared and their simulations all use one arrangement.

Contribution: In this paper, we demonstrate that the choice of global link arrangement changes the achievable network performance. Specifically, we take the arrangements defined by Camarero et al. [11] (absolute, relative, and circulant-based)[2] and evaluate them in two ways.

First of all, we evaluate the global link arrangements in terms of their effect on *bisection bandwidth*, the minimum bandwidth between 2 equal-sized parts of the system. This is a common network quality measure (e.g. [12, pg. 661]); it tries to capture the worst-case communication bottleneck in a large computation. It is also particularly suitable for Dragonfly networks since the use of randomized indirect routing will tend to equalize the link utilizations. We show that the choice of global link arrangement can have great impact on the bisection bandwidth by solving for it on a small Dragonfly network (36 switches) and demonstrating that the circulant-based and relative arrangements can deliver up to a 50% higher bisection bandwidth than the absolute arrangement. The specific improvement depends on the relative bandwidth of global and local links, but a 33% improvement is realized with the links used in PERCS.

In addition to showing the importance of global link arrangements on this small network, we analyze global link structure in general Dragonfly networks and identify some cases where the bisection bandwidth depends on global links and others where it is determined entirely by local links.

Our second approach to evaluating the global link arrangements is in terms of their effect on task mapping. *Task mapping* is the assignment of a job's tasks to the processing elements assigned to that job. Quality task mapping has been shown to improve application performance in a variety of settings by improving bandwidth utilization

[2] Camarero et al. [11] used the names *consecutive arrangement* and *palmtree arrangement* for what we call the *absolute arrangement* and *relative arrangement*.

(e.g. [13], [14], [15]). This paper lays out criteria for task mapping in Dragonfly networks. It also considers mapping for an important communication pattern under all three of the global link arrangements and shows that, while good mappings are possible under each, the relative arrangement allows mappings with a regularity that will simplify code and improve generalizability.

Organization: The rest of this paper is organized as follows. Section II defines terminology related to the Dragonfly interconnect and the global link arrangements. Section III gives the comparison of the global link arrangements using bisection bandwidth on the small network. Section IV examines the structure of global links for general networks and uses this structure to draw conclusions about the bisection bandwidth for each arrangement. Section V presents the comparison of the arrangements based on task mapping. Section VI summarizes related work. Finally, Section VII concludes and discusses future work.

II. Definitions

The size of a Dragonfly system is described with the following three parameters:

p the number of nodes connected to a switch,
a the number of switches in a group, and
h the number of optical links on a switch.

These three parameters immediately also determine

g $= ah + 1$, the number of groups in the network

since each of the a switches in a group connects to h other groups. The number of nodes is $pag = pa(ah + 1)$. The system shown in Figure 3 is called a $(p, 4, 2)$-Dragonfly since its parameters are $(p, a, h) = (p, 4, 2)$; the nodes are not shown so p is left as a variable. In this paper, we assume $h \geq 2$ because otherwise all global link arrangements are isomorphic.

We name each switch with an ordered pair (i, j), where i is the group number and $0 \leq j < a$ is the switch number within that group. We also refer to switch (i, j) as switch j of group i. For simplicity of notation, we use modular arithmetic on group numbers so that (i, j) refers to the same switch as $(i + g, j)$, $(i - g, j)$, and similar pairs.

As mentioned above, one of the methods we use to evaluate Dragonfly networks is bisection bandwidth. A *cut* in such a network is a division of the network switches into two sets. A *bisection* is a cut in which the two sets have equal size (within 1 when the number of switches is odd). In this paper we use these terms interchangeably because all cuts considered are bisections. We specify cuts by describing one of these sets. The *bandwidth of a cut* is the bandwidth of all links with one endpoint in each side; these links are said to be *crossing* the cut. To deal with the presence of two types of links in Dragonfly networks, we assign local and global links bandwidth 1 and α respectively. The network's *bisection bandwidth* is the smallest bandwidth achieved by any bisection of the network.

To look at the effect of global link arrangements, we consider the following natural ideas:

1) The *absolute arrangement* conceptually connects port k of each group's virtual switch to group k. Once we ignore the unnecessary link that this would give each group to itself, this arrangement connects port k of group i to group k if $k < i$ and to group $k + 1$ otherwise. The absolute arrangement is depicted in Figure 2.
2) In the *relative arrangement*, each group uses its port 0 to connect to the "next" group, its port 1 to two groups down, etc. More formally, port k of group i ($k \in \{0, \dots, g-2\}$ and $i \in \{0, \dots, g-1\}$) connects to group $(i + k + 1)$. This is the absolute global link arrangement with each group using its own numbering for the groups so it takes the role of group 0. The relative arrangement is depicted in Figure 3.
3) The *circulant-based arrangement* assumes that h is even, i.e. that every switch has an even number of optical links. Each group uses its port 0 to connect to the next group, its port 1 to the previous group, its port 2 to the group two ahead, its port 3 to the group two behind, etc. More formally, port k of group i ($k \in \{0, \dots, g-2\}$ and $i \in \{0, \dots, g-1\}$) connects to group $(i+k/2+1)$ if k is even and group $(i - \lfloor k/2 \rfloor - 1)$ if k is odd. The circulant-based arrangement is depicted in Figure 4. Unlike with the other arrangements, links in the circulant-based arrangement always connect like-numbered switches; switch (i, j) has ports $k \in \{hj, hj + 1, \dots, h(j + 1) - 1\}$, linking to switches $(i \pm (\lfloor k/2 \rfloor + 1), j)$ for these values of k.

Each of these global link arrangements make it fairly easy to determine which physical switch connects to each group, simplifying both routing and network installation.

III. Bisection bandwidth for $(p, 4, 2)$

To illustrate the separation between the different global link arrangements, we determine the bisection bandwidth for each on the $(p, 4, 2)$-Dragonfly network. For a given global link arrangement and cut, the bandwidth varies linearly with α, with the rate depending on the number of global links crossing the cut. Because the bandwidth of each cut varies with α, the min-bandwidth cut also changes. For small values of α, the min-bandwidth cut is one crossed by the minimum number of local links, even if it is crossed by many global links. As α grows, the global links become more important and eventually the min-bandwidth cut is one crossed by the minimum number of global links. Thus, the bisection bandwidth is a piecewise linear function of α, with slope changes whenever the min-bandwidth cut changes to one crossed by fewer global links. This function is the lower envelope of the functions determined by the different min-bandwidth cuts.

We used a divide and conquer algorithm to identify the bisection bandwidth function for a specific Dragonfly network and global link arrangement. For a given value of α, the min-bandwidth cut is found with a branch and bound strategy. (Finding the min-bandwidth cut is NP-hard, though poly-time approximable to within an $O(\log^2 n)$ factor [16].) The algorithm begins by doing this for $\alpha = 0$ and a large value of α. Each of the resulting cuts corresponds to a linear function of α. Then it determines the cut for the value of α at which these functions intersect. If the min-bandwidth cut for this α gives the same value as the known cuts, this value of α is the transition between them. Otherwise, we have found a cut with less bandwidth for this α so the computation proceeds by finding the intersections of the functions corresponding to this cut and the previously-known cuts. This proceeds until all the transition points are confirmed.

The branch and bound portion of this algorithm is computationally expensive, which restricts the size of Dragonfly networks whose bisection bandwidth can be practically determined by this method. We were able to run it on a $(p, 4, 2)$-Dragonfly, however. At this size, each part has 18 switches. Figure 5 shows the bisection bandwidth for each global link arrangement. The functions are the same until $\alpha = 1.25$, after which the absolute arrangement provides a constant bisection bandwidth of 24 while the others continue to improve until reaching bandwidth 36. The circulant-based arrangement continues to improve at a higher rate than the relative one; it reaches 36 at $\alpha = 8/3$ while the relative arrangement only does so at $\alpha = 4$. Overall, the circulant-based arrangement always provides at least as high a value as the relative arrangement, which in turn is always at least as high as absolute. The circulant-based arrangement is strictly better than relative for $1.25 < \alpha < 4$. Both circulant-based and relative are strictly better than absolute for $\alpha > 1.25$, and they provide 50% higher bisection bandwidth for $\alpha \geq 8/3$ and $\alpha \geq 4$ respectively.

For context, consider the value of α used by the current implementations of (variations on) Dragonfly. In the Cray XC, global links consist of four parallel 4.7 GB/s optical cables and local ones are provided by a triple of 5.25 GB/s electrical cables [7]. Based on these bandwidths, the Cray XC has $\alpha = 4(4.7)/3(5.25) \approx 1.19$. At this value, all three arrangements provide the same bisection bandwidth.

The other implementation is PERCS, which uses 2 different bandwidths for local links. Depending on which is selected, the system's value of α is either $10/24 \approx 0.42$ or $10/5 = 2$. All three arrangements give the same bisection bandwidth for the former value, but the circulant-based and relative arrangements give improvements over absolute of approximately 33% and 17% respectively for the latter.

Thus, we see that the importance of the global link arrangement depends on α, but that it can be quite significant for practical values.

Absolute arrangement: Now we describe the cuts that achieve the bisection bandwidths depicted in Figure 5, beginning with those for the absolute global link arrangement. One cut that achieves bandwidth $4 + 16\alpha$ for the absolute arrangement takes the first 18 switches: all of groups 0–3, plus switches 0 and 1 of group 4. This is crossed by 4 local links because group 4 has two switches on each side of the cut. The 16 global links crossing the cut are those from groups 0–3 to groups 5–8; note that all global links from switches 0 and 1 of group 4 go to the first four groups.

One cut that achieves bandwidth 24 takes all of groups 0–1, switches 0–1 of groups 2–4, and switch 0 of groups 5–8. No global links cross this cut because groups 0–1 are connected to switch 0 in all the other groups and the switch 1s in groups 2–4 connect to each other. As for local links, none cross the cut in groups 0–1, 4 in each of groups 2–4 (2 switches on each side), and 3 each in groups 5–8 (1 switch on one side and 3 on the other).

Relative arrangement: Now we describe the cuts that achieve the bisection bandwidth values depicted in Figure 5 for the relative global link arrangement.

One cut that achieves bandwidth $4 + 16\alpha$ is to take all of groups 0–3, plus switches 2 and 3 of group 4. This is the same as the cut of the same bandwidth for the absolute arrangement except for taking the other half of group 4 since that is the half that connects to groups 0–3 in the relative arrangement. The bandwidth calculation is essentially the same as well, with 4 local edges crossing the cut because group 4 is split and 16 global edges crossing it between groups 0–3 and 5–8.

One cut that achieves bandwidth $14 + 8\alpha$ is to take all of groups 0 and 5, switches 1–3 of group 1, switches 0–2 of group 4, switches 1 and 3 of group 6, and switches 0 and 2 of group 8. A partial description of this cut is to not take any of groups 2 or 3 and then try to exclude the switches connecting to them (hence not including switch 0 of group 1, switch 3 of group 4, or switch 2 of group 6).

One cut that achieves bandwidth $20 + 4\alpha$ is to take all of groups 0 and 1, switches 0, 2, and 3 of group 2, switches 2–3 of group 3, switch 2 of group 4, switch 1 of group 6, switch 1 of group 7, and switches 0–1 of group 8. A partial description of this cut is to not take any of group 5 and to try to exclude the switches connecting to it.

One cut that achieves bandwidth 36 is to take switches 0 and 3 from each group. Global links join switch 0 with switch 3 in the previous two groups and switch 3 with switch 0 in the next two groups so none of these links cross the cut. Four local links cross the cut in each group since half the group's switches are on each side of the cut.

The series of min-bandwidth cuts defy simple description, but the general trend is that each cut in the series excludes fewer groups (4 initially, then 2, then 1, and finally none) in order to take both endpoints of more global links. In exchange, each cut in the series increases the number of

Range of α	Bandwidth by global link arrangement		
	Absolute	Relative	Circulant-based
0 – 1.25	$4+16\alpha$	$4+16\alpha$	$4+16\alpha$
1.25 – 1.5	24	$14+8\alpha$	
1.5 – 2		$20+4\alpha$	$16+8\alpha$
2 – $2\frac{2}{3}$			$20+6\alpha$
$2\frac{2}{3}$ – 4			36
>4		36	

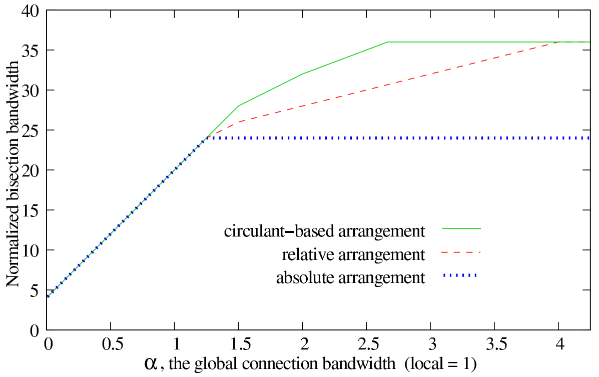

Figure 5. Normalized bisection bandwidth for the global link arrangements. For one cut, the bandwidth is (# local links) $+ \alpha$(# global links). The table (left) shows the bandwidth of each arrangement's min-bandwidth cut as a function of α. The graph (right) plots these.

crossing local links since it includes more partial groups.

Circulant-based arrangement: Now we describe the cuts that achieve the bisection bandwidth values depicted in Figure 5 for the circulant-based global link arrangement.

One cut that achieves bandwidth $4+16\alpha$ is to take all of groups 0, 1, 4, and 5, plus the middle two switches of group 7. The chosen switches in group 7 are those with links to the others so the only global links crossing the cut are from the 4 chosen groups to the remaining 4 groups.

One cut that achieves bandwidth $16+8\alpha$ is to take all of groups 0 and 5, switches 0–2 of groups 6 and 8, switches 0–1 of group 7, and switch 2 of groups 2 and 3. The higher score than the $14+8\alpha$ cut for the relative arrangement occurs because a group with two switches has been replaced with two groups of one switch each.

One cut that achieves bandwidth $20+6\alpha$ is to take all of group 0, switches 0, 1, and 3 of groups 1 and 2, switches 0, 1, and 2 of group 3, switches 1 and 3 of group 5, switches 2 and 3 of group 6, and switch 1 of group 7.

One cut that achieves bandwidth 36 is to take switches 0 and 1 from each group. (Any 2 switches work as long as the same ones are taken in each group.) Because every global link connects like-numbered switches, none cross the cut.

As with the relative arrangement, the series of min-bandwidth cuts defies simple description. It has a similar trend of excluding fewer groups, though the $20+6\alpha$ cut violates the trend; the number of excluded groups is 4, 2, 2, and then none.

IV. Good cuts for large α

Now that we have completely determined the bisection bandwidth for $(p, 4, 2)$, we turn to larger Dragonfly instances. Without a concrete configuration, we cannot completely solve for bisection bandwidth. Instead, we examine a particular case, that of large values of global bandwidth α. Currently, most concern about hotspots in Dragonfly systems focuses on the global edges, suggesting large values of α. Our analysis shows that the benefits of this are limited in many cases.

For sufficiently large α, the bisection bandwidth is determined by the bisection crossed by the fewest global links. Our approach to finding this cut is to identify the connected components in the graph of global links. Formally, a *globally connected component* (GCC) is a connected component of the network with only global links, ignoring local links.

A. Circulant-based arrangement

The easiest case is the circulant-based arrangement. Because every global link in this arrangement connects a pair of like-numbered switches, every network has at least a GCCs, one per switch number. When a is even, this guarantees the presence of bisections that are not crossed by global links by just taking half of each group as one part:

Observation 1: When a is even and α is sufficiently large, the circulant-based arrangement's bisection bandwidth is $(a/2)^2 g$.

The structure of the GCCs is potentially more complicated than just having one per switch number. A single switch number can be split into more than one GCC if g is a multiple of the distance traversed by the switch's links. Figure 4 shows this; switch 2 connects to groups three down and there are nine groups in all so the 2nd switches form three GCCs, each consisting of the 2nd switches in an equivalence class of groups mod 3.

B. Relative arrangement

It is not obvious that the other arrangements would create more than a single GCC, but both actually yield GCCs with a regular structure. We first solve for the structure of the relative arrangement because it is simpler.

Theorem 2: When a is even, the relative arrangement gives $a/2$ GCCs of size $2g$. If a is odd, then there are $\lfloor a/2 \rfloor$ GCCs of size $2g$ and one GCC of size g.

Proof: In the relative arrangement, the global links from a group's 0^{th} switch join it to the last (aka $(a-1)^{\text{st}}$) switch in the next h groups. Similarly, the global links from a group's last switch join it to the 0^{th} switch in the previous h groups. Thus, the GCC containing $(0,0)$ will contain only 0th and $(a-1)$st switches. In fact, this GCC contains all such switches because they all occur in the path $(0,0),(h,a-1),(1,0),(h+1,a-1),(2,0),(h+2,a-1),(3,0),(h+3,a-1)$ etc. (Recall our assumption that $h \geq 2$.) Since it has 2 switches from each group, this GCC has size $2g$.

A similar argument shows that all the i^{th} and $(a-1-i)^{\text{th}}$ switches form a group for each $i \leq \lfloor a/2 \rfloor$; for any group j, switch (j,i) is connected to switch $a-1-i$ in groups $j+hi, j+hi+1, \ldots, j+hi+(h-1)$ and switch $(j, a-1-i)$ is connected to switch i in groups $j-ih-1, j-ih-2, \ldots, j-ih-h$. As above, each of these GCCs has size $2g$ since it contains 2 switches from each group.

The above gives $\lfloor a/2 \rfloor$ GCCs of size $2g$. Thus, we are done when a is even. When a is odd, the middle switches of each group form a GCC of size g since all are on the path $(0,\mu),(h\mu,\mu),(1,\mu),(h\mu+1,\mu),\ldots$, where $\mu = \lceil a/2 \rceil$. ■

Theorem 2 not only gives the GCCs sizes, but shows that all the GCCs in the relative arrangement have similar structure. (With one degenerate GCC when a is odd.) This structure lets us reason about the bisection bandwidth at large α. To simplify the discussion, we use *large GCC* to denote a GCC of size $2g$ and *small GCC* to denote one of size g. In addition, let $x \equiv_b y$ mean that x is congruent to y modulo b, i.e. that $x = y + bk$ for some integer k.

The simplest case is when $a \equiv_4 0$, i.e. a is a multiple of 4. In this case, there are an even number of large GCCs and no small ones. For sufficiently large α, the bisection bandwidth will be determined by a cut with half the GCCs on each side. Since every GCC includes 2 switches in each group, every bisection places $a/2$ switches in every group on each side of the cut. Thus, we have the following:

Corollary 3: When a is a multiple of four and α is sufficiently large, the relative arrangement's bisection bandwidth is $(a/2)^2 g$.

For other values of a, every bisection is crossed by at least one global link. When a is odd, the network has a single small GCC. When $a \equiv_4 2$, it has an odd number of large GCCs. In either case, any bisection is crossed by at least one global link so the bisection bandwidth includes α with a non-zero coefficient. This gives the following:

Corollary 4: When a is not a multiple of four, the relative arrangement's bisection bandwidth is $\Theta(\alpha)$.

C. Absolute arrangement

Now we consider the absolute arrangement, whose GCCs have a more complicated structure, with two qualitatively-different types.

Theorem 5: The absolute arrangement gives $\binom{a}{2} = a(a-1)/2$ GCCs of size $2h$ and a GCCs of size $h+1$.

To simplify the discussion, we overload the terms *large GCC* and *small GCC* to specify GCCs of these sizes.

Proof: The proof relies on classifying each switch into one of three types. We call the first type *skip switches* because they are mostly the switches with the virtual switch port that would connect to their group number and thus we 'skip' that group number. Formally, we define a skip switch as one in

$$Q = \{(i,j) : jh \leq i \leq (j+1)h\}$$

This slightly deviates from the intuitive description above because we also count as a skip switch one whose group's port number occurs immediately after it.

Note that each switch number $0 \leq j < a$ occurs in $h+1$ skip switches $((jh,j),(jh+1,j),\ldots,((j+1)h,j))$ since $(j+1)h$ is always less than $ah+1$, the number of groups. We will show that these switches form a clique of size $h+1$ for each of the a switch numbers. Thus, they form the small GCCs.

To prove that the skip switches with a given switch number form a clique, consider the groups to which a switch $(i,j) \in Q$ connects. The virtual port numbers on this switch are jh through $(j+1)h-1$. Since (i,j) is a skip switch, the transition between port k connecting to group k and port k connecting to group $k+1$ occurs during this switch. Thus, the first group these ports connect to is jh (unless this group is skipped because $jh = i$) and the last is $((j+1)h-1)+1 = (j+1)h$ (unless $(j+1)h = i$). Therefore, (i,j) connects to groups $N^Q_{i,j} = \{k : jh \leq k \leq (j+1)h, k \neq i\}$.

Suppose $i' \in N^Q_{i,j}$. Then (i',j) is a member of Q and thus also a skip switch. By the argument above, it connects to groups in the set $N^Q_{i',j} = \{k : jh \leq k \leq (j+1)h, k \neq i'\}$, which includes i. Because (i,j) and (i',j) each have a link to the other's group and there is only one link between each pair of groups, these two switches are connected. The choice of i' was arbitrary except for its membership in $N^Q_{i,j}$ so all skip switches with a switch number j are connected, completing the proof of our claim about skip switches.

The other two types of switches are defined in relation to skip switches. Switches are of type P if they come before their group's skip switch(es) and of type R if they come after. Formally, these types are defined by membership in the following sets:

$$\begin{aligned} P &= \{(i,j) : i > (j+1)h\} \\ R &= \{(i,j) : i < jh\} \end{aligned}$$

Note that P, Q, and R partition the set of switches.

We will start this proof by showing that type P switches cannot be adjacent to other type P switches. Let $p = (i,j)$ be a type P switch. It has ports jh through $(j+1)h-1$. Since the switch's group number is above this range, each of these ports has a link to the group with the same number and (i,j) has links to groups in the set $N^P_{i,j} = \{g : jh \leq g < (j+1)h\}$. By the definition of P, $i > (j+1)h$. Thus, i is greater than every member of $N^P_{i,j}$ and p has links only to lower-numbered groups. Because our only assumption about p was its membership in P, we conclude that type P switches are only adjacent to lower-numbered groups. This implies that no pair of type P switches can be adjacent since otherwise whichever has the smaller group number would be adjacent to a higher-numbered group.

Since skip switches are adjacent only to other skip switches, type P switches are not adjacent to them either. Thus, members of P are adjacent only to members of R. A similar argument shows that members of R are adjacent only to members of P.

In fact, for each pair of switch numbers j_1 and j_2 satisfying $0 \leq j_1 < j_2 < a$, there are h switches of type P with switch j_1 and h switches of type R with switch j_2 that form a GCC that is a complete bipartite graph connecting these switches. These account for the larger GCCs so proving this claim will complete the theorem.

For a particular $j_1 < j_2$, we define the following:

$$\begin{aligned} A &= \{(i, j_1) : j_2 h < i \leq (j_2 + 1)h\} \\ B &= \{(i, j_2) : j_1 h \leq i < (j_1 + 1)h\} \end{aligned}$$

Note that both sets contain h switches. The smallest group number in A is greater than $j_2 h \geq (j_1 + 1)h$ and thus $A \subset P$. Similarly, the largest group number in B is less than $(j_1 + 1)h \leq j_2 h$ and thus $B \subset R$.

As observed above, all links from switches in A go to the groups in B. Similar reasoning shows that all links from switches in B go to the groups in A. Since there is only one link between each pair of groups, the switches in A and B must be adjacent. Because we accounted for all links leaving each set, $A \cup B$ forms a GCC of size $2h$. ■

As with the relative global link arrangement, we can use this information about the size and structure of the GCCs to reason about the bisection bandwidth for large values of α. Again, a being a multiple of 4 is the simplest case. In this case, there are an even number of both large and small GCCs, guaranteeing the existence of bisections without crossing global links:

Corollary 6: When a is a multiple of four, the absolute arrangement's bisection bandwidth is $O(1)$.

Unlike in the relative case, however, this is not the only situation when the min-bandwidth cut is not crossed by global links because it is possible for several small GCCs to match with a collection of large ones. In these situations, the absolute arrangement's bisection bandwidth is bounded while the relative arrangement's continues to grow with α.

V. Mapping stencil jobs

As a second way to compare the global link arrangements, we consider their effect on task mapping. Since the entire goal of the Dragonfly topology is to place all nodes within a few hops of each other, at first it may seem that the topology renders careful task mapping unnecessary. This is not correct. A short version of the argument is that hotspots can still occur; they are the entire reason for implementing indirect routing.

A more nuanced argument considers the tradeoff inherent in optimizing network performance on a Dragonfly system. On one hand, nodes within a group are all connected, encouraging allocations that concentrate jobs into few groups. On the other hand, there is only one link between each pair of groups so global links will tend to be hotspots in concentrated allocations. Improved task mapping represents a way to escape this tradeoff by reducing traffic at potential hotspots through task placement, hence allowing more concentrated allocations.

This argument is borne out in recent work by Prisacari et al. [17]. They improved the performance of personalized all-to-all communication in which each node sends a separate message to every other node. Because the nodes are symmetric in this communication pattern, task mapping is not an issue, but they found that its time analog was crucial; the algorithm used phases, each specifying the communication partner for each node. In particular, they found that achieving high performance required two things:

- The first (and more important) was to distribute messages evenly between paths to prevent congestion on the optical links.
- A second-order consideration was to ensure that all paths in a phase have the same length (in particular, that all use an optical link or none do). This lets all parts of the phase end together and minimizes synchronization delays or inter-phase contention.

Although [17] solves a scheduling problem rather than a mapping one, these observations give us desired criteria for mappings: the mapping must allow communication in phases that balance traffic and have equal-length paths.

Based on these criteria, we now give a preliminary comparison of the global link arrangements in terms of their impact on the task mapping problem. In particular, we look at jobs that communicate in a 2D nearest neighbor stencil pattern. In this pattern, tasks that correspond to integer points in a two-dimensional grid and each communicates with it nearest neighbors, the closest points in each cardinal direction ($+x$, $-x$, $+y$, and $-y$). This is a very common communication pattern in computational science applications, arising naturally from spatial decompositions into regular rectangular regions. If large Dragonfly systems are built, some of their applications will use this important pattern.

For our evaluation, we looked at good task mappings for a specific case: a 6×6 2D stencil job on a $(p, 4, 2)$-Dragonfly (depicted in Figures 2–4) where each task is assigned p nodes (i.e. we are mapping switches rather than nodes). Figures 6(a), 6(b), and 6(c) show mappings for the relative, absolute, and circulant-based arrangements respectively. For example, in Figure 6(a), the top left task is mapped to switch 3 of group 0. In each mapping, 2×2 subgraphs of the task communication graph are mapped to each group of switches (encircled by dotted lines). The tasks can communicate with

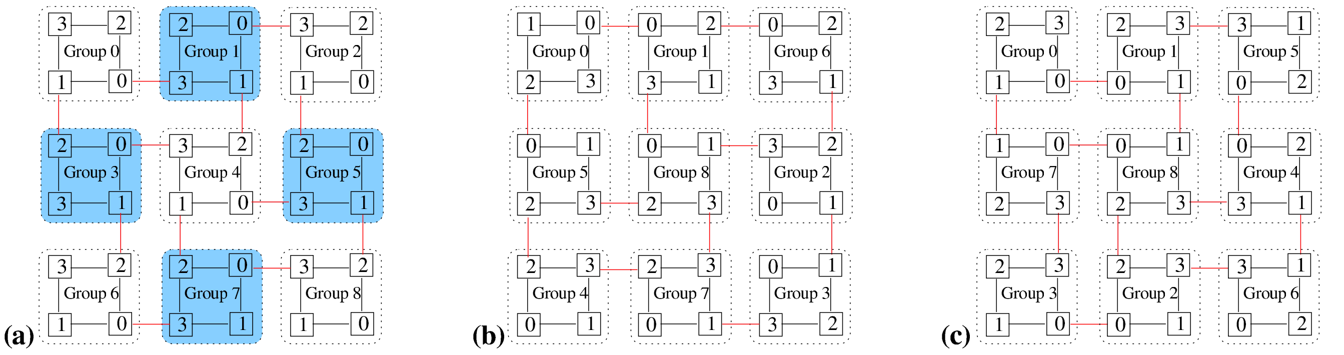

Figure 6. Three mappings for a 6 × 6 stencil job. Switches in a group are numbered and surrounded by a dotted line. Red edges between groups are direct global links. **(a)** shows a mapping with the relative arrangement. The checkerboard pattern reflects the repeated use of 2 group configurations. **(b)** shows a mapping for the absolute arrangement. There are 6 group configurations. **(c)** shows a mapping for the circulant-based arrangement. There are 6 group configurations.

all their neighbors in the following phases:

- First, communication within each subgraphs is performed using direct local links. (This phase could be split to separate communication by dimension.)
- Next, neighboring tasks connected by a global link communicate. In the mapping for the relative arrangement (Figure 6(a)), switches 0 and 3 in neighboring groups communicate in the x direction and switches 1 and 2 communicate in the y direction.
- Finally, neighboring tasks without a direct link use a multi-hop path, each taking a local link, a global link to the adjacent group, and then a local link.

In each phase, all paths are the same length, each link is used at most once, and each switch handles the same amount of traffic. Thus, these arrangements both avoid congestion and have equal-length phases.

Although we found mappings for each global link arrangement meeting our criteria, there is a qualitative difference between them that favors the relative global link arrangement. An appealing feature of that arrangement's mapping (Figure 6(a)) is its regularity. It uses two types of groups organized in a checkerboard pattern (shown via shading). Within a type, all the groups give the same relative position and role to each switch. For example, in the unshaded groups, switch 0 is always the bottom right switch and it always has a global link to the group in the $+x$ direction. This pattern generalizes to give mappings for at least some other mesh and Dragonfly sizes; switch 0 can always connect in the $+x$ direction since it goes to the numerically-next group and switch 1 can connect in the y dimension by going to the $(h+1)^{\text{st}}$ next group (recall that h is the number of global links per switch). Figure 7 shows the application of this technique to a 12 × 8 2D stencil job on 16 groups of a $(p, 6, 3)$-Dragonfly; note that the larger groups on this machine allow the mapping to use a single type of group.

None of the mappings we found for the other global link arrangements exhibit this regularity. The mappings shown in Figures 6(b) and 6(c) use groups with 6 different relative positions of the switches. For the absolute mapping, even some groups with the same relative positions use different outgoing global links to communicate with neighboring groups (e.g. groups 1 and 6). The lack of regularity would significantly complicate code implementing these mappings, not to mention finding the mappings themselves.

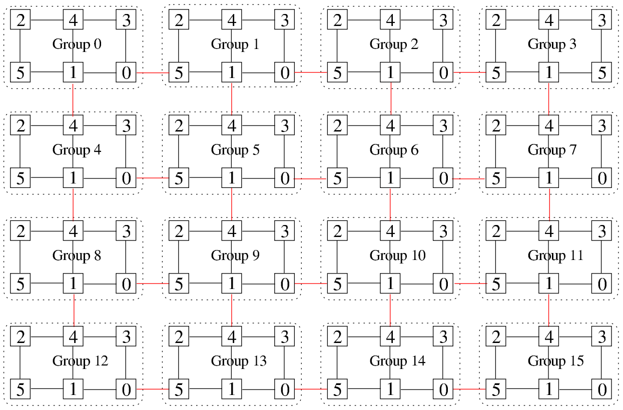

Figure 7. Mapping for a 12 × 8 stencil job on 16 groups of a $(p, 6, 3)$-Dragonfly with the relative arrangement

VI. Related work

As mentioned previously, we are not aware of any previous work comparing global link arrangements. Camarero et al. [11] defined the three arrangements we use here in the context of a paper focused on routing. To our knowledge, this is the only prior work to recognize that there are systematically different ways to make the global links.

Chakaravarthy et al. [18] give a sufficiently-detailed description of the PERCS system (described below) to determine that it uses the absolute arrangement.

In addition, Prisacari et al. [19] use what seems to be the absolute arrangement in an example and Garcia et al. [20] similarly use the relative arrangement. In both cases, the authors use the arrangement in a figure to illustrate the Dragonfly topology without describing the arrangement in general or acknowledging that different choices exist.

Arguably, the most closely related prior work involves task mapping for Dragonfly systems. Bhatele et al. [21] examined different ways to map a nearest neighbor stencil computation based on blocking (assigning pieces of the mesh to nodes or groups). They found that blocking helped relative to a default strategy, but also that using indirect routing gave comparable performance. Chakaravarthy et al. [18] expand on this with a coloring-based scheme to distribute submeshes between groups to distribute traffic on the global links; this may benefit our mapping approach as the groups get larger. Jokanovic et al. [22] and Prisacari et al. [19] give theoretical models that predict network bottlenecks for different communication patterns, particularly looking at the effect of random task mapping.

The Dragonfly topology has also been the subject of other research efforts. Garcia et al. [23] consider variations of the routing algorithm. Prisacari et al. [17] consider scheduling personalized all-to-all messages to minimize contention. Jain et al. [24] use large simulations to consider the interaction of routing strategies, node allocation strategies, and communication patterns.

Systems built based on the Dragonfly interconnect topology have thus far implemented variations of it. One of these is the Cray XC [7], [8], which forms each group from 6 chassis organized in 2 cabinets. Each chassis contains 16 switches, for a total of $6 \times 16 = 96$ switches per group. Switches in the group are connected to all other switches in their chassis and their peer in the other chassis. Thus, instead of the group forming a clique, it is actually six cliques with corresponding switches connected ($K_{16} \times K_6$). The switches are Aries ASICs, each connected to 4 dual-socket nodes. Each Aries can support up to 10 optical links, but global links are formed from 4 of these. Thus, each group can have $96 \times 10/4 = 240$ outgoing global links (implying 241 groups), but some links are split between a pair of switches.

Cordery et al. [25] evaluate the performance of the Cray XC on a variety of scientific applications. The system's bisection bandwidth was previously computed by Alverson et al. [7], but their calculation treats each group as a vertex, effectively assuming infinite bandwidth on the local links.

The other system built using a variation of the Dragonfly topology is PERCS [9], which is now sold as the IBM Power Systems 775 [26]. (The bandwidth parameters given here are from a later reference [21], which gives slightly different values.) PERCS uses three kinds of links rather than just 2. The groups (called *supernodes*) are connected with a 10 GB/s optical link. The supernodes themselves are formed from 4 *drawers* with 8 switches each. The switches within a drawer are connected to each other with 24 GB/s electrical links and also to other switches in their supernode with 5 GB/s optical links. Thus, the local links are non-uniform, with some being the system's fastest links and some being its slowest. The switches themselves are each connected to four processors.

VII. Discussion

We have demonstrated that the specification of a global link arrangement is important for Dragonfly interconnects. The structure of GCCs we uncovered can aid in the selection of parameters for larger systems. In particular, it identifies situations where the bisection bandwidth is determined by local links rather than global ones.

The obvious open question is to determine the bisection bandwidth for general Dragonflies at lower values of α, where some global edges still cross min-bandwidth cuts. We are also interested in whether the absolute arrangement ever gives higher bisection bandwidth than the others or if the relative arrangement ever gives higher bisection bandwidth than the circulant-based one; we never saw such a situation, but cannot rule it out.

Another avenue for future work is designing other global link arrangements. In addition to trying the arrangements described here, we also considered *random global link arrangement* for the $(p, 4, 2)$-Dragonfly. To generate one of these, we had each group generate a random permutation of the other groups and used these as the destination group of each its ports. None of the arrangements we generated was superior to the circulant-based arrangement except at $\alpha > 4$, for which some of them continued to improve.

VIII. Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. We also thank Logan Ayers and Hai Le for finding an error in an earlier version of Figure 7. D. Rincon-Cruz, M. Spehlmann, S. Meyers, and D.P. Bunde were partially supported for this work by the National Science Foundation under grant CNS-1423413. E. Hastings, A. Xu, and D.P. Bunde were partially supported for this work by Sandia National Laboratories under contract 899808. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] P. Kogge, K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, K. Hill, J. Hiller, S. Karp, S. Keckler, D. Klein, R. Lucas, M. Richards, A. Scarpelli, S. Scott, A. Snavely, T. Sterling, R. Williams, and K. Yelick, "Exascale computing study: Technology challenges in achieving exascale systems," Tech. Rep., 2008.

[2] J. Kim, W. Dally, B. Towles, and A. Gupta, "Microarchitecture of a high-radix router," in *Proc. 32nd Ann. Intern. Symp. Comput. Arch. (ISCA)*, 2005, pp. 420–431.

[3] S. Scott, D. Abts, J. Kim, and W. Dally, "The BlackWidow high-radix Clos network," in *Proc. 33rd Ann. Intern. Symp. Comput. Arch. (ISCA)*, 2006, pp. 16–28.

[4] D. Abts, A. Bataineh, S. Scott, G. Faanes, J. Schwarzmeier, E. Lundberg, T. Johnson, M. Bye, and G. Schwoerer, "The Cray BlackWidow: A highly scalable vector multiprocessor," in *Proc. ACM/IEEE Conf. High Performance Networking and Computing (SC)*, 2007.

[5] J. Kim, W. Dally, and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," in *Proc. 34th Ann. Intern. Symp. Comput. Arch. (ISCA)*, 2007, pp. 126–137.

[6] J. Kim, W. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *Proc. 35th Ann. Intern. Symp. Comput. Arch. (ISCA)*, 2008, pp. 77–78.

[7] B. Alverson, E. Froese, L. Kaplan, and D. Roweth, "Cray XC series network," Cray, Inc., White paper, 2012.

[8] G. Faanes, A. Bataineh, D. Roweth, T. Court, E. Froese, B. Alverson, T. Johnson, J. Kopnick, M. Higgins, and J. Reinhard, "Cray cascade: a scalable HPC system based on a Dragonfly network," in *Proc. Conf. High Performance Computing, Networking, Storage and Analysis (SC)*, 2012, p. 103.

[9] B. Arimilli, R. Arimilli, V. Chung, S. Clark, W. Denzel, B. Drerup, T. Hoefler, J. Joyner, J. Lewis, J. Li, N. Ni, and R. Rajamony, "The PERCS high-performance interconnect," in *Proc. 18th Symp. High-Performance Interconnects (Hot Interconnects)*, 2010.

[10] L. Valiant, "A scheme for fast parallel communication," *SIAM J. Computing*, vol. 11, no. 2, pp. 350–361, 1982.

[11] C. Camarero, E. Vallejo, and R. Beivide, "Topological characterization of hamming and dragonfly networks and its implications on routing," *ACM Trans. Architec. Code Optim.*, vol. 11, no. 4, p. 39, 2014.

[12] D. Patterson and J. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 4th ed. Morgan Kaufmann, 2012.

[13] M. Deveci, S. Rajamanickam, V. Leung, K. Pedretti, S. Olivier, D. Bunde, Ü. Çatalyürek, and K. Devine, "Exploiting geometric partitioning in task mapping for parallel computers," in *Proc. 28th IEEE Intern. Parallel and Distributed Processing Symp. (IPDPS)*, 2014.

[14] A. Bhatele and L. Kale, "Benefits of topology-aware mapping for mesh topologies," *Parallel Processing Letters*, vol. 18, no. 4, pp. 549–566, 2008.

[15] F. Gygi, E. W. Draeger, M. Schulz, B. de Supinski, J. Gunnels, V. Austel, J. Sexton, F. Franchetti, S. Kral, C. Ueberhuber, and J. Lorenz, "Large-scale electronic structure calculations of high-Z metals on the BlueGene/L platform," in *Proc. ACM/IEEE Conf. High Performance Networking and Computing (SC)*, 2006.

[16] U. Feige and R. Krauthgamer, "A polylogarithmic approximation of the minimum bisection," *SIAM J. Comput.*, vol. 31, no. 4, pp. 1090–1118, 2002.

[17] B. Prisacari, G. Rodriguez, and C. Minkenberg, "Generalized hierarchical all-to-all exchange patterns," in *Proc. 27th IEEE Intern. Parallel and Distributed Processing Symp. (IPDPS)*, 2013.

[18] V. Chakaravarthy, M. Kedia, Y. Sabharwal, N. Katta, R. Rajamony, and A. Ramanan, "Mapping strategies for the PERCS architecture," in *Proc. 19th Intern. Conf. High Performance Computing (HiPC)*, 2012.

[19] B. Prisacari, G. Rodriguez, P. Heidelberger, D. Chen, C. Minkenberg, and T. Hoefler, "Efficient task placement and routing in dragonfly networks," in *Proc. 23rd ACM Intern. Symp. High-Performance Parallel and Distributed Computing (HPDC)*, 2014.

[20] M. Garcia, E. Vallejo, R. Beivide, M. Odriozola, C. Camarero, M. Valero, G. Rodriguez, J. Labarta, and C. Minkenberg, "On-the-fly adaptive routing in high-radix hierarchical networks," in *Proc. 41st Intern. Conf. Parallel Processing (ICPP)*, 2012, pp. 279–288.

[21] A. Bhatele, N. Jain, W. Gropp, and L. Kale, "Avoiding hot-spots on two-level direct networks," in *Proc. Conf. High Performance Computing, Networking, Storage and Analysis (SC)*, 2011.

[22] A. Jokanovic, B. Prisacari, G. Rodriguez, and C. Minkenberg, "Randomizing task placement does not randomize traffic (enough)," in *Proc. 2013 Interconnection Network Architecture: On-Chip, Multi-Chip (IMA-OCMC)*, 2013, pp. 9–12.

[23] M. García, E. Vallejo, R. Beivide, M. Odriozola, and M. Valero, "Efficient routing mechanisms for dragonfly networks," in *Proc. 42nd Intern. Conf. Parallel Processing (ICPP)*, 2013, pp. 582–592.

[24] N. Jain, A. Bhatele, X. Ni, N. Wright, and L. Kale, "Maximizing throughput on a dragonfly network," in *Proc. Conf. High Performance Computing, Networking, Storage and Analysis (SC)*, 2014.

[25] M. Cordery, N. Wright, B. Austin, C. Daley, H. Wasserman, S. Hammond, and D.Doerfler, "Analysis of Cray XC30 performance using Trinity-NERSC-8 benchmarks and comparison with Cray XE6 and IBM BG/Q," in *Proc. 4th Intern. Workshop Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, 2013.

[26] D. Quintero, K. Bosworth, P. Chaudhary, R. de Silva, B. Ha, J. Higino, M.-E. Cahle, T. Kamenoue, J. Pearson, M. Perez, F. Pizzano, R. Simon, and K. Sun, "IBM Power Systems 775 for AIX and Linux HPC solution," IBM, Redbook, 2012, http://www.redbooks.ibm.com/redbooks/pdfs/sg248003.pdf.