Analyzing Network Health and Congestion in Dragonfly-based Supercomputers

Abhinav Bhatele[†], Nikhil Jain[*], Yarden Livnat[‡], Valerio Pascucci[‡], Peer-Timo Bremer[†,‡]

[†]*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, California 94551 USA*
[*]*Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801 USA*
[‡]*Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, Utah 84112 USA*
E-mail: [†]{*bhatele, ptbremer*}*@llnl.gov,* [*]*nikhil@illinois.edu,* [‡]{*yarden, pascucci*}*@sci.utah.edu*

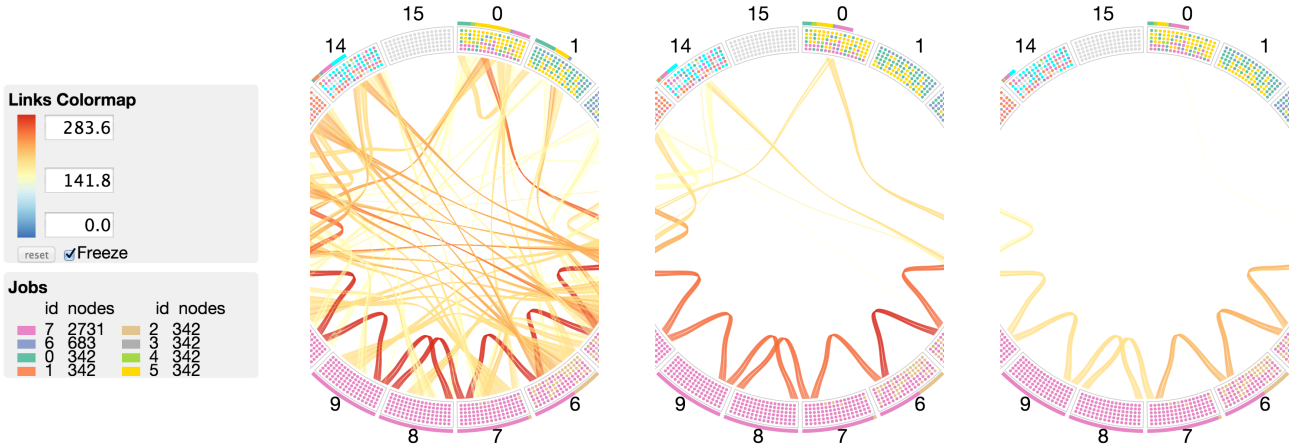

Figure 1. Congested network links in a dragonfly-based supercomputer simulating an eight-job parallel workload. Left to right: Different network configurations using two, three, and five inter-group links per router pair respectively. The additional links minimize hot-spots and reduce the overall average network load.

***Abstract*—The dragonfly topology is a popular choice for building high-radix, low-diameter, hierarchical networks with high-bandwidth links. On Cray installations of the dragonfly network, job placement policies and routing inefficiencies can lead to significant network congestion for a single job and multi-job workloads. In this paper, we explore the effects of job placement, parallel workloads and network configurations on network health to develop a better understanding of inter-job interference. We have developed a functional network simulator, Damselfly, to model the network behavior of Cray Cascade, and a visual analytics tool, DragonView, to analyze the simulation output. We simulate several parallel workloads based on five representative communication patterns on up to 131,072 cores. Our simulations and visualizations provide unique insight into the buildup of network congestion and present a trade-off between deployment dollar costs and performance of the network.**

***Keywords*-dragonfly network; congestion; inter-job interference; simulation; visual analytics**

I. Introduction

The increase in on-node flop/s capacity and memory bandwidth at a higher rate than the inter-node link bandwidth is making many large-scale parallel applications communication-bound, meaning their performance is now limited by the available network resources. Further, on most supercomputers the network is shared among all running jobs, which can result in significant performance degradation as multiple communication-heavy jobs compete for the same bandwidth. Taking the specifications of the upcoming DOE CORAL [1], [2], [3], Trinity [4] and NERSC-8 [5] procurements as an indication of future trends, this problem will continue to grow as the flop/s per node will jump significantly with only a nominal increase in network capacity.

One strategy to address this challenge is to move toward higher-radix, lower-diameter network topologies such as the dragonfly. Several of the upcoming DOE procurements (Aurora [3] at Argonne, Cori [5] at NERSC and Trinity [4] at Los Alamos/Sandia) will deploy some variation of the dragonfly topology [6], [7] with the expectation that such highly connected network topologies coupled with adaptive routing will greatly reduce or eliminate the effects of job interference and network congestion [8], [9], [10], [11], [12]. However, preliminary experiments on Edison, a Cray XC30 at NERSC, have shown that for communication-heavy applications, inter-job interference and thus network congestion remains an important factor. Consequently, understanding the causes of congestion and identifying potential ways to mitigate it are crucial for maintaining a high throughput and predictable performance on current and future supercomputers.

Understanding inter-job interference and its root causes is a challenging task. Routing algorithms, job placement policies and network wiring configurations are some of the key factors

that may cause job interference and contribute to performance degradation. In this work, we focus on job placements and network configurations. The former can provide insights that lead to a comparatively low cost avenue to reduce network congestion. The latter enables the understanding of factors that affect small-scale dragonfly installations such as Edison and future dragonfly-based supercomputers.

Our work focuses on Edison – a 30-cabinet Cray Cascade (XC30) system [7] installed at NERSC. Edison consists of only 15 groups compared to 241 groups in the full-scale dragonfly design, and uses only ∼1,400 routers compared to ∼23,000 routers in the full-scale design (see Figure 2). As a result, only one-third of the inter-group or global ports are populated with optical links, which results in lower bisection bandwidth compared to the full system. Since the inter-group optical links are expensive, understanding how to maximize network performance on a fixed budget is of significant interest.

In this paper, we explore the effects of job placement, parallel workloads and network configurations on network health and congestion to develop a better understanding of inter-job interference on Edison. Since Edison is a production machine, it is difficult to collect system-wide network data on it or perform controlled experiments. Hence, we have extended a functional network simulator to model the network behavior of the Cray Cascade architecture. Damselfly provides system-wide and per-job network hardware counters for multi-job simulations, and allows us to easily explore different job placements, routing policies, and even network hardware configurations by adding and/or removing network links.

We have also developed new visualizations of the dragonfly topology to provide a more intuitive understanding of and help in analyzing the resulting simulation data. For our case studies, we have simulated 44 parallel workloads based on five representative communication patterns on up to 131,072 cores. We present a detailed analysis of 588 simulations showing insights into the causes of network congestion provided by the resulting data and corresponding visualizations. Our contributions are:

- Damselfly, a model-based network simulator extended to handle multi-job parallel workloads and per-job network performance counters;
- An ensemble of 44 workloads and 588 simulations exploring different job combinations, placements and network configurations;
- An in-depth analysis of network congestion using both traditional statistics as well as novel, intuitive visualizations of the dragonfly topology; and
- Trade-offs between network performance and dollar costs when adding or removing links from the network.

II. Background and Related Work

The overall performance or system throughput of a supercomputer depends on a variety of factors: one-time decisions made by the procurement division or the hardware team during system installation; day-to-day execution policy decisions made by the system administrators; and the mix of jobs that run on the system. In this study, we focus on the factors that specifically affect network health, which directly impacts the overall performance of the system.

Network topology and link bandwidths: The communication performance of parallel jobs running on a supercomputer depends heavily on the interconnection network deployed on the system, its topology, and the latency and bandwidth of network links. These decisions are made once during procurement. Using additional links and/or increasing their bandwidth can improve the communication performance at a high monetary cost with possibly diminishing returns.

Routing policy: Another important factor that determines application and system performance is the routing policy used for sending messages over the network. Shortest-path static routing policies can lead to congestion and hot spots on the network where a few links become the bottleneck. Adaptive dynamic routing can route messages around hot-spots and avoid delays by using alternate and/or longer routes.

Job placement policy: This policy determines the resource allocation of available nodes to new jobs. The Blue Gene family of supercomputers employs a network isolation policy and allocates only contiguous partitions that do not share network links with one another. This policy leads to predictable performance and faster execution at the cost of longer job queue wait times and lower system utilization due to fragmentation. In contrast, supercomputers such as the Cray XT/XE and XC families typically employ a topology-oblivious resource allocation policy that allows multiple jobs to share network links. Topology-oblivious policies often lead to higher system utilization at the cost of slower execution of individual jobs due to network interference and thus lower overall system throughput (number of jobs retired).

Parallel workload or job mix: The effects of routing and job placement policies on individual job and overall system performance depend on the system's parallel workload signature, i.e., the typical collection of jobs that run on the machine. If most of the jobs require a small number of nodes and are not communication-intensive, then inter-job interference and network congestion are less of a concern. However, even a few communication-heavy jobs with a somewhat significant allocation size can impact the performance of other jobs running alongside. The choice of routing and job placement policies can either amplify or mitigate such effects.

A. Related Work

Several researchers have investigated the impact of job placement on performance [9], [10], [13], [14]. Skinner et al. [11] and Wright et al. [12] noted significant performance variability due to network contention. Recently, Bhatele et al. [8] studied the impact of inter-job interference on Cray torus networks. Several modeling tools and simulators have been developed to study high-performance networks. Hoefler et al. [15] developed analytical models for network traffic on different network topologies. Bhatele et al. [14] used BigSim to model the PERCS network and study different placement

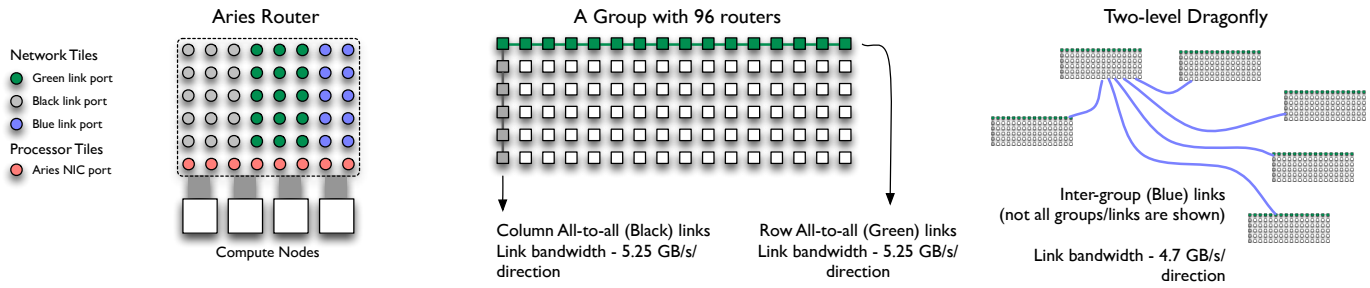

Figure 2. Various components and logical units in the Cray Cascade (dragonfly) design: An Aries router has 40 network and 8 processor tiles (left). 96 such routers form a group (center) and several groups connect together to form the two-level network (right).

and routing policies. SST [16] also supports various network topologies including the dragonfly network.

Previously, we have shown that a randomized placement or indirect routing can lead to good performance on the IBM PERCS machine [14]. We have also studied various routing choices for different communication patterns and parallel workloads on the dragonfly topology [17]. We found that the Universal Globally Adaptive Load-balanced (UGAL) routing [18] is the best for minimizing congestion on the dragonfly network. Several aspects that differentiate our work from previous research are: 1. We use a parallel simulator that implements an analytical modeling approach to network congestion, which enables us to perform large-scale simulations very quickly as compared to detailed discrete-event simulations; 2. We use Damselfly to simulate parallel workloads and record network traffic at the granularity of individual jobs. This functionality is not available in other network simulators, to the best of our knowledge; and 3. Our work also analyzes the effects of changing the underlying network topology and its impact on network health.

III. Modeling using Damselfly

In order to study the effects of job placement, parallel workloads, and network configurations on network health and congestion, we need information about the attributes of each message. These include the message size, its source and destination router, its routing path and the job to which it belongs. Since one cannot currently collect such system-wide network data on Edison and performing controlled experiments in production is difficult, we have extended a functional network simulator to model the network behavior of the Cray Cascade architecture [17]. We first describe the baseline version of Damselfly and then the enhancements done to the simulator to enable the studies in this paper.

Given a router connectivity graph and an application communication pattern, the network model in Damselfly performs an iterative solve to redistribute traffic from congested to less loaded links. The exact algorithm used for traffic or load redistribution is determined by the routing policy. In addition to other routing policies, Damselfly implements the UGAL routing [18], a variation of which is deployed on Edison.

UGAL routing and its variants try to balance the traffic on global channels by using dynamic load information available at the source. Various schemes are used to make up-to-date information available at the source, such as queue occupancy, selective virtual-channel discrimination and credit round-trip latency to sense back-pressure (more details in [6]). In Damselfly, four routes are chosen for each packet in a message – up to two direct paths and the remainder indirect. We assume global knowledge of congestion, which corresponds to UGAL-G routing, considered an ideal implementation of the UGAL routing. All experiments in this paper use the UGAL-G routing.

The research studies proposed in this work required several new features in Damselfly. These features and their implementations are described below.

Simulation of arbitrary interconnect graph: The previous version of Damselfly had pre-defined connectivity between the routers in the network as described in [6]. The inter-group connections could be distributed only in a round-robin fashion among the routers in a group, and only one intra-group connection per router pair was permitted. To provide more flexibility, the user can now specify an arbitrary interconnection graph. This allows us to connect the inter-group ports in arbitrary ways and we can also use more than one link to connect a router pair. This can be used to increase bandwidth between different router pairs as done on Edison, for example.

Most of the experiments in the paper use the network connectivity provided to us by NERSC system administrators for the Edison installation. The modification to Damselfly allows us to read in the connectivity file as input. Edison has 15 groups, each with 96 Aries routers arranged in a two-dimensional grid of 6×16 (Figure 2). Each row and column of this grid is connected in an all-to-all fashion by so-called *green* (rows) and *black* (columns) links. Each router is also connected to routers in other groups via *blue* (inter-group) links. To increase the overall bandwidth, the facility decided to utilize some of the spare ports to put three black links per router pair in each column and two blue links per router pair for inter-group connections. The default configuration we use in our simulator mimics this setup exactly except for a small adjustment in the per-link bandwidth. To simplify the subsequent analysis, we assume a bandwidth of 5.0 GB/s for links of all colors unlike the 5.25 GB/s for black and green and 4.75 GB/s for the blue links on Edison.

Link traffic of individual jobs: The second feature added

Figure 3. Link traffic statistics of green, black and blue links for individual jobs with different communication patterns and core counts (16k, 32k, 64k).

to Damselfly is the ability to record the traffic generated by each job executing in a parallel workload. For each link on the network, packets or bytes can now be attributed to specific jobs. Attributing the traffic on each link to individual jobs can be useful for studying inter-job interference and identifying jobs that negatively impact other jobs and as a result, overall system utilization.

Communication graph and placement of individual jobs: Finally, Damselfly now includes an easier interface for providing the communication graph and node placement of individual jobs executing in a workload. Previously, the graph and placement information for all jobs was provided in one file each. Now, users can provide MPI rank-based communication graphs and placements of individual jobs in per-job input files, and Damselfly combines all of this information to perform a full-machine simulation.

For the experiments in this paper, we implemented a job placement policy that resembles placement scenarios common on Edison in a steady state (once the machine has been up and running for several days). Most of the nodes of a job are located as close together as possible within the same group or adjacent groups (by logical group ID). However, large jobs usually contain a number of outliers scattered on the system depending on which nodes are available.

Since Damselfly models steady state network traffic for aggregated communication graphs as opposed to communication traces with time stamps, it does not model and predict execution time. In previous papers [14], [19], [20], we have shown that average and maximum traffic on network links are reasonably good predictors of execution time and performance. Hence, in this paper, we use these network metrics to compare the predicted performance between different simulations.

IV. Simulation Setup

Our aim is to model parallel workloads that make up the job mix on Edison and other DOE systems. We use five communication patterns that are representative of application codes run at NERSC and other DOE facilities:

2D Stencil (2D): Each MPI process communicates with four neighbors in a two-dimensional Cartesian grid (64 KB messages), which is representative of a two-dimensional Jacobi relaxation problem.

4D Stencil (4D): Each MPI process communicates with eight neighbors in a four-dimensional Cartesian grid (4 MB messages). This is representative of the communication in MILC [21], a Lattice QCD application frequently used on NERSC machines.

Many-to-many (M2M): Groups of 16, 32 or 64 MPI processes (depending on job size) participate in all-to-alls over sub-communicators of `MPI_COMM_WORLD` (32 KB messages per process). This is representative of parallel FFTs.

Spread: Each process sends 512 KB messages to randomly selected neighbors. The number of neighbors is fixed for each process and varies from 6 to 27. This pattern is used to add random background noise on the machine.

Unstructured Mesh (UMesh): Each process sends 512 KB messages to a carefully selected set of 6 to 27 neighbors, which is representative of an unstructured mesh computation.

We combine these five patterns to create multi-job workloads with different number of jobs and job sizes: 28 workloads with four jobs and 16 workloads with eight jobs.

In order to understand how the network behavior of one job may affect other jobs, we first analyze the behavior of individual jobs running on a subset of the system. We simulate each communication pattern or job on 16k, 32k and 64k cores, which is close to 12.5%, 25% and 50% of Edison. Figure 3 shows the average and maximum traffic predicted by Damselfly for the intra-group (green, black) and inter-group (blue) links in the system. As expected, the average link traffic increases as job sizes become larger. The 4D Stencil and Spread communication patterns send significantly higher traffic on the network compared to others (note the different y-axis ranges in individual plots).

Comparing between different link types, black links exhibit lower maximum traffic than blue or green links and are often not a bottleneck. In general, the maximum traffic through green or blue links is an indication of network hot-spots. As we can see, green links have higher maximum traffic than blue links for 2D Stencil, Many-to-many and UMesh. Blue links have the highest maximum traffic in 4D Stencil and Spread,

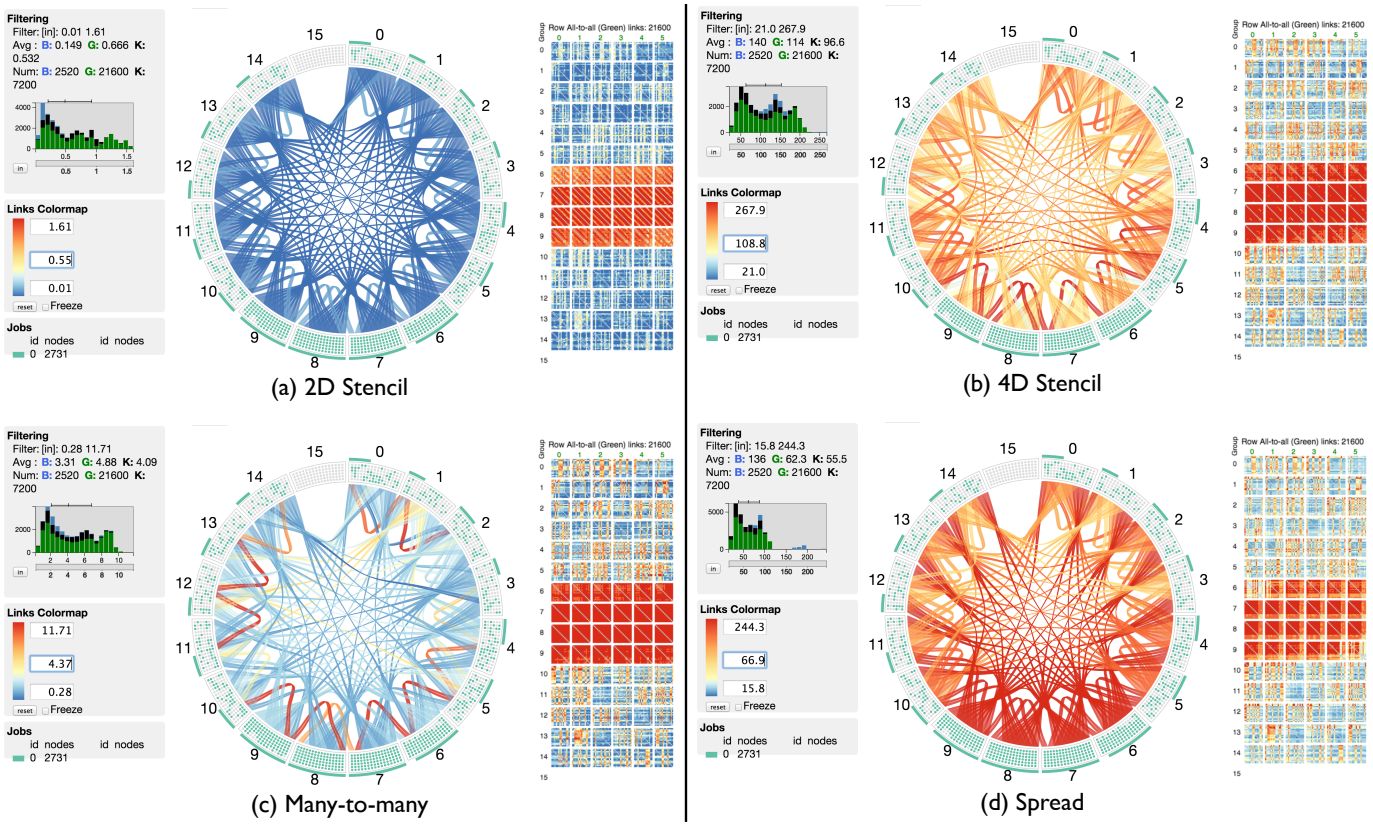

Figure 4. Traffic on blue (radial view) and green (matrix view) links for individual jobs running alone on 64k cores.

which are the communication-heavy patterns. Both green and blue links are congested for 4D Stencil but blue links are the obvious bottleneck for Spread. A more detailed analysis of the individual traffic patterns confirms these conclusions.

Figure 4 shows the network traffic of the first four patterns simulated on their own on 64k cores of Edison. In each quadrant, the left column shows a histogram of link traffic stacked by link color and the color map used. The central radial view depicts the potential 16 groups (only 15 are actually used), each consisting on 96 routers arranged in a 6x16 grid. Each router is colored by the id of the job running on its nodes (grey if the corresponding nodes are unused and cyan if different jobs are running on the four nodes of a router). The inter-group (blue) links are shown as curved connectors between the inner rims of the groups and colored based on the network traffic passing through them. The matrices on the right within each quadrant show the network traffic on the green links. Each row depicts six 16×16 communication matrices for each group representing six cliques formed by the green all-to-all links. Black links are not shown because they have significantly less traffic, as shown in Figure 3.

The color maps in the four quadrants in Figure 4 are not normalized to a single range in order to emphasize the particular network traffic behavior of each pattern. Green links are the primary bottleneck for 2D Stencil (Figure 4(a)) and UMesh (not shown). 4D Stencil generates a significant amount of traffic on both blue and green links with blue links exhibiting more congestion (Figure 4(b)). Many-to-many, on the other hand, generates more traffic on green links but some congestion can be seen over the blue links as well (Figure 4(c)). In the case of Spread, blue links are the clear bottleneck with less pressure on green links (Figure 4(d)).

V. Analyzing Network Health and Congestion

In this section, we use detailed simulations to study various factors that impact network congestion and overall network health. In particular, we focus on job placements, job interference in parallel workloads and network wiring configurations.

A. Job Placements

The policy used for assigning available nodes to eligible jobs in the queue can impact the performance of individual jobs and the overall system throughput. Resource managers typically allocate nodes without regard to their relative location in the network topology, which can significantly hurt performance depending on the network. On torus networks, having a compact allocation minimizes interference with other jobs. On dragonfly systems, distributing nodes of one job over multiple groups can help randomize traffic and better exploit a larger number of inter-group links [14], [17]. However, interference from other jobs precludes this from being the best policy universally.

Figure 5 compares the network traffic generated by simulating two different placements of the same 4D Stencil job running alone on 32k cores of Edison. The job in the left figure,

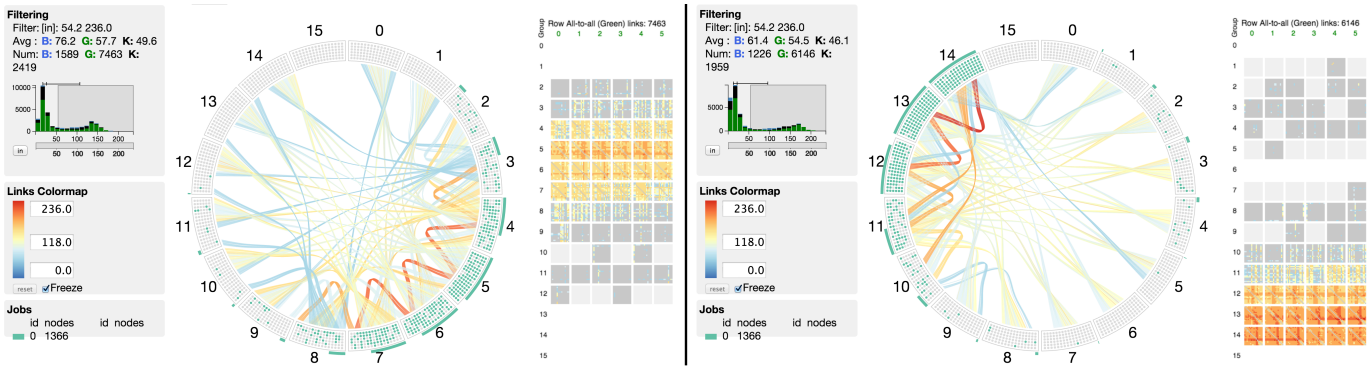

Figure 5. Job placement affects the traffic on blue and green links for a simulation of 4D Stencil running on 32k cores. On the left, the job is scattered which leads to lower maximum traffic (191 MB) but higher average traffic per link (55.5 MB) in comparison to the job on the right which is more compact (maximum traffic: 236 MB, average traffic: 51.4 MB). Note that only links with traffic above a certain threshold are shown.

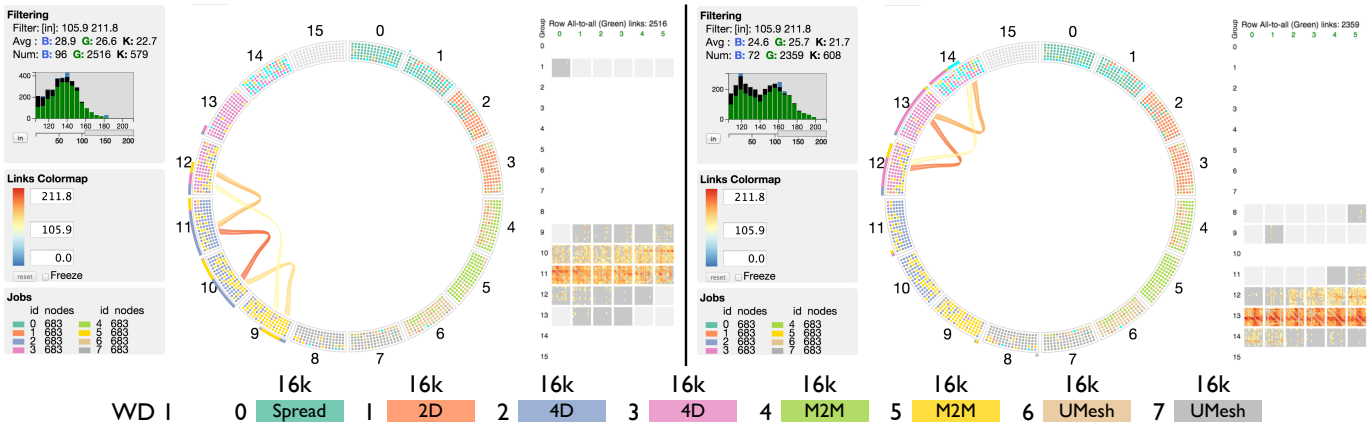

Figure 6. Both figures show the traffic attributed to the same communication pattern: 4D Stencil on 16k cores (Job 2 on the left, Job 3 on the right) in an 8-job workload (WD 1). Job 2 is scattered on eight groups and shares its groups primarily with Job 5 (Many-to-many), which is not communication-heavy. Job 3 (right) is more compact and occupies an entire group (13) and shares other groups with Jobs 2, 6 and 7. The maximum traffic is higher for Job 3 by 8% (on green links), while both jobs have similar average traffic. Note that only links with traffic above a certain threshold are shown.

which features a more scattered node placement, uses more blue links (1589 versus 1226, see numbers under the Filtering section on the top left of each figure) and has lower maximum traffic across all the links. However, the average traffic on blue links is lower for the job on the right – the compact placement balances intra-group (green) and inter-group (blue) link traffic.

To analyze the effect of job placement when other jobs are running alongside a communication pattern, we consider scenarios in which two jobs in a workload have the same communication pattern and use the same number of nodes. The only difference is that the two jobs are placed differently relative to other jobs. One scenario, depicted in Figure 6, is based on workload WD 1 and features two 4D Stencil jobs (Job 2 and 3). Network traffic due to Job 2 and Job 3 is shown in the left and right figure, respectively. Job 2 nodes are spread among eight groups, though most of the nodes are in groups 10 and 11. Job 2 primarily shares network links with Job 5, which is not communication-heavy. On the other hand, the placement of Job 3 is more compact, occupying all of group 13 and parts of groups 12 and 14. It shares links with Job 2 which is communication-heavy. This forces Job 3 to use the direct green and black links connecting its nodes, and thus results in it exhibiting an 8% higher maximum traffic compared to Job 2. In contrast, Job 2 is able to use indirect paths through sparingly used links to route its traffic.

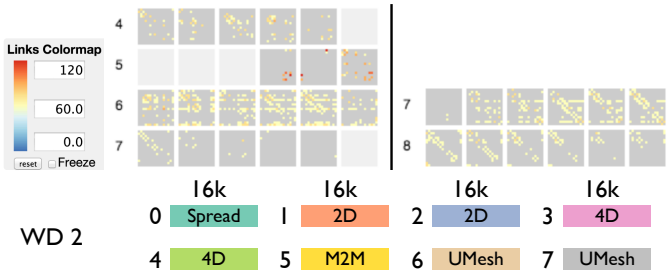

Figure 7. Traffic on green links attributed to Job 6 (left) and Job 7 (right) in WD 2. Both jobs represent UMesh running on 16k cores. Job 6 shares group 5 with a communication-heavy Job 4 (4D Stencil), which in turn leads to higher congestion (maximum traffic: 119.5 MB for Job 6 versus 72.4 MB for Job 7).

A similar pattern emerges for green links in the second scenario for WD 2 with two UMesh jobs (6 and 7, see Figure 7). The main difference is that Job 6 shares its groups with Job

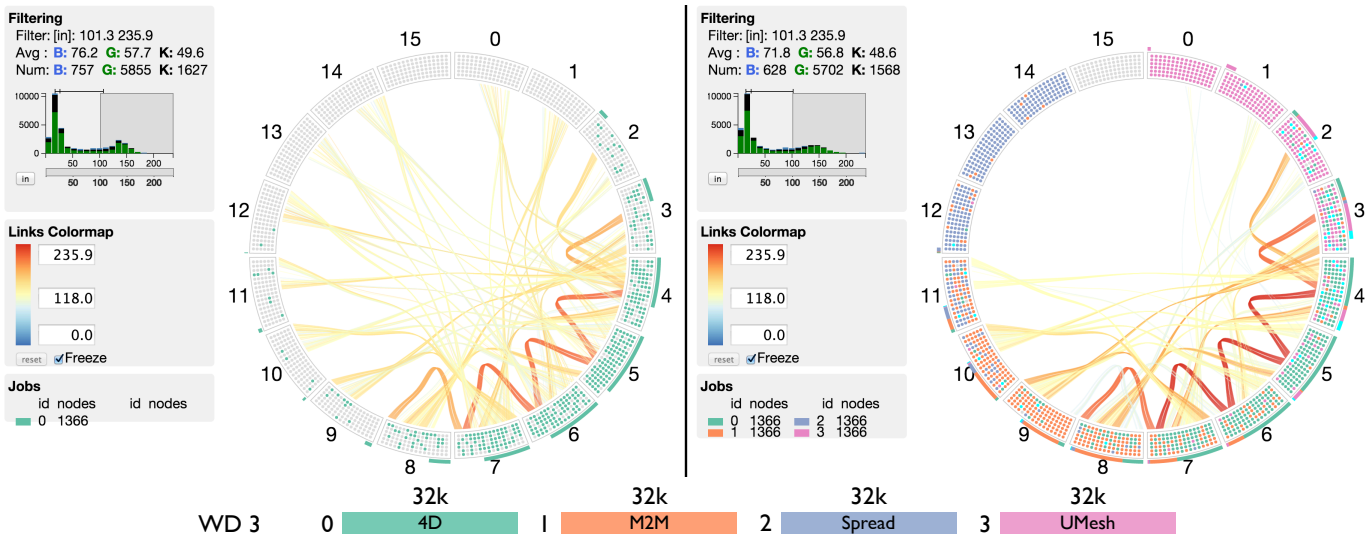

Figure 8. When Job 0 (4D Stencil) is run in a workload (WD 3) along side other jobs (right), the number of links with traffic above a certain threshold decreases and overall maximum traffic on blue links increases (231 MB) as opposed to when it is run individually (191 MB, left). In the parallel workload run, Job 0's traffic is confined to fewer blue links in order to share bandwidth with other jobs.

4 (4D Stencil), which leads to hot-spots on the green links in group 5. The difference in maximum traffic between Jobs 6 and 7 is significant (119.5 MB versus 72.4 MB).

These studies suggest that in many cases, the job placement policy has a significant impact on routing of messages on the network. The important lessons we learn from these experiments are: 1) While compact placement can help reduce average traffic by a small margin, it constrains the routing to fewer groups. Hence, it may lead to network congestion and hot-spots. 2) When co-located with other jobs, it is advisable to scatter jobs with heavy-communication away from one another.

B. Inter-job Interference

An interesting observation that is apparent only in the DragonView visualization is that adaptive routing *appears* to redistribute the traffic of each job to provide a fair share of bandwidth to other jobs. In order to compare the network traffic generated by a job running with and without other jobs in a parallel workload, we simulate each workload in two settings. In the first case, we run all jobs in the workload together. In the other setting, we run each job in the workload by itself but using the same placement as in the workload. Since the simulator outputs the link traffic for each job separately, we can compare the average and maximum traffic that a job generates in the parallel workload versus its individual execution.

Figure 8 shows the blue link traffic for 4D Stencil (Job 0) when run individually and in WD 3 with other jobs (we only show links with traffic above a certain threshold). When Job 0 runs individually on the machine (left figure), it uses more blue links between several groups to route its messages and thus increases its effective bandwidth. However, in a parallel workload setting, the inter-group and intra-group links are shared between multiple jobs. This results in the traffic of individual jobs being confined to fewer blue links, especially to links that directly connect routers allocated to the job. The reduction in available blue links per job increases the congestion on these links, as can be seen in Figure 8 (right). However, this allows other jobs to use the other less utilized blue links.

This phenomenon of traffic redistribution and increase in maximum traffic attributed to an individual job in a workload is highly dependent on the job mix. In Figure 9, the same job (Spread) observes higher or lower congestion depending on whether there are other communication-heavy jobs running along side it. WD 4 has 4D Stencil running on 64k cores which is communication-heavy (middle figure). In contrast, WD 5 has Many-to-many running on 64k cores which does not send nearly as much traffic (right figure). As a result, the maximum traffic on blue links originating from Spread (Job 0) is much higher in the middle figure (WD 4) as compared to the figure on the right (WD 5).

Figure 10 shows a similar effect in the green link traffic of UMesh running in two different workloads – WD 6 and WD 5. The largest job in WD 6 and WD 5 is 4D Stencil (64k cores) and Many-to-many (64k cores) respectively. Since the largest job in WD 6 is communication-heavy, it has a higher impact on Job 1 (UMesh). This leads to a higher maximum traffic on the green links (88 MB versus 58 MB).

From these results and other simulation experiments not presented here, we conclude that communication-heavy jobs in a workload can impact the traffic distribution of other jobs significantly. Due to the adaptive nature of routing on Edison, presence of communication-heavy jobs forces other jobs to restrict their traffic to fewer links. This can result in higher maximum traffic for those jobs and impact their performance. At the same time, this suggests that if nodes assigned to jobs are not arbitrarily spread throughout the system, adaptive routing limits the impact of individual jobs to the groups they are placed on by the scheduler.

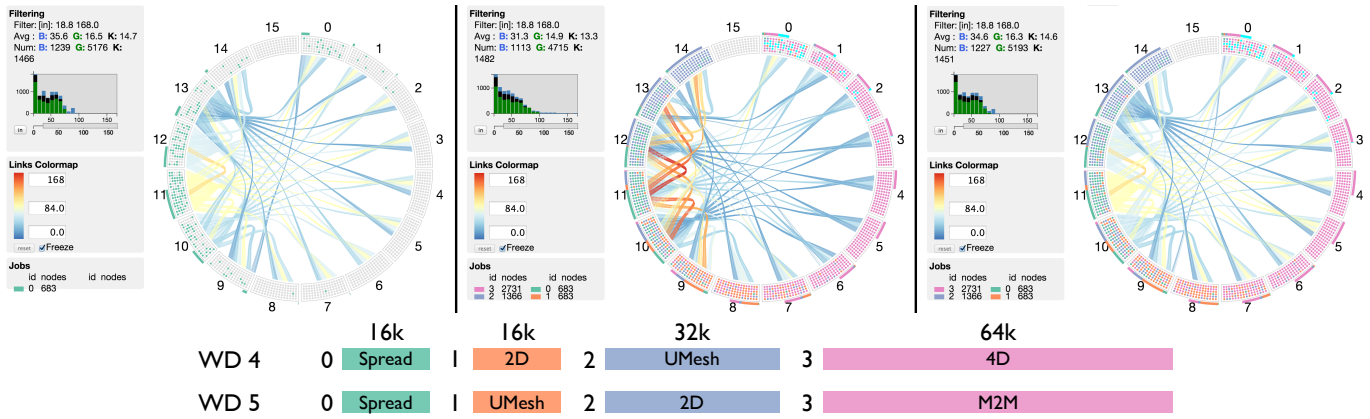

Figure 9. Maximum traffic on blue links attributed to Job 0 (Spread) increases when it runs in a workload (WD 4) alongside other jobs (middle) versus when run individually (left). However, if other jobs are not communication-intensive, the effect on a particular job might be minimal (Job 0, WD 5, right figure).

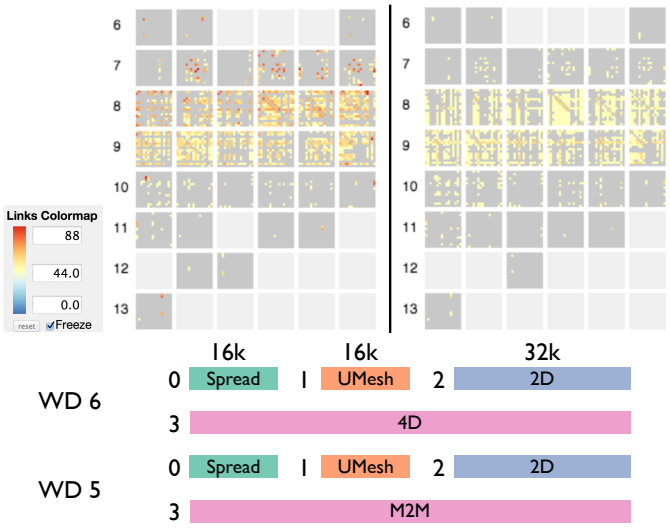

Figure 10. Max. traffic on green links for Job 1 (UMesh) is higher when situated near a communication-heavy job (4D Stencil) in WD 6 (left) than when situated near a communication-light job (Many-to-many) in WD5 (right).

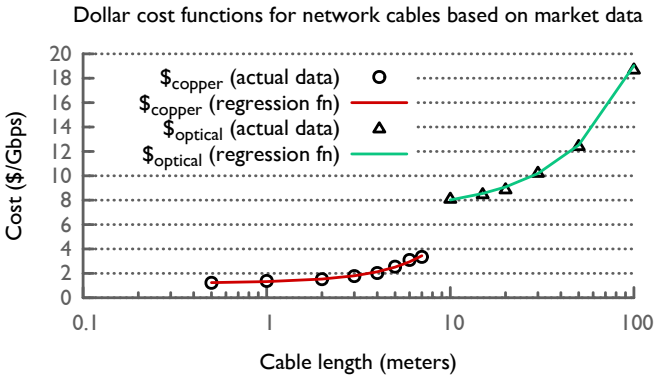

Figure 11. Dollar cost of copper and optical network cables as a function of the cable length in meters.

C. Network Wiring Configurations

Finally, we analyze the effects of changing the network wiring configuration on congestion and traffic. In the current configuration on Edison (referred to as the baseline), there is one green link per router pair, three black links per router pair and two blue links per router pair. All green and black ports on each Aries router are used. However, six out of ten blue ports are unused and can be connected using more cables to add additional inter-group or global bandwidth to the system.

Before we analyze the results of removing or adding links, we calculate the monetary cost of network cables used in Edison. Figure 11 shows the dollar cost per Gbps of a copper and optical cable based on current market rates [22]. Copper cables are used for the shorter intra-group black links, while optical cables are used for the longer inter-group blue links on Edison. Using regression, we obtain cost functions in terms of the cable length, x, as follows:

$$\$_{\text{copper}}(x) = 0.0288 \times x^2 + 0.123 \times x + 1.172 \quad (1)$$

$$\$_{\text{optical}}(x) = 0.0002 \times x^2 + 0.1 \times x + 6.995 \quad (2)$$

To estimate the average length of cables, we make reasonable assumptions about the machine room set up for a 15-group or 30-cabinet dragonfly. We assume that the cabinets are arranged on the machine floor in five rows of six cabinets each. Three pairs of cabinets in each row form three groups. Each cabinet pair is connected via black (copper) cables, while all groups are connected in an all-to-all fashion via blue (optical) cables. Based on these assumptions, we derive the average length of a copper cable to be ~3 meters and that of an optical cable to be ~11 meters. Substituting these lengths in Equations 1 and 2, the cost of a 5 GB/s copper cable is $72 and that of a 5 GB/s optical cable is $325 approximately.

With the knowledge that there are three black links per router pair and only four out of ten blue link ports on each router are used, we set up some experiments to model various potential modifications to Edison's baseline configuration. The backplane on each cabinet provides the green links, and thus we have not modified the configuration of green links. In our analysis so far, we have observed that the maximum traffic on

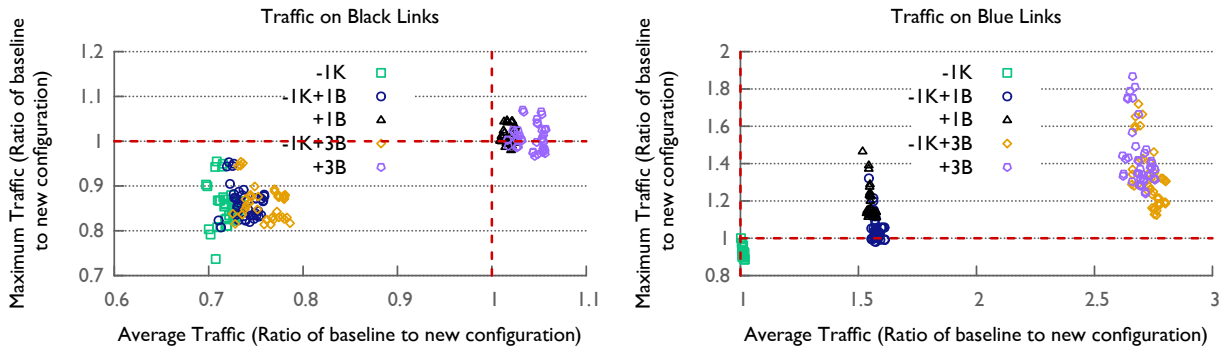

Figure 12. Scatter plots comparing the average and maximum traffic in the baseline configuration with the new configurations. If the ratio is greater than one, it reflects that the new configuration had less congestion. All 44 workloads in each of the five configurations are shown on each plot (each color/symbol represents one network wiring configuration.)

black links is 5 to 20% lower than on green links. We thus conducted five experiments centered on removing black links and/or adding blue links:

Remove 1 black link per router pair (–1K): Tests if the number of black cables on Edison is unwarranted and the potential impact of removing one out of three black links per router pair. *Cost: −$259,200* (potential saving!).

Remove 1 black link and add 1 blue link per router pair (–1K+1B): Trades intra-group bandwidth for inter-group bandwidth at a modest cost. *Cost: +$208,800.*

Add 1 blue link per router pair (+1B): Increasing the inter-group bandwidth allows us to explore whether this could mitigate the congestion on global links. *Cost: +$468,000.*

Remove 1 black link and add 3 blue links per router pair (–1K+3B): Significant change of balance in favor of inter-group bandwidth at a substantial cost. *Cost: +$1.15 million.*

Add 3 blue links per router pair (+3B): Adding three additional blue links per router pair uses all ten blue link ports and creates a configuration closest to a full-scale dragonfly design. *Cost: +$1.4 million.*

For experiments in this section, we ran all 44 workloads in each of the five network configurations (220 simulations in total). For each workload, we calculate the average and maximum traffic on each link type (green, black, blue) in the system, as shown in Figure 12. Since we did not modify any green link connections, the change in green link traffic is small and hence, we do not show that data. Each plot shows the ratio of traffic (average and maximum) of the baseline Edison configuration to that of a new configuration. If the ratio is greater than one, it means that the new configuration had a smaller value for the average or maximum than the baseline and hence is a better configuration in terms of performance. $x = 1$ and $y = 1$ dotted lines in red act as guides to show the good configurations. If a point is in the upper right quadrant formed by these dotted red lines, it is a better performing configuration than the baseline.

Figure 12 (left) shows that when we remove black cables (–1K, –1K+1B, –1K+3B), as expected, the average and maximum traffic over black links increase significantly. Interestingly, this also impacts the traffic on blue links in the –1K configuration possibly due to an increased use of certain blue links more than others. This is the only case for blue links where a new configuration performs worse than the baseline in terms of maximum traffic. In terms of the maximum traffic on black and blue links, the –1K configuration is worse than the baseline by 5 to 20% depending on the workload. However, it represents a saving of more than a quarter million dollars. This is a trade-off that the procurement division has to consider based on prior knowledge of the job mix on a future system.

As we only added blue links and never removed any, all new configurations perform better than the baseline with the exception of –1K (Figure 12, right). Even –1K performs better than the baseline in terms of average traffic (note that the x-axis in the right plot starts at a ratio of 1.0). Adding blue links on the system (+1B, +3B) reduces the average and maximum traffic over blue links significantly (up to 1.9 times in some cases). Figure 1 (baseline, +1B, +3B respectively) shows this significant reduction in the number of hot-spots as we add blue links on the network. However, this comes with a hefty price tag – adding one blue link per router pair costs nearly half a million dollars for a 33% reduction in maximum traffic and 10 to 30% reduction in average traffic. A good comprise might be removing a black link and adding a blue link if that does not impact performance significantly.

In order to analyze the overall impact of these network wiring changes, for each simulation, we pick the link color that has the highest maximum traffic and use the average and maximum values for that link type to plot its data point in Figure 13. A significant number of the +1B and +3B simulations are better than the baseline (above the $y = 1$ line). However, this plot shows that procurement decisions need to consider the tradeoffs between performance and dollar costs carefully with respect to the expected workloads on the system.

VI. Conclusion

Procurement, installation and operation of supercomputers at leadership computing facilities is expensive in terms of time and money. It is important to understand and evaluate

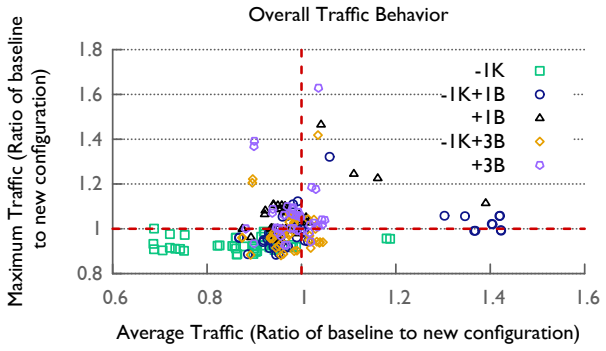

Figure 13. Plot comparing the overall traffic behavior in the baseline configuration with the new configurations.

various factors that can impact overall system utilization and performance. In this paper, we focused on analyzing network health and congestion as communication is a common performance bottleneck. Using the network configuration of a production supercomputer (Edison) as a baseline and five communication patterns, we studied inter-job interference as well as different network wiring configurations.

Based on our experiments, we conclude that in the default configuration, black links are usually not contended for. Depending upon the application pattern, the bottleneck is either on inter-group (blue) or intra-group (green) links. When multiple jobs execute simultaneously in a parallel workload, the communication of each job gets restricted to fewer links to provide a fair share of bandwidth to other jobs. This leads to higher bandwidth availability on other links that can be used by other jobs. Finally, we presented experiments that change the number of network cables (black and blue) on a dragonfly-based system. Removing one of the three black links per router pair has a small negative impact on the overall congestion in the network but leads to significant monetary savings. On the other hand, adding a blue link and removing a black link per router pair can mitigate hot-spots on inter-group links but requires an additional investment.

Insights presented in this paper, especially when coupled with the monetary costs of configuration changes can inform future purchases and potential upgrades. We have presented a simulation tool called Damselfly and a corresponding visual analytics system called DragonView that can be useful in performing such what-if analyses for current and future HPC systems. These tools can be used by machine architects, system administrators and end users to understand application, network and/or overall system performance.

Acknowledgment

This work was performed under the auspices of the U.S. Dept. of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-CONF-678293).

References

[1] "The Sierra Advanced Technology System," http://computation.llnl.gov/computers/sierra-advanced-technology-system.

[2] "Summit (OLCF)," http://www.olcf.ornl.gov/summit.

[3] "Aurora (ALCF)," http://aurora.alcf.anl.gov.

[4] "The Trinity Advanced Technology System," http://www.lanl.gov/projects/trinity.

[5] "Cori (NERSC)," www.nersc.gov/users/computational-systems/cori.

[6] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," *SIGARCH Comput. Archit. News*, vol. 36, pp. 77–88, June 2008.

[7] 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 *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '12. Los Alamitos, CA, USA: IEEE Computer Society Press, 2012.

[8] A. Bhatele, K. Mohror, S. H. Langer, and K. E. Isaacs, "There goes the neighborhood: performance degradation due to nearby jobs," in *ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '13. IEEE Computer Society, Nov. 2013, LLNL-CONF-635776.

[9] J. Brandt, K. Devine, A. Gentile, and K. Pedretti, "Demonstrating improved application performance using dynamic monitoring and task mapping," in *Proceedings of the 1st Workshop on Monitoring and Analysis for High Performance Computing Systems Plus Applications*, ser. HPCMASPA '14, 2014.

[10] W. T. C. Kramer and C. Ryan, "Performance Variability of Highly Parallel Architectures," in *Proceedings of the 2003 international conference on Computational science: PartIII*, ser. ICCS'03, 2003.

[11] D. Skinner and W. Kramer, "Understanding the Causes of Performance Variability in HPC Workloads," in *Proceedings of the IEEE International Workload Characterization Symposium, 2005*, 2005, pp. 137–149.

[12] N. Wright, S. Smallen, C. Olschanowsky, J. Hayes, and A. Snavely, "Measuring and Understanding Variation in Benchmark Performance," in *DoD High Performance Computing Modernization Program Users Group Conference (HPCMP-UGC), 2009*, 2009, pp. 438–443.

[13] J. J. Evans, C. S. Hood, and W. D. Gropp, "Exploring the Relationship Between Parallel Application Run-Time Variability and Network Performance in Clusters," in *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*, ser. LCN '03, 2003.

[14] A. Bhatele, N. Jain, W. D. Gropp, and L. V. Kale, "Avoiding hot-spots on two-level direct networks," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '11. ACM, Nov. 2011, LLNL-CONF-491454.

[15] T. Hoefler and M. Snir, "Generic topology mapping strategies for large-scale parallel architectures," in *Proceedings of the international conference on Supercomputing*, ser. ICS '11. New York, NY, USA: ACM, 2011, pp. 75–84.

[16] K. Underwood, M. Levenhagen, and A. Rodrigues, "Simulating red storm: Challenges and successes in building a system simulation," in *IEEE International Parallel and Distributed Processing Symposium (IPDPS '07)*, 2007.

[17] N. Jain, A. Bhatele, X. Ni, N. J. Wright, and L. V. Kale, "Maximizing throughput on a dragonfly network," in *Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '14. IEEE Computer Society, Nov. 2014, LLNL-CONF-653557.

[18] A. Singh, "Load-balanced routing in interconnection networks," Ph.D. dissertation, Dept. of Electrical Engineering, Stanford University, 2005.

[19] N. Jain, A. Bhatele, M. P. Robson, T. Gamblin, and L. V. Kale, "Predicting application performance using supervised learning on communication features," in *ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '13. IEEE Computer Society, Nov. 2013, LLNL-CONF-635857.

[20] A. Bhatele, N. Jain, K. E. Isaacs, R. Buch, T. Gamblin, S. H. Langer, and L. V. Kale, "Improving application performance via task mapping on IBM Blue Gene/Q," in *Proceedings of IEEE International Conference on High Performance Computing*, ser. HiPC '14. IEEE Computer Society, Dec. 2014, LLNL-CONF-655465.

[21] C. Bernard, T. Burch, T. A. DeGrand, C. DeTar, S. Gottlieb, U. M. Heller, J. E. Hetrick, K. Orginos, B. Sugar, and D. Toussaint, "Scaling tests of the improved Kogut-Susskind quark action," *Physical Review D*, no. 61, 2000.

[22] "COLFAX DIRECT: HPC and Data Center Gear," www.colfaxdirect.com.