LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-670006

Evaluating System Parameters on a Dragonfly using Simulation and Visualization

A. Bhatele, N. Jain, Y. Livnat, V. Pascucci, P. T. Bremer

April 24, 2015

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Evaluating System Parameters on a Dragonfly using Simulation and Visualization

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

[†]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA
[⋆]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL
[§]Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT
[†]{bhatele, ptbremer}@llnl.gov, [⋆]nikhil@illinois.edu, [§]{yarden, pascucci}@sci.utah.edu

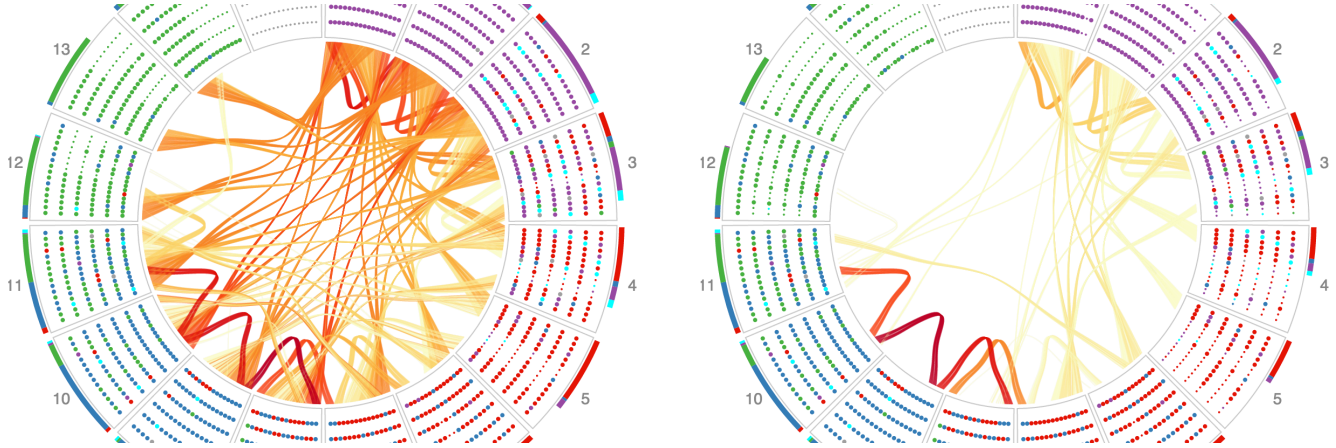

Figure 1: Visualization of the inter-group links on a dragonfly network for a parallel workload comprised of four jobs (colored circles). The left configuration has fewer inter-group cables connecting the groups which leads to more network hot spots in comparison to the network configuration on the right.

ABSTRACT

The dragonfly topology is becoming a popular choice for building high-radix, low-diameter networks with high-bandwidth links. Even with a powerful network, preliminary experiments on Edison at NERSC have shown that for communication heavy applications, job interference and thus presumably job placement remains an important factor. In this paper, we explore the effects of job placement, job sizes, parallel workloads and network configurations on network throughput to better understand inter-job interference. We use a simulation tool called Damselfly to model the network behavior of Edison and study the impact of various system parameters on network throughput. Parallel workloads based on five representative communication patters are used and the simulation studies on up to 131,072 cores are aided by a new visualization of the dragonfly network.

1. INTRODUCTION

System throughput, defined as the number of jobs retired over time, is a complicated metric that depends on several factors but indicates the overall performance of a supercomputer installation. System throughput depends on the hardware configuration (nodes and network), the job scheduling and placement policies and the mix of jobs that comprise the day-to-day workload among other things. Several of these factors depend on the type of interconnection network used to connect the processing elements.

Torus [13] and fat-tree networks [12] have been the topologies of choice for large supercomputer and cluster installations. However, the low degree and large diameters of torus networks have made them less attractive for building multi-Petaflop/s machines and they are being slowly replaced by fat-tree and dragonfly [10] topologies, as is evident from the results of the Trinity and CORAL procurement results. The dragonfly topology, in particular, boasts high-radix and low-diameter connections to improve overall network throughput and is being used in IBM's PERCS network [1] and Cray's Cascade network [7].

The general expectation has been that such highly-connected network topologies coupled with adaptive routing strategies would greatly reduce or eliminate the effects of job interference [4, 5, 11, 14, 16] and job placement in general. However, preliminary experiments on Edison at NERSC have shown that for communication-heavy applications, inter-job interference and thus presumably job placement remains an important factor.

This is in part because the current installations of the dragonfly network such as Edison at NERSC and Piz Daint at CSCS only use ∼1,400 routers compared to the equivalent full-scale design with ∼23,000 routers. This results in fewer network cables and a smaller bisection bandwidth compared to the full system. However, deploying smaller instances of

the full-scale design leads to an interesting degree of freedom as a large number of network ports on the high-radix routers are not used in smaller installations. These can be and have been re-purposed to selectively increase the bandwidth of certain connections as well as to change the topology of the network. However, the number of cables used represents a significant monetary cost and thus understanding the best practices for deploying a limited number of additional connections is of significant interest to the computing facilities. System throughput compared with monetary cost of the hardware can indicate the level of success of procurement, installation and operation of a supercomputer.

In this paper, we explore the effects of job placement, job sizes, parallel workloads and network configurations on network throughput to better understand inter-job interference. In particular, we focus on the Edison supercomputer installed at NERSC/Lawrence Berkeley National Laboratory for studying pre- and post-deployment system configuration issues. Edison is a 30-cabinet (15 groups) Cray Cascade system [7] based on the dragonfly topology [10].

We use a simulation tool called Damselfly to estimate the steady state behavior of the dragonfly network for various parallel workloads and network configurations. The simulation tool provides: 1) Access to hardware counters for all routers on the system, something impossible to gather in production; and 2) The ability to easily study the impact of removing or adding network cables on network throughput. Finally, we use visual analytics to explore, analyze, and illustrate the simulation data. In summary, this paper makes the following contributions:

- We present an enhanced version of Damselfly, a network simulator that can model the steady state behavior of dragonfly networks. It supports arbitrary network connections, different job placement policies and can simulate parallel workloads.
- We perform simulations for up to 131,072 cores that mimic the network configuration of and parallel workloads run on the Edison supercomputer at NERSC.
- Using simulations, we study the impact of five representative application communication patterns on other jobs in a parallel workload.
- We study the impact of changing the number of cables that connect different routers on Edison on network throughput and congestion.

2. RELATED WORK

Several researchers have investigated the impact of job placement on performance [6, 11, 3]. Skinner et al. [14] and Wright et al. [16] noted significant performance variability due to network contention. Recently, Bhatele et al. [4] 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. [8] developed analytical models for network traffic on different network topologies. Bhatele et al. [3] used BigSim [17] to model the PERCS network and study different placement and routing policies. SST [15] also supports various network topologies including the dragonfly network.

Several things distinguish our work from previous research – 1. We use a parallel simulator that implements an analytical modeling approach to network congestion. This enables us to perform large simulations very quickly, 2. Damselfly can simulate parallel workloads and record network traffic at the granularity of individual jobs, and 3. In this paper, we focus on changing the network topology itself and studying its impact on network throughput.

3. SYSTEM PARAMETERS AFFECTING PERFORMANCE

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 type and behavior of the actual jobs. In this study, we focus on the factors that affect network throughput 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 the network links. Using additional cables and increasing the bandwidth can improve the communication performance at a high monetary cost and with diminishing returns.

Job scheduling policy. The policy used to determine which jobs in the queue should run next directly impacts the overall system utilization i.e. the number of nodes being used at any point of time. Job scheduling is closely tied with the job placement policy.

Job placement policy. This determines the allocation of available nodes to new jobs. Node allocation can be done arbitrarily or by optimizing certain aspects of the available resources such as minimizing gaps by allocating contiguous blocks to each job. The Blue Gene family of supercomputers employs a network isolation policy and only allocates contiguous partitions that do not share network links with one another. This leads to more predictable performance and faster executions at the cost of lower system utilization and performance degradation due to fragmentation. In contrast, supercomputers such as the Cray XT/XE and XC families typically employ a topology-oblivious resource allocation policy that allow multiple jobs to share network links. Topology-oblivious policies often lead to higher system utilization at the cost of slower executions due to network interference and thus lower system throughput overall (number of jobs retired).

Routing policy. Another important factor deciding application and system performance is the global routing policy used for sending messages over the network. Static and shortest path routing policies can lead to congestion and hot spots on the network where a few inter-group links become the bottleneck. Adaptive, dynamic routing aims to route messages around hot spots and to avoid delays by employing random jumps through other nodes at the cost of using longer paths.

Previously, we have shown that a randomized placement or indirect routing can lead to good performance on the IBM PERCS machine [3]. We have also studied various routing choices for different communication patterns and parallel workloads. We found that the UGAL routing used on Edison is the best for minimizing congestion on the dragonfly network [9]. In this paper, we focus on choices made during

procurement and system installation to determine the number of cables used to build the machine and the impact of inter-job interference on network throughput.

In order to study inter-job interference and the impact of the network topology and links on system throughput, one needs information about the attributes of each message such as size and job id, which links each message was routed through and the variables per router that were used to determine each route. Unfortunately such comprehensive data is not available to end users on current production machines. Hardware counters on Edison provide only rudimentary aggregated information per link such as a rough measure of the overall amount of data transferred over a link and aggregated information about the number of stalled flits. Further, even this restricted set of information can only be collected on routers that are a part of the user's allocation and not system wide. Job scheduling and placement policies make it impractical to deploy a monitoring program on each and every router on a production machine. Naturally, experimenting with different machine configurations such as disconnecting or rearranging some cables is impossible for all practical purposes. To address all these limitations, we leveraged and expanded a network simulator we developed earlier [9].

4. APPROACH AND TOOLS

In this section, we describe our approach for studying the impact of different system parameters discussed in the previous section and the corresponding tools we use for the analysis.

4.1 Damselfly: Network Simulation

We use modeling and simulation for performing what-if studies and gathering data presented in this paper. This is because collection of empirical data from actual runs is expensive at times due to the large number of scenarios under consideration. Additionally, it is impossible to obtain empirical data for certain situations, e.g. there exists no mechanism to obtain link traffic of individual jobs executing in production on a shared network machine. We have used Damselfly to generate the traffic distribution on networks links for various parallel communication traces [9].

Given a network connectivity graph and an application communication pattern, the network model in Damselfly performs an iterative solve to iteratively redistribute traffic from congested links to less loaded links. The choice of how to redistribute and which less loaded links to use is determined by the routing policy. In our experiments, we have used a flavor of Edison's UGAL routing policy, wherein direct and indirect paths are selected based on their availability. For the purpose of this work, several new features have been added to the previously published version of Damselfly. These features are described below.

Simulation of arbitrary interconnect graph.: The previous version of Damselfly had pre-defined connectivity for routers in the system as described in [10]. While the inter-group connections could only be distributed in a round-robin manner among the routers of a group, the number of intra-group connections was restricted to one per router pair. In order to predict link traffic for Edison, both these restrictions were removed by letting the user specify an arbitrary interconnection graph. The user is free to specify any connection ordering for inter-group links, and router pairs can be connected via multiple links (which is the case for the column-wise all-to-alls on Edison).

Communication graph and placement for individual jobs. The second addition to Damselfly is an easier interface for providing the communication graph and placement for individual jobs executing in a workload. Now, the users can provide the MPI-rank based communication graph and the placements of various jobs on the system and Damselfly aggregates them to perform a full-machine simulation.

Link traffic for individual jobs. Finally, Damselfly has been extended to keep track of the traffic generated by each job executing in a parallel workload. Attributing the traffic on each link to individual jobs can be useful in studying inter-job interference and finding applications that negatively impact the overall system utilization.

4.2 Experimental Setup

In this section, we describe the experiments and scenarios we have simulated using Damselfly and various logistics related to performing the simulations. We use the network connectivity provided to us by NERSC system administrators for the Edison installation. Edison has 15 groups where each group is a cabinet-pair. Each group has 96 Aries routers that are strongly connected. The routers are arranged in a two-dimensional grid of 6 by 16. Routers in each of the six rows are connected in an all-to-all fashion by green links. Routers in each of the 16 columns are also connected all-to-all by black links. On Edison, the facility decided to put 3 black links between each pair of routers in the columns to utilize spare ports. Each router is also connected to several routers in other groups via blue links. The blue links are also used in 2-link bundles between each pair of routers.

We use a placement policy similar to that used on Edison where we try to locate most nodes of a job as close to one another as possible within the same group or adjoining groups. Some nodes of large jobs get scattered on the system depending on which nodes are available. We use five different communication patterns that are representative of some of the application codes run at NERSC:

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

4D Stencil: Each MPI process communicates with eight neighbors in a four-dimensional Cartesian grid (4 MB messages). This is representative of the communication in MILC [2], a Lattice QCD application widely used on NERSC machines.

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

Spread: Each process sends messages of 512 KB to randomly selected neighbors (the number of neighbors for different processes spans between 6 and 27). This pattern is supposed to add random background noise on the machine.

Unstructured Mesh: Each process sends messages of 512 KB to carefully selected neighbors (number of neighbors between 6–27). This is representative of an unstructured

Job Size (#cores)	Job 0 32,768	Job 1 32,768	Job 2 32,768	Job 3 32,768
WD 1	Spread	2D Stencil	4D Stencil	Many-to-many
WD 2	2D Stencil	4D Stencil	UMesh	Spread
WD 3	UMesh	Spread	Many-to-many	2D Stencil
WD 4	4D Stencil	Many-to-many	Spread	UMesh

Table 1: Parallel workloads (denoted by WD 1, 2, 3 and 4) and their constituent communication patterns.

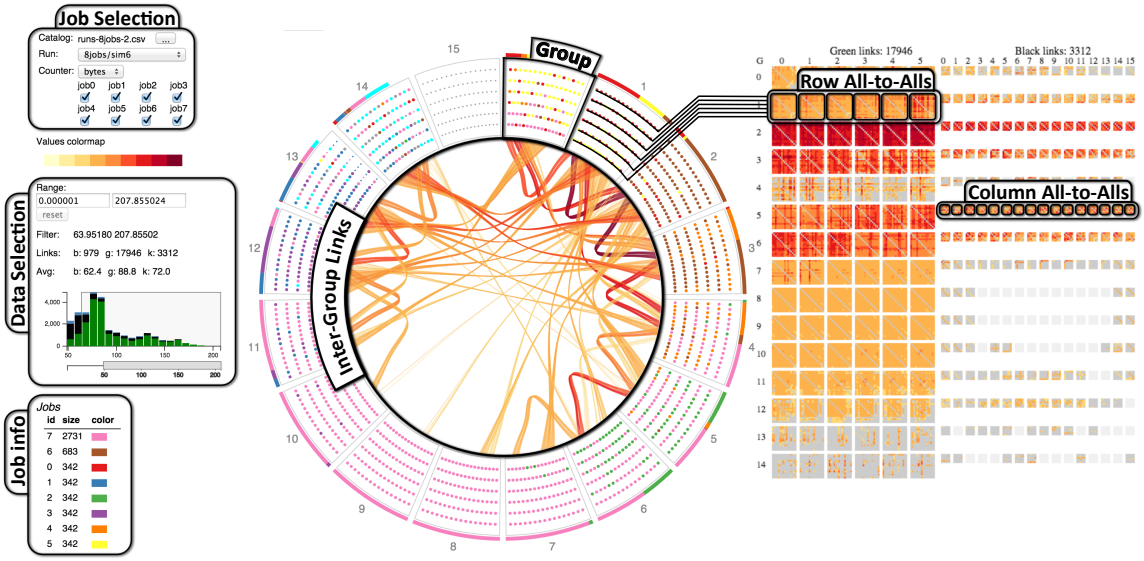

Figure 2: Visualizing a simulation of an 8-job run on Edison. On the left: run and counters selections; filtering via brushing over a histogram of links' counter values. Center: the radial layout depicts groups, routers and jobs placement along with network traffic over the inter-group links. Right: The 16x16 and 6x6 communication matrices show per-group all-to-alls links along the rows of routers (left) and columns (right).

mesh computation.

Using these five patterns, we create different parallel workloads that are a mix of these patterns running at different job sizes ranging from 8,192 to 131,072 cores. Even though we ran hundreds of parallel workloads for our experiments, we focus on four in this paper (described in Table 1).

4.3 Damsels and Dragons: Visual Analytics

Damsels and Dragons (DnD) is an interactive visual analytics system we developed to evaluate the impact of job placement policies, network interference and system configuration on the network utilization. The system can visualize data from simulation runs and from hardware counters collected directly on the Edison installation at NERSC. Direct comparison of different runs on a production system is not feasible as we do not have control over the job placement of our job nor of the selection of the other jobs running on the machine. Instead we focus on visualizing simulation runs consisting of one or more jobs while facilitating evaluation of the contribution of each job within that run.

The visualization system features several displays though for the purpose of this study we focus on the main radial view depicted in Figure 2. The display comprises of selection and filtering components on the left, a radial view of job assignment and inter-group (blue link) connections in the middle, and intra-group connections (green and black) on the right.

Controls.: The user interface on the left enables the user to switch between different runs and select which link attribute to visualize. The histogram depict the distribution of values associated with the selected attribute and the user can brush over the histogram to filter and select a subset of links. Direct comparisons of alternative scenarios (as in Figure 3) are easily achieved by setting side by side multiple displays with common basic setting and selective change of parameters of interest. This maximizes the ability to exploit the ability of the user to spot visually patters and understand similarities or differences due to the change of input parameters.

Jobs and inter-group links.: The central radial view of Figure 2 depicts the various groups as ring sectors each consisting of 96 routers (small circles) organized in a 6 by 16 grid. The color of the router indicates the job being run on the four nodes associated with the router based on the jobs color map shown on the left. A router with no assigned jobs is shown in gray while a router with several jobs running on its nodes is shown in cyan. Source and destination routers of the links selected via the histogram are shown as larger circles. The colored bands to the outside of the groups provide a quick reference to the proportional number of highlighted routers associated with each job in that group.

We depict inter-group (blue) links as bundled arcs in the interior of the ring. The arcs are colored using a yellow-orange-red colormap where red represents the maximum value. To reduce screen clutter the arcs do not span all the

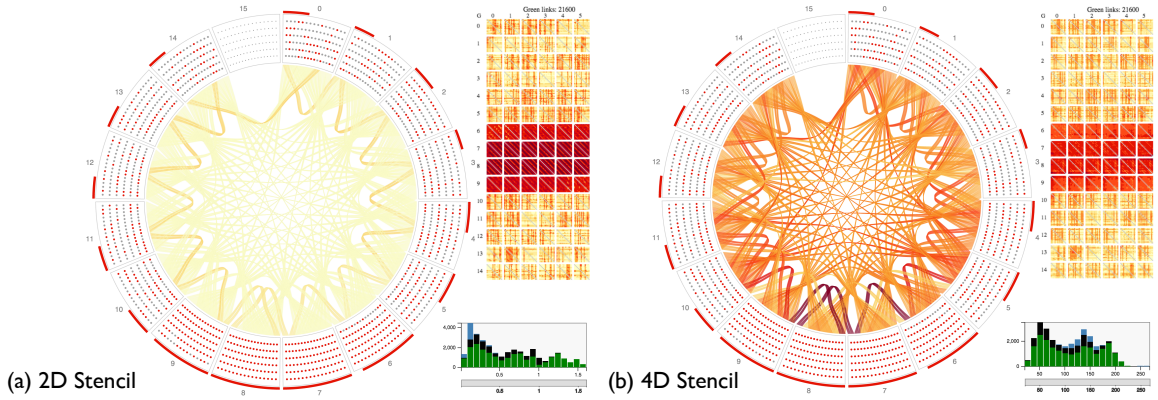

Figure 3: Traffic on blue (radial view) and green (matrix view) links for (a) 2D stencil and (b) 4D Stencil running on 64k cores (individual jobs).

way from their source to destination routers and instead start from the inner radius of the group's ring. We divide the inner edge of each group into 96 sections where each set of 6 sections are associated with one column of routers in that group. The arcs than start and end at the section corresponding to their source and destination routers. Users can also hover over a link or a router to highlight the link and the corresponding source and destination routers but we found that in general the visualization provide sufficient spatial relationship between the links and their routers.

Intra-group links.: The grid view on the right depicts inter-group communication matrices for each group. Each grid row consist of six green link communication matrices of size 16x16 and 16 black link communication matrices of size 6x6. The communications are colored using the same yellow-orange-red colormap used in the radial view. The communication matrices depict only the selected links and in order to assist in identifying which rows and columns have selected links we use a gray background for each such matrix. Light gray background indicates an empty matrix while dark gray a communication matrix with at least one selected link. The light gray backgrounds provide spatial visual anchors to help the user quickly comprehend the location of the active communication matrices.

With this user interface it is possible to explore in detail the trends in the data generated by the simulation and perform side by side comparisons of competing alternative to determine which option may be more effective.

5. INTER-JOB INTERFERENCE

We begin with a discussion of the experiments to study the traffic patterns for individual jobs of different sizes and how these patterns change when multiple jobs run together in a parallel workload.

5.1 Individual Jobs

We simulated the five communication patterns described in Section 4.2 with different job sizes ranging from 16,384 to 131,072 cores. Each simulation consisted of a single job occupying a part of the machine. Figure 4 shows the average and maximum traffic on the green and blue links in the system (black links are not shown because of lower utilization than the other two). We can clearly see that the average and maximum traffic on the links increases as we run larger jobs. It is also obvious that some patterns such as 4D Stencil and Spread send significantly higher traffic on the network compared to others (note the log scale on the y-axis).

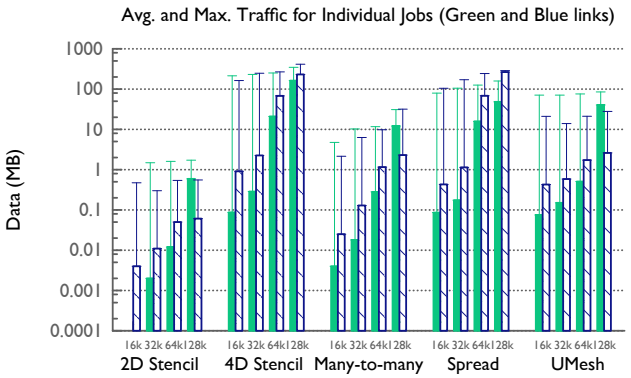

Figure 4: Average and maximum traffic on green and blue links for individual jobs with different communication patterns and sizes.

Maximum traffic through green and blue links is an indication of network hot-spots (black links were less utilized). As we can see, green links have higher maximum traffic than blue links for 2D Stencil and UMesh. The same applies to the Many-to-many pattern for lower core counts. However, blue links start to be the bottleneck for this pattern at 128k cores. For 4D Stencil and Spread, which are the communication-heavy patterns, blue links have higher maximum traffic. Both green and blue links are congested for 4D Stencil but blue links are the clear problem for Spread.

We can also derive similar conclusions if we look at the visualization of the traffic patterns for different simulations. Figures 3 and 5 show the bytes over various links for 64k-core simulations of 2D stencil, 4D Stencil, Many-to-many and Spread. It is to be noted that the value ranges for the four patterns are very different (the maximum is under 2 and 12 for 2D Stencil and Many-to-many respectively, and over 200 for 4D Stencil and Spread). We opted not to normalized the values to a single range in order to emphasize the particular traffic pattern of each. Green links are the primary bottleneck

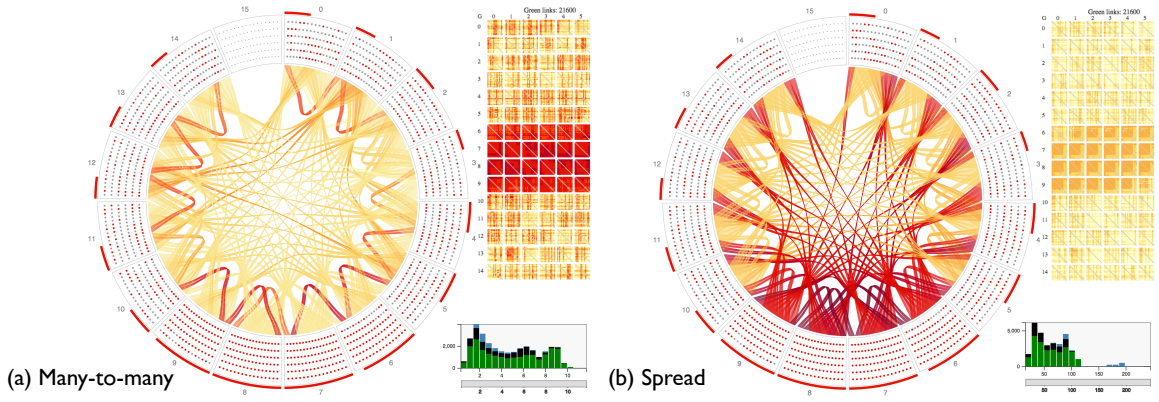

Figure 5: Traffic on blue (radial view) and green (matrix view) links for (a) Spread and (b) Many-to-many running on 64k cores (individual jobs).

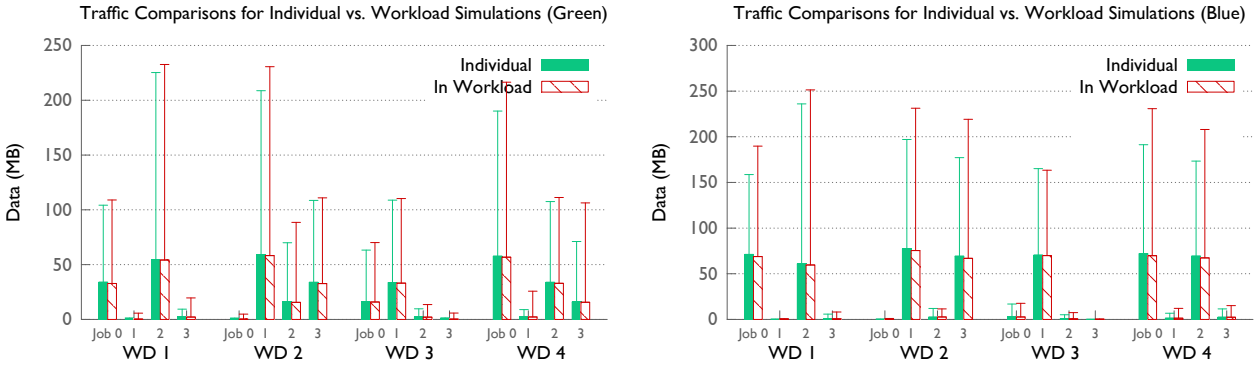

Figure 6: Average and maximum traffic on green and blue links for the jobs in the four workloads. The "Individual" bars show the traffic if the job was running by itself on the system. The "In Workload" bars show the portion of overall traffic in the parallel workload simulation that can be attributed to specific jobs.

for 2D Stencil (Figure 3 (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 3 (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 5 (a)). This effect becomes more pronounced on 128k-core runs. For Spread, blue links are the clear bottleneck with less pressure on green links (Figure 5 (b)).

5.2 Parallel Workloads

With the understanding of network congestion for each job and its communication pattern running by itself, we can shift our attention to the effect of running multiple such jobs together in a parallel workload. We simulated workloads consisting of four parallel jobs, each of size 32k cores as shown in Table 1. For each workload, we pick four out of the five patterns one of which is always Spread to ensure some background traffic or congestion.

In order to perform meaningful direct comparisons between network traffic generate by a job running with and without other jobs in a parallel workload, we run each workload in five settings. One setting consists of all the jobs running at the same time while the other 4 settings consist of running each job by itself but in the same placement as in the first parallel setting. Since the simulator outputs the link traffics for each job separately, we can then compare the average and maximum traffic over different types of links (green, black and blue) for the parallel workload versus individual job simulations.

Figure 6 compares the average and maximum traffic on green and blue links for the jobs in the four workloads. The *Individual* bars show the traffic if the job was running by itself on the system. The *In Workload* bars show the amount of overall traffic in the parallel workload simulation that can be attributed to particular jobs. As in the case of individual jobs, 4D Stencil and Spread generate a lot more traffic on the network compared to Many-to-many, UMesh and 2D Stencil. Considering WD 4 in the left plot, we can see the maximum traffic on the green links increases significantly for Job 0 (4D Stencil), 1 (Many-to-many) and 3 (UMesh). The average traffic on the green links, however, does not change much. For the blue links (right plot) the maximum traffic increases for Job 0 (4D Stencil) and 2 (Spread) in WD 4. This suggests that when jobs run in parallel they suffer from congestion due to other jobs creating more severe hot spots in comparison to with running alone.

Another interesting observation that is apparent only in the DnD visualization is that the adaptive routing appears to redistribute the traffic of each job to provide a fair share

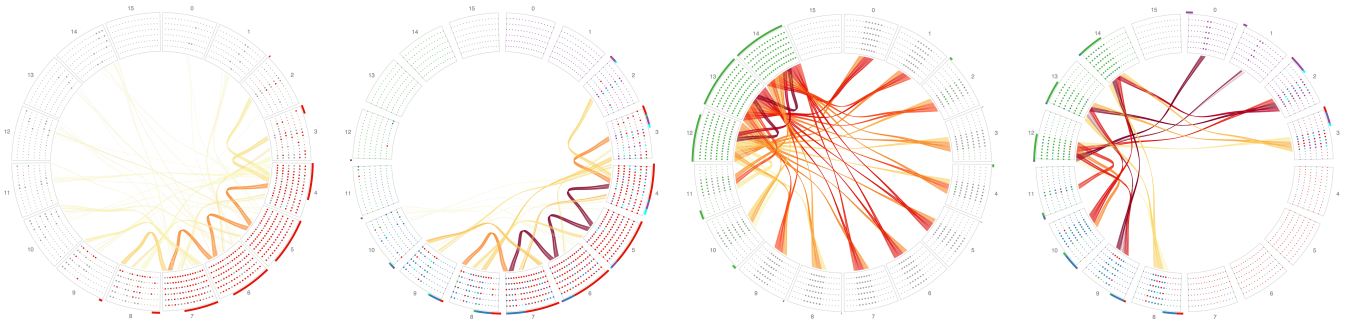

Figure 7: Traffic on blue links above a certain threshold that can be attributed to 4D Stencil: (a) running individually, (b) in WD 4, and Spread: (c) running individually, (d) in WD 4.

of bandwidth to other jobs. Figure 7 shows the blue link traffic for 4D Stencil and Spread when run individually and in WD 4 with other jobs (here we only show links with traffic above a certain threshold). We can see that when the two jobs are running alone on the machine, they use blue link connections between several groups to route their messages which likely increases their effective bandwidth. However, in a parallel workload setting, these indirect routes are also being utilized by other jobs leading to a heavier traffic load on the blue links directly attached to the jobs' allocated routers. This leads to the traffic for individual jobs being restricted to fewer blue links and thus increased congestion on them. However, this makes it possible for other jobs to use other blue links that are less busy now.

We see something similar in the green link traffic for Many-to-many and UMesh. When running by themselves, the jobs spread their traffic across most of the green links in the groups that are allocated to them (Figure 8). We can see that when running with other jobs, the matrices become sparser which indicates that the jobs are putting more traffic on some of the links to avoid links already utilized by other jobs. Note that, we only show links with traffic above a certain threshold, so the sparser matrices do not indicate fewer links being used.

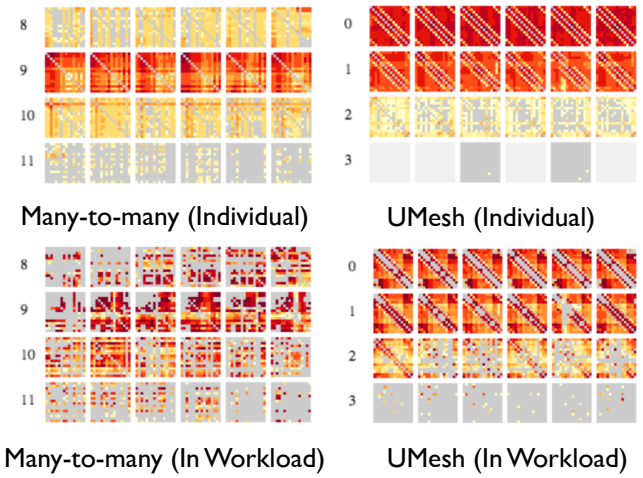

Figure 8: Traffic on green links above a certain threshold that can be attributed to Many-to-many running individually (top left), in WD 4 (bottom left), and Spread running individually (top right), in WD 4 (bottom right).

6. MODIFYING THE NETWORK

The number of cables to deploy on a smaller instance of the full-scale dragonfly design is a tradeoff between extra bandwidth and performance versus monetary cost. In our analysis so far, we observed that black links are seldom the most congested links for any of the workloads that we simulated. So we designed some experiments to remove some black cables from the system and analyze the impact on network throughput. These are the four experiments that we set up:

Remove 2 black links per router pair (-2K): On the Edison machine, there are three black links between each router pair. We wanted to test if this amount of bandwidth was unnecessary and what would be the impact of removing two of the three black links per router pair.

Remove 1 black link per router pair (-K): A less drastic change would be if we remove 1 black link per router pair.

Remove 1 blue link per router pair (-B): Although we know that blue links are an important commodity on dragonfly networks, we wanted to see the impact of removing 1 blue link per router pair. Note that on Edison, there are 2 blue links per router pair.

Add 1 blue link per router pair (+B-K): We could also add blue links in addition to the existing two blue links per router pair. For doing this, we removed a black link per router pair and added a blue link.

In Figure 9 we present a summary of the average and maximum traffic over different types of links for two of the workloads, WD 2 and WD 4. **DEF** represents the traffic for using the default configuration of Edison. Looking at the total traffic, we can observe that removing 1 black link (-K) does not increase congestion significantly. The maximum traffic on any link goes up by ∼10% compared to the baseline (DEF). On the other hand, if we add a blue link (+B-K) in addition to removing a black link the maximum traffic goes up only by 4% for WD 2 and goes down slightly for

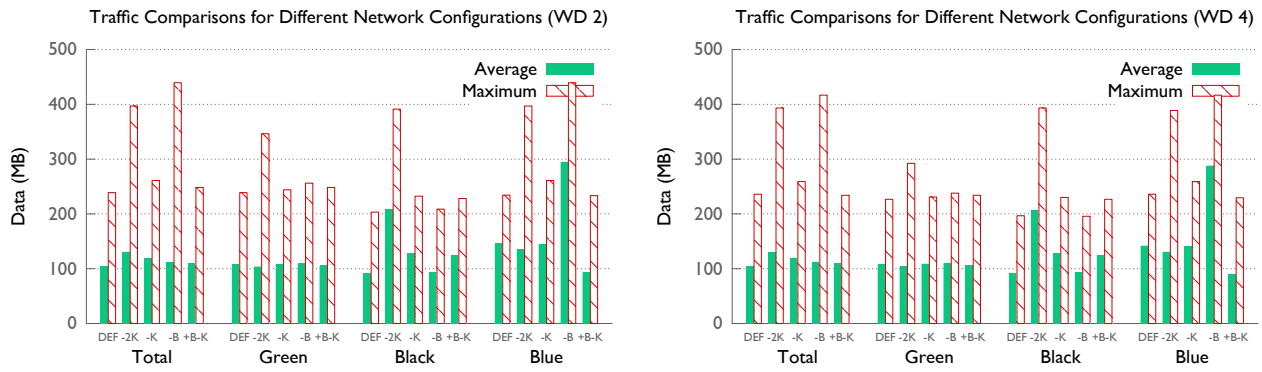

Figure 9: The impact of removing and adding black and blue cables respectively on the average and maximum traffic for different link types for two workloads, WD 2 (left) and WD 4 (right).

WD 4. So, depending upon our budget constraints, we could manage with two-thirds of the black cables on the system or use 33% fewer black cables and add 50% more blue cables.

Figure 10 shows the visualizations of network traffic for two specific cases (DEF versus +B-K) for workload, WD 2. We focus on links with traffic above a certain threshold (140 MB in this case). The first interesting thing to notice is the histogram on the left. In the baseline (DEF) case, blue links tend to be most congested followed by green links. This changes for +B-K because adding more blue links redistributes their load and removing black links brings them to the forefront in the histogram (and in the matrices on the right). We can also see that very few blue links (in the radial view) in the bottom figure suffer from congestion as opposed to that in the top figure. Hence, adding blue links can lower the inter-group congestion at the cost of a slight increase in black link traffic.

7. CONCLUSION

The procurement, installation and operation of supercomputers at leadership computing facilities is expensive in terms of time and money. It is important that we understand and evaluate various system parameters that can impact overall system utilization and performance. In this paper, we touch upon two aspects of system utilization – inter-job interference and the impact of the network configuration on congestion. Using the network configuration of a production supercomputer (Edison) as a baseline and five different communication patterns, we evaluated the impact of one job's traffic on other jobs. We presented a simulation tool called Damselfly and a visual analytics system called Damsels and Dragons (DnD), both of which can be useful tools for machine architects, system administrators and end users to understand network performance.

We observed that 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. We showed that when multiple jobs run in a parallel workload, the communication of each job gets restricted to fewer links to provide a fair share of bandwidth to other jobs. However, this leads to higher maximum traffic on the links. Again, this is observed on blue links for some patterns and green links for other patterns.

Finally, we performed some experiments that change the number of network cables (black and blue) on the dragonfly system. We found that removing one out of the three black links per router pair only has a small impact on the overall congestion in the network. However, adding a blue link and removing a black link per router pair can lower the hotspots on inter-group connections. Such knowledge coupled with monetary costs of different cables can help purchasing heads and system administrators in deciding the best network configuration for their parallel workloads.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This work was funded by the Laboratory Directed Research and Development Program at LLNL under project tracking code 13-ERD-055 (LLNL-TR-670006).

8. REFERENCES

[1] 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 *2010 IEEE 18th Annual Symposium on High Performance Interconnects (HOTI)*, pages 75–82, August 2010.

[2] 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*, (61), 2000.

[3] 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*, SC '11. ACM, Nov. 2011. LLNL-CONF-491454.

[4] 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*, SC '13. IEEE Computer Society, Nov. 2013. LLNL-CONF-635776.

[5] J. Brandt, K. Devine, A. Gentile, and K. Pedretti. Demonstrating improved application performance using

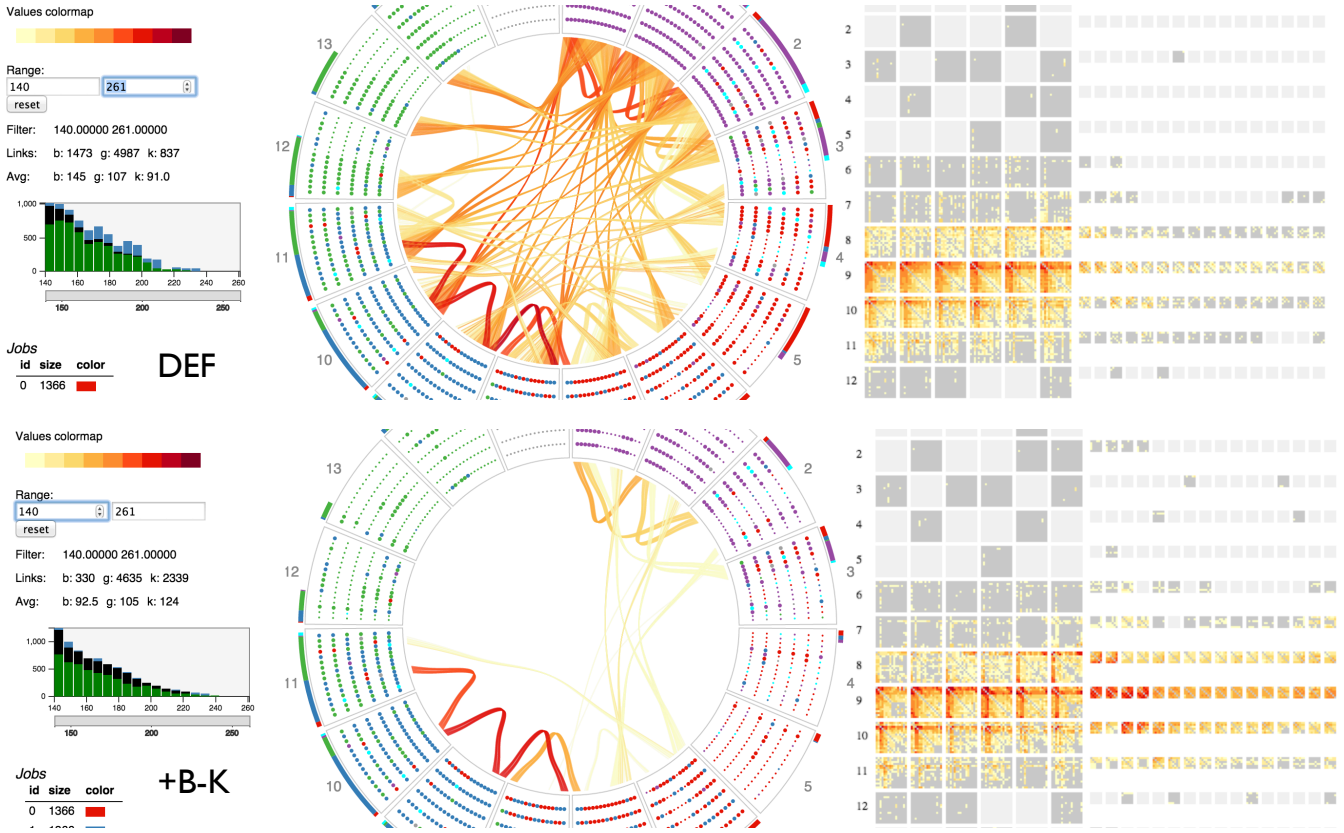

Figure 10: Visualization of the network traffic using DnD for two cases: the baseline (DEF, top) and replacing a black link per router pair by a blue link (+B-K, bottom). It is evident that adding blue links reduces the congestion on inter-group links significantly.

dynamic monitoring and task mapping. In *Proceedings of the 1st Workshop on Monitoring and Analysis for High Performance Computing Systems Plus Applications*, HPCMASPA '14, 2014.

[6] 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*, LCN '03, 2003.

[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*, SC '12, Los Alamitos, CA, USA, 2012. IEEE Computer Society Press.

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

[9] 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*, SC '14. IEEE Computer Society, Nov. 2014. LLNL-CONF-653557.

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

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

[12] C. Leiserson. Fat-trees: Universal Networks for Hardware-Efficient Supercomputing. *IEEE Transactions on Computers*, 34(10), October 1985.

[13] M.Blumrich, D.Chen, P.Coteus, A.Gara, M.Giampapa, P.Heidelberger, S.Singh, B.Steinmacher-Burow, T.Takken, and P.Vranas. Design and Analysis of the Blue Gene/L Torus Interconnection Network. *IBM Research Report*, December 2003.

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

[15] 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.

[16] 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*, pages 438–443, 2009.

[17] G. Zheng, G. Kakulapati, and L. V. Kale. BigSim: A Parallel Simulator for Performance Prediction of Extremely Large Parallel Machines. In *18th International Parallel and Distributed Processing Symposium (IPDPS)*, page 78, Santa Fe, New Mexico, April 2004.