SAND2018-8648C

Characterizing Supercomputer Traffic Networks Through Link-Level Analysis

Workshop paper: HPCMASPA 2018

Saurabh Jha*, Jim Brandt†, Ann Gentile†, Zbigniew Kalbarczyk*, and Ravishankar K. Iyer*
*University of Illinois at Urbana-Champaign, Urbana-Champaign, IL 61801
Email: (sjha8|kalbarcz|rkiyer)@illinois.edu
†Sandia National Laboratories (SNL), Albuquerque, NM USA 87123
Email: (brandt|gentile)@sandia.gov

***Abstract*—We present techniques for characterizing bandwidth and congestion characteristics of supercomputer High-Speed Networks (HSN). By utilizing a link-level perspective, we gain generality over analyses which are tied to specific topologies. We illustrate these techniques using five months of a Blue Waters production dataset consisting of network utilization and congestion counters. We find that: i) execution time of the communication-heavy applications is highly correlated to network stalls observed in the network topology and increase in application runtime can be as high as 1.7x with nominal increase in stalls, ii) heterogeneity in the available link bandwidth in the network can lead to back-pressure and congestion even when the network is not under-provisioned , and (iii) links connected to I/O nodes are no more likely to observe congestion during operational hours than any other link in the system.**

I. Introduction

Modern supercomputers contain 10s of thousands of compute nodes connected by a High Speed Network (HSN). While the technologies, topologies, and routing characteristics vary from system to system, they all have two common limiting characteristics: 1) finite bandwidth and 2) greater than zero latency. Because of these limiting characteristics they all exhibit performance impacting congestion under some common application and I/O communication scenarios. Understanding the performance impacting scenarios on a large scale system running production workflows can be very time consuming and often attribution for any particular performance degrading congestion event is impossible.

In this paper we present some technology, topology, and routing policy independent techniques for characterizing bandwidth and congestion characteristics of supercomputer HSNs. The prerequisites for using these techniques are the ability to synchronously capture and store appropriate HSN and I/O performance metrics. This paper provides a methodology for assessing the degree to which concurrent use of HSN resources by multiple consumers impairs communication between computational and storage elements of a supercomputer. As described in the paper, the fidelity of collection will impact the degree to which the behavioral characteristics can be resolved. The examples and use cases presented utilize data and application information from the National Center for Supercomputing Applications (NCSA) large scale Cray XE/XK system Blue Waters.

The rest of this paper is organized as follows: Section II introduces the problem and deployment details of the Blue Waters system. The types of data being collected as well as derived data definitions are presented in Section II-D. Section III provides evidence of impact of congestion on the run times of some benchmark applications. System wide characterizations are presented in Section IV followed by comparisons with congestion close to LNet routers in Section V. A topology independent visualization is presented in Section VI. Related work followed by conclusions are presented in Sections VII and VIII, respectively.

II. Background

Central to a supercomputers' performance is its High Speed Network (HSN) which enables high bandwidth, low latency communication between all of its computational and storage elements. There are a wide variety of network technologies and topologies currently being deployed on supercomputers.

Degradation in HSN performance due to oversubscription of its resources is commonly referred to as network congestion. In this paper we focus on the Cray Gemini based network of the National Center for Supercomputing Applications (NCSA) Blue Waters system. The approaches presented, however, are generally applicable to all HSN technologies and topologies for which applicable data can be acquired in a synchronized and periodic fashion.

A. Cray XE/XK Gemini Networks

Blue Waters is built on the Cray XE/XK compute platform which employs the Cray Gemini [1] network router Application-Specific Integrated Circuits (ASIC) as its fundamental HSN building block. The Blue Waters Gemini Network is a 3D Torus of dimension 24x24x24. Blue Waters is comprised of 27,648 nodes and utilizes a Lustre parallel file system for high performance storage.

On the Cray XE/XK platform, four compute *nodes* are packaged on a *blade*. Each blade contains a pair of Gemini ASICs. Each Gemini ASIC consists of 48 *tiles* each of which provides a bi-directional 3 bit link to a tile on another Gemini ASIC. Each bit in this case is referred to as a "lane" and a tile link is functional as long as at least one "lane" is active. The 3-dimensional torus network utilizes aggregates of Gemini tile links to form "directional links" in each of 6 directions, X+/-, Y+/-, Z+/-, in the torus. We will henceforth refer to these

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

"directional links" as d-links. An individual tile-to-tile link will be referred to as a t-link.

The maximum bandwidth for a particular d-link is dependent on the t-link type (e.g. electrical vs. optical have different signaling rates) in addition to the number (n) of t-links of which the d-link [2] is comprised. X and Y d-links have an aggregate bandwidth of 9.4 GB/s (564 GB/min) and 4.7 GB/s (282 GB/min) respectively. Z d-link aggregates are predominantly 15 GB/s (900 GB/min) but 1/8 of them are 9.4 GB/s (564 GB/min). Traffic is directionally-routed first in X, then Y, and finally Z dimensions. The shortest path, in terms of hops, in + or - is chosen for direction with a deterministic rule to handle tie breaking. A credit-based flow control mechanism is used [3] both within and between Gemini router ASICs to prevent data loss.

B. Network Congestion

In data networks, congestion is typically characterized by a decrease in a data stream's throughput, increase in the latency of its constituent components, or both, due to the offered load along its path exceeding the capacity of the network to handle it. The Cray Gemini network utilizes hop by hop credit based flow control to prevent data loss. Data buffered and queued to be transmitted from one network element to another, for which there are insufficient credits at the time it would be sent, results in a credit based "stall". Such a stall between elements within a Gemini ASIC is called an "inq stall" and between two t-links is called a "credit stall" (See Figure 1). Henceforth we denote a time interval as T_i with units in nano-seconds and time spent stalled over a time interval as T_{is} with units also in nano-seconds.

Definition 2.1: Average Time Stalled $(\overline{T_{is}})$ for any link with N tiles over a time interval (T_i) is:

$$\overline{T_{is}} = \frac{\sum_{n=1}^{N_{tiles}} T_{is_n}}{\sum_{n=1}^{N_{tiles}} T_{i_n}} \tag{1}$$

In this work we characterize congestion, using "credit stall" and "inq stall" metrics, as *Percent Time Stalled* in the following way:

Definition 2.2: Percent Time Stalled (P_{Ts}) utilizes the average time spent stalled over all n tiles of a link of interest over T_i

$$P_{Ts} = 100 * \overline{T_{is}}/T_i \tag{2}$$

Note that a "link" in the context of equations 1 and 2 can refer to a d-link, a t-link, or a link internal to a Gemini ASIC. We address the use of inter-/intra-router P_{Ts} as an indication of congestion severity in Section III.

Congestion in the HSN may result from a variety of causes including:

- Applications that use one-sided programming models such as PGAS (Partitioned Global Address Space) [4] and SHMEM (Shared Memory) [5] which performs all-to-all or all-to-one communication
- File system utilization patterns in an application
- Failure of network lanes and links causing either a decrease in available bandwidth or a change in application traffic patterns

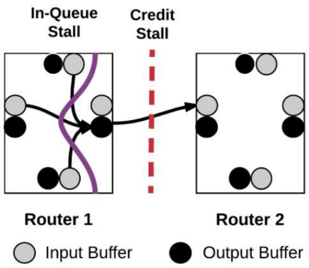


Fig. 1. In a credit-based flow control scheme, when a source router cannot send data to another router and needs to wait to send data until it has sufficient credits, it is referred to as credit stall. Likewise, within the router, when data cannot be transferred from input buffers to output buffers, it is referred to as inq-stall.

In extreme cases this congestion degrades system performance, and can cause the system to take action to protect itself and may even cause application runs to abort.

Because of the distributed nature of the network elements, an application's traffic may transit routers not physically associated with its node allocation and be in competition for network resources being used concurrently by other applications. The credit-based flow control can cause buffer depletion related backpressure, manifest as stalls, to spread backwards along communication paths. There is insufficient source attribution information available from the network performance counters for direct diagnosis of the root cause of a particular congestion occurrence. Finally, the combination of directional-order routing rules combined with the mismatch in directional bandwidth can itself cause network congestion.

C. Network Utilization

Another quantity of interest is the bandwidth utilization.

Definition 2.3: Percentage bandwidth utilized (P_{bu}) for any d-link over a time interval (T_i) is:

$$\begin{aligned} P_{bu} = 100 * ((BC_c - BC_p)/T_i))/ \\ ((BW_{t-link}) * (N_{t-links})) \end{aligned} \tag{3}$$

Where BC_c = current d-link byte count, BC_p = previous d-link byte count, BW_{t-link} = t-link bandwidth in (B/s), and $N_{t-links}$ = number of t-links in a particular d-link.

Note that the number of application data bytes injected into the HSN is not the same as the number of bytes transiting the network as a result of that injection. This is due to the automated hardware data compression performed upon injection into the network; experiences on the impact of this compression were reported by Pedretti et al [2]. An interesting artifact of this is that the same application network communication patterns may cause different levels of HSN traffic/congestion depending on how well the data being injected compresses.

D. Data Sources

On Blue Waters, HSN performance counter data as well as a variety of other node and system resource utilization information (e.g., Lustre file system and RDMA bytes read and written) are periodically sampled and stored, in a common database, for analysis using the Lightweight Distributed Metric Service (LDMS) monitoring framework [6]. The HSN performance counter data is sampled using Cray's *gpcdr* [7] kernel module. Sampling of all data is performed, synchronously across all nodes, at 60 second intervals. Synchronization is performed in order to provide coherent *snapshots* across the whole system. Note that clock skew is not accounted for in the node clock based synchronization.

The HSN related information is utilized for gaining an understanding of how congestion levels are driven by d-link bandwidth utilization. While *gpcdr* provides raw information on how much time was spent in stalls or how many bytes crossed a d-link, translating these into percent of time spent in stalls (P_{Ts}) or percent of d-link bandwidth utilized (P_{bu}) requires the additional information and processing described in Sections II-A, II-B, and II-C.

III. Impact of Congestion in Blue Waters Network on HPC Benchmarks

Application runtimes may be severely impacted by a seemingly small degree of network congestion along its communication paths (i.e., $P_{Ts} \sim 10 - 15\%$). Accurate detection and response to congestion can therefore, result in substantial performance improvement. In this study, the baseline runtimes of the applications launched by Blue Waters users (i.e., in the absence of failures or network congestion) was not known. Therefore, to estimate the impact of congestion on application runtime, we ran two representative HPC applications in production on 256 Blue Waters nodes:

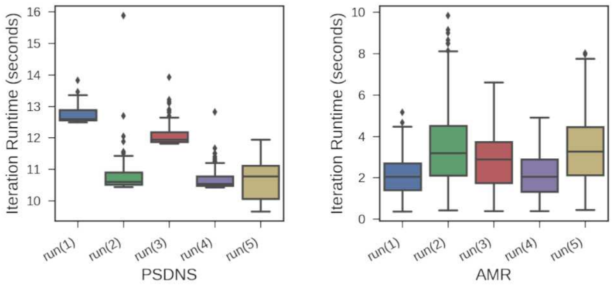
- PSDNS [8] : PSDNS is a highly parallelized application code used for performing direct numerical simulations (DNS) of three-dimensional unsteady turbulent fluid flows, under the assumption of statistical homogeneity in space. It solves partial differential equations using Fourier pseudo-spectral methods which requires multiple FFTs to be taken in three directions per time step, resulting in communication-intensive operations due to transposes involving collective communication among processors. We configured PSDNS to run for 300 time steps.
- AMR [9]: AMR is mesh restructuring algorithm for adaptive mesh refinement computations. The parallel mesh restructuring algorithm operates in terms of near-neighbor communication among individual blocks, and a single synchronization-only collective. Meshing occurs at discrete time steps. We configure AMR to run for 2,200 time steps.

For both applications a time step is equivalent to one iteration in the application. Allocations and workload mix, and thus contention conditions, were subject to natural production variance.

Figure 2 shows the boxplot of the compute time for each time step labeled as ”Iteration Runtime” (on y-axis) for each run of the application. Comparing the boxplots of the different runs of the same application allows us to compare the distribution of the iteration times. Minimum and maximum iteration time (in seconds) across five runs were respectively ~ 9.7 and 16 seconds for PSDNS and ~ 0.2 seconds and 10 seconds for AMR. While all iterations may not be the same (in terms of required compute resources) in an application run, we expect their distributions to be the similar across runs. Variations in the distributions can indicate possible HSN congestion during the application runs. We also compare the total application runtime. Minimum vs. Maximum total run times (in minutes) were 41 vs. 70 for PSDNS and 59 vs. 93 for AMR.

In order to determine the relationship between P_{Ts} values and application performance impact, ideally we would use the values along each of the application's communication paths when communication is occurring. Since the communication occurrences cannot be resolved at the fidelity of collection, as an approximation we consider the P_{Ts} on d-links directly associated with the application's topology. The correlation value between the application runtime and stall was found to be 0.87 for PSDNS and 0.96 for AMR.

PSDNS run(1) and run(3) suffered an average P_{Ts} of 11% and 8% respectively while AMR run(2), run(3) and run(5) suffered an average P_{Ts} of 12%, 7% and 11% respectively. Less than 4% P_{Ts} was observed for all other runs.



(a) PSDNS

(b) AMR

Fig. 2. Boxplot of iteration time for [a] PSDNS code and [b] AMR code running on 256 nodes

A. Summary and Insights

The benchmarks results demonstrate that even at marginal congestion levels (P_{Ts} of 12% in our benchmark runs), the slowdown of the application was observed to be as high as $1.7X$ with a corresponding loss of 123 node-hours. For large-scale applications, congestion and corresponding application slowdown can in practice be much larger therefore leading to loss of large amounts of computational node-hours.

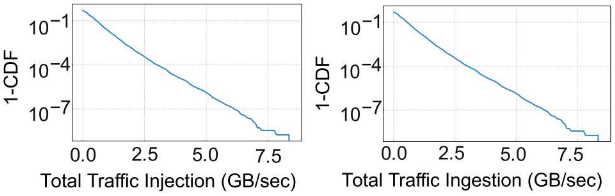
Further, our results show strong correlation between values of P_{Ts} values and application runtime, thus justifying our choice of this metric to analyze the dataset in the rest of the paper.

IV. System-wide Link-Level Characterization Results

We studied five months of a Blue Waters production dataset (January 2017 - May 2017) to characterize (a) the overall network injection and ingestion rate, (b) the d-link-level utilization and, (c) stall characteristics.

A. Traffic Injection and Ingestion Rates

Figure 3(a) and Figure 3(b) show the complementary CDF (CCDF), i.e., $1 - CDF$, of the sum of traffic injection (from nodes to the network) and ingestion (from network to nodes) by all nodes during measurement periods, where each measurement period is sixty seconds, across the whole study period. The complementary CDF is used to analyze the tail distributions of a metric and is used to ask how often a random variable is above a particular level. Both ingestion and injection rates are linear in log-scale indicating that applications ability to use the Blue Waters network bandwidth across the d-link exponentially decreases with an increase in both ingestion and injection rates.



(a) Traffic Injection by all Nodes

(b) Traffic Ingestion by all Nodes

Fig. 3. Node Traffic Statistics

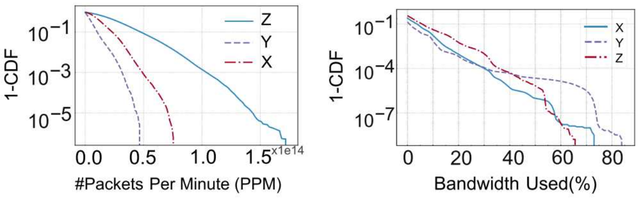
B. *Link utilization statistics*

Characterization at the d-link level gives insights into congestion effects that result from differences in directional bandwidth, application communication, patterns and from the directional aspects of the routing rules (described in Section II-A). Figure 4 shows the CCDF distribution of traffic (in terms of number of packets) and load (in terms of % of the d-link bandwidth utilized) across d-link-minutes in the study period (i.e., a d-link used for a minute). In this figure a higher CCDF value along the y-axis for a particular Packet-Per-Minute (PPM) value PPM_p along the x-axis means that more d-link-minutes (i.e., measurement samples) were observed to attain values greater than PPM_p. As can be seen in Figure 4(a), the PPM probability $PPM > x$ is always higher for d-links in the 'Z' direction and lower for 'Y'. This may be expected, since 'Z' is the final direction in the routing rules and hence cannot be stalled waiting on additional directional changes. In addition, there is a potential mismatch and possible downgrade in bandwidth affecting d-links in the 'X' direction since 'Y' has half the number of t-links that 'X' has, despite the potentially higher bandwidth of some of the 'Y' t-links. As a result, 'Y' can become a bottleneck which reduces the PPM in the 'X' direction.

Figure 4(b) shows the relatively low d-link bandwidth utilization seen in this study in which only 11% of 'Z', 2.7% of 'Y', and 7.5% of 'X' d-links experienced bandwidth utilizations of greater than 5%. The percentage of d-links that use higher amounts of bandwidth decreases exponentially with increasing bandwidth. For example, a CCDF value of around $10^{-4} \sim 10^{-6}$ is observed for achieved bandwidth utilization of more than 50%. For a fixed utilization (up to 30%), the number of d-links that achieve this utilization is highest along d-links in the 'Z' direction, followed by those in the 'X' direction, and 'Y' direction. However, there are more 'Y' d-link minutes that achieved utilization greater than 40% compared to 'X' and 'Z' direction. The reason for the difference can be attributed to limited available bandwidth for d-links in 'Y' direction.

C. *Link stall statistics*

Figure 5 shows the CCDF distribution of percent time spent in inq- and credit-stalls (P_{Ts}) per d-link-minute. A higher directional (X, Y, or Z) CCDF value for a particular P_{Ts} (P_{Ts_p}) implies more d-link-minutes spent at P_{Ts} values greater than P_{Ts_p}. The percent of time stalled waiting on buffer credits for both the inq and credit cases, for any given point on the x-axis (in Figure 5) are seen to be highest for 'X' (up to about 75%). This is consistent with expectations because of directional-order routing and directional bandwidth differences. This observation is in alignment with observations of higher stall counts in 'X' for known traffic injection levels [2].

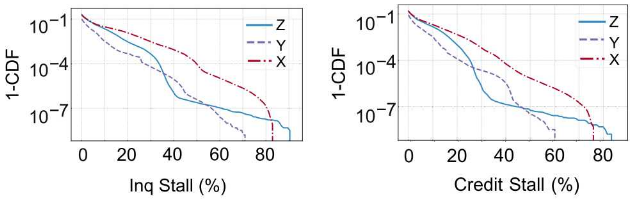


(a) Complement CDF of #packets flowing per minute for a d-link

(b) Complement CDF of bandwidth utilization of a d-link measured at one minute interval (in %)

Fig. 4. *D-Link* utilization statistics.

In comparing the CCDF values for both inq and credit related P_{Ts} in Figures 5(a) and 5(b), the CCDF for inq tends to be higher than for credit. For example, the percent of d-link-minutes with inq P_{Ts} greater than 10% in 'X'-dir is 4.1% whereas the corresponding credit CCDF value is 3.1% [1]. This is because the packets are first routed internally within the router and then on the d-link. The resource (buffer) contention within the router is higher because of inbound packets (coming from different directions) competing for buffers within the router while the outbound packets (leaving the router) wait for the buffers to be free on the receiving end (i.e., in the next router) along this d-link (see Figure 1).



(a) Complement CDF of inq stall across d-link minutes along 'X', 'Y' and 'Z' dir

(b) Complement CDF of credit stall across d-link minutes along 'X', 'Y' and 'Z' dir

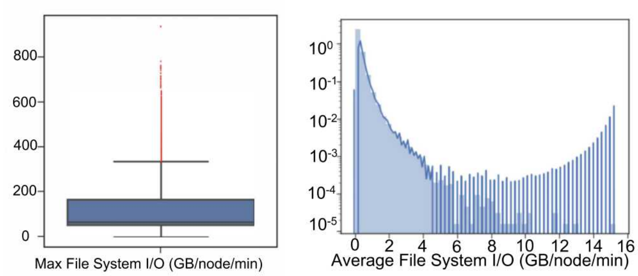
Fig. 5. *d-link* Stall Statistics

Table I presents the correlation between d-link utilization and stall values which indicates a moderate positive correlation between the two. Note that the measurements were taken at a sixty second granularity which averages out bursty utilization and stall values associated with the d-link. The moderate positive correlation is expected because stalls on the d-link increase with increasing traffic. The increase in traffic corresponds to a higher occupancy of the buffers on the routers and thus a decrease in available credits for additional traffic. After a certain threshold P_{Ts} value, further increases in stall rates severely decreases the network traffic flow. Due to the granularity of data sampling, we were unable to capture this threshold value. We do, however, show the coarse grained linear relationship between the two metrics.

[1] It is difficult to see the difference between 4.1% and 3.1% in the figure due to the log-scale representation of CCDF

TABLE I
PEARSON CORRELATION COEFFICIENT BETWEEN *Link* UTILIZATION AND CONGESTION

Direction	Credit	Inq
X	0.60	0.52
Y	0.50	0.41
Z	0.66	0.65



(a) Boxplot of maximum I/O rate across all nodes

(b) PDF of average I/O Rate across all nodes

Fig. 6. I/O Statistics

D. Summary and Insights

Differences in the available link bandwidth along 'X', 'Y', and 'Z' directions lead to significant differences in an applications' ability to inject/ingest data into/from the network due to congestion (caused by back-pressure along the path) along these directional paths. In general, the 'X' direction suffers the highest congestion due to directional-order routing and bandwidth limitation in the 'Y' direction.

Moreover, our results indicate that the network is not under-provisioned as only a few links ($\sim 7\%$ of the measurement samples) use more than 5% of the available bandwidth. However d-links do occasionally attain more than 50% bandwidth utilization (tail utilization in Figure 4(b)). The resolution of the datasets hides any bursty traffic that is sustained for less than a minute thus making it difficult to quantify the effects of bursty traffic on congestion and application runtime.

V. CHARACTERIZING LNET CONNECTED LINKS

This section characterizes the role of LNet nodes in network congestion. To facilitate this research question, we first characterize the I/O behavior of the Blue Waters system, followed by comparing the probability of d-link congestion around the LNet nodes with that seen in d-links associated with non-LNet nodes in Blue Waters. There are 576 LNet nodes in the Blue Waters system distributed evenly in the network.

A. Blue Waters I/O Characteristics

Users of the Blue Waters system submit a variety of jobs with different network and I/O access behaviors. Figure 6(a) shows a boxplot of maximum I/O (read + write) and Figure 6(b) shows the probability density function (PDF) of average I/O per node per minute across all nodes of the system. Although, there are nodes in the network which do very heavy I/O at any given time in the system, even during the 5% of the period of study with the heaviest I/O, the average rate never exceeded 1 GB/min (less than 0.4% of the slowest d-link in the network).

B. Link Stall Statistics

Next, we compute the CCDF metric for P_{Ts} of 'inq' and 'credit' for the X, Y, Z d-links that are directly connected to the routers of the LNet node. The CCDF metric is compared with the P_{Ts} of 'inq' and 'credit' stall metrics computed for all of the d-links (Figure 5) of Blue Waters as shown in Figure 7 and 8. It can be seen from the figure that LNet-connected d-links are no more likely to get congested than any other d-link in any direction in the network. However, there is a higher probability that a non-LNet-connected d-link is more susceptible to tail-end congestion (high stall values) as compared to LNet-connected d-links. We further confirmed this observation by randomly sampling 576 nodes of the Blue Waters system thirty times and calculating ks-statistics to find if there are any observable difference between the Blue Waters non-I/O node connected d-links versus I/O-node connected d-links. The p-value for ks-statistics was found to be 0.8. Therefore, we cannot reject the null hypothesis that the distributions of the two samples are the same. The result, however, should not be generalized to bursty traffic patterns and needs additional correlation study between bursty I/O traffic and the likelihood of congestion on the d-links. The ks-test statistics generally fail to differentiate the tail-ends of distributions and can mask the presence of rare-events.

C. Summary and Insights

The metrics used for characterizing the role of LNet nodes in d-link congestion show that on average, LNet-connected d-links are no more likely to congest than any other d-link in the network. However, the presented result is for the average case and we cannot rule out the possibility of congestion due to short burst traffic or due to sustained (greater than or equal to 60 seconds) high-bandwidth data transfer (e.g., tail-end of the distribution in Figure 6).

VI. CONGESTION EVOLUTION VISUALIZATION

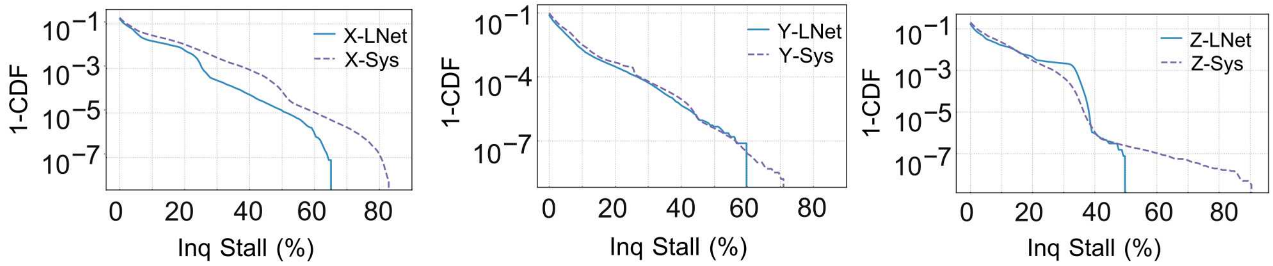
We can use the d-link data values to create visualizations that can help us to understand system utilization over time or congestion evolution.

Visual representations must provide information in a manner in which the desired information is easily understood. Networks seek to minimize the number of hops between endpoints and thus are multi-dimensional. This multi-dimensionality does not lend itself to simple visualizations. As a result, many visualizations which seek to capture the full topology are difficult to read and attempts to simplify this through reduced dimensionality (including projections) or dropping links can result in a representation perhaps even more difficult to understand (e.g., projections and slicing of a 5D torus in [10]).

For evolution, time is a significant variable and attempts to capture both time and space put significant limitations on the ability to provide simple visualizations. Attempts to capture only space require time for a human to look at a sequence of point-in-time events.

Here, we opt to drop the details of the spatial representation to emphasize time. We have seen in Section IV-C that congested d-links are infrequent. Thus a time-focused representation could enable easier discovery of congestion occurrences, since the display would be sparse.

For each timestep, we use the topology coordinates to determine contiguous groups of d-links whose values exceed some threshold(s) and group these into a *feature*. Each feature is characterized by its extent and severity. Since we consider

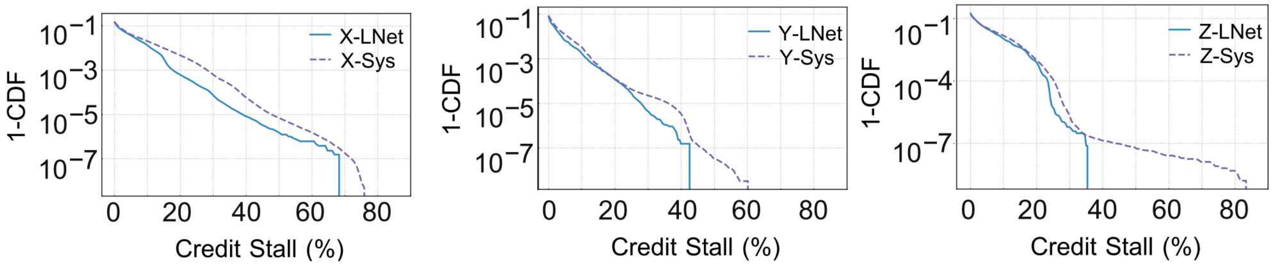


(a) Complement CDF of inq stall across X-dir d-link minutes

(b) Complement CDF of inq stall across Y-dir d-link minutes

(c) Complement CDF of inq stall across Z-dir d-link minutes

Fig. 7. Link Stall Statistics



(a) Complement CDF of credit stall across X-dir d-link minutes

(b) Complement CDF of credit stall across Y-dir d-link minutes

(c) Complement CDF of credit stall across Z-dir d-link minutes

Fig. 8. Link Stall Statistics

each topological direction separately, each feature will be at most as big as the number of routers in that direction. Features are extracted from all locations in the torus, so that the maximum possible number of features at any point in time is $\prod_{i=0}^{D-1}(Nrouters_i/2)$, where $D-1$ = number of orthogonal dimensions. For the 3D torus, these are, for example, Y and Z, if the feature value is stalls in X; for Blue Waters then each feature will be at most 24 units in size and there will be a maximum possible 144 features per time step. (In this section only, we do use topology information to determine contiguous d-links. We do not, however use any information about routing nor require a 3D torus)

If a coordinate is in two features in sequential timesteps, that indicates an evolution of the feature in time. Features can thus split and merge in the next timestep. Coordinates in features that do not persist over sequential timesteps will not be associated with each other (e.g., features at $t0$ and $t2$ may comprise the same d-links, but they will not be recognizably associated unless those d-links also are in a feature at $t1$).

In our visualization, each *individual* feature evolution is a directed acyclic graph, where each feature is a vertex, sized by its extent in the topology and colored by a value (in this case the maximum value in a feature), with directed edges indicating evolution of one feature to another. All other location information is dropped from consideration. A benefit of a graph representation is that a variety of techniques exist for graph comparisons. Graphs are drawn using Graphviz [11].

Multiple individual feature graphs may co-exist in time. We call any representation, regardless of number of individual evolving features or timerange, a *feature graph* and, in the case where the features represent congestion, a *congestion evolution graph*. Such a graph, with detailed examination of subsections, is shown in Figure 9. Figure 9(a) on the left, in the black box, is a representation of the congestion over a full 24 hour production day. Time starts at the top at midnight and progresses downward. Occurrences of congestion based on P_{Ts} of credit stalls in X+, are marked on the right by the red arrows and labeled by approximate duration; non-congested occurrences are marked in green. Over this entire day, there are 3 major times of lasting congestion: at about 2:30 pm (16.5 hours after midnight) lasting for 2.25 hrs, at about 7:00 pm with 2 graph sections slightly separated in time, and at about 10:00 pm again in two graph sections slightly separated in time. (There is also a short duration event at about 10:00 am). This representation provides full day information compactly.

A ~ 1 hour occurrence at about 8:00 pm is outlined in red; this is enlarged to show more detail in (b) in the upper right of the figure, also outlined in red. Time increases from top to bottom; this is too small to resolve in the figure, however each line presents features at that sampling time, so in this case they are at one minute intervals. The horizontal layout is arbitrary with respect to actual topological location and is determined by Graphviz. Features aligned vertically are only associated if they are connected, however, in most cases, a stack of features in a graph is a connected set, and thus does represent the evolution of overlapping features. Each individual feature at any given time is uniquely numbered (too small to resolve in the figure) so that it can be associated with the raw data for further investigation.

Note that the features, their characteristics, and the graph will be dependent on the resolution chosen. In this example only values of $P_{Ts} >= 40$ are included in the graph (legend in the figure). These are significantly higher values than those evinced in the variable performance benchmark runs in Section III.

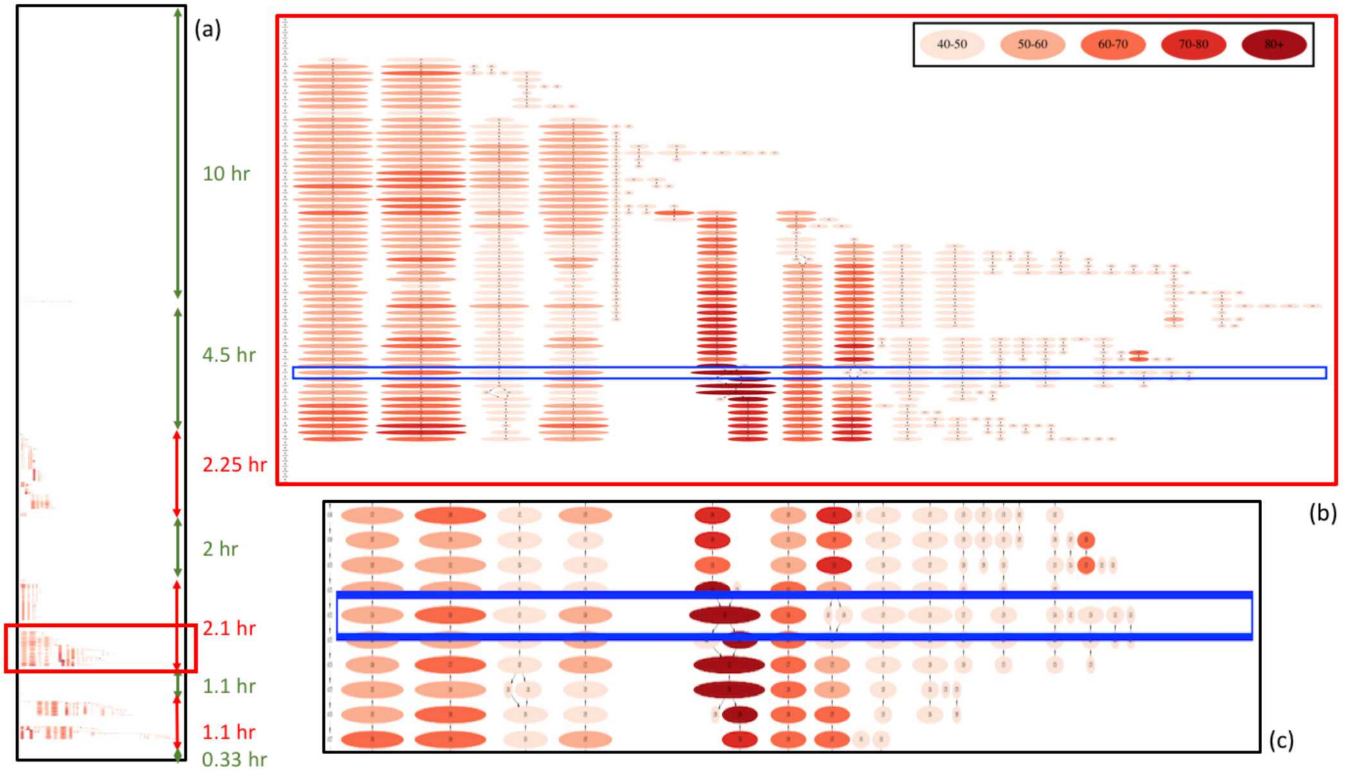


Fig. 9. Congestion evolution is more clearly seen in representations that favor time over location and is represented more compactly. P_{Ts} for credit stalls in X+ are shown; minimum value for inclusion in the figure is 40 percent. Time increases top to bottom at one minute intervals. Features are sized by extent (in routers) and colored by value (legend is boxed in black in (b)). Connections indicate persistence of congestion in routers with similar coordinates; all other location information is deliberately lost. (a) Full day of production data. (b) Expanded feature graph of congestion occurrence marked in red from (a). Blue boxed timeslice corresponds to the 3D torus representation in Figure 10. Times surrounding this time slice are expanded in (c) to enable examination of some feature connectivity.

This type of graph can be used to qualitatively assess if congestion spreads and intensifies – for instance, the increasing number of regions with time here indicates that congestion is spreading with smaller intensity (40-50) and in smaller sized features.

This representation allows the capture of evolution in a single figure, unlike the literal topology representation in Figure 10 [6] which shows the slice in time marked by the blue box in the feature graph. The feature graph time slice, and surrounding times, have been expanded in Figure 9 (c), in the lower right and outlined in black. Here, the darkest feature of size 8 represents the circled values in the wrap around in X at Y=23 and Z=16. This time slice was chosen since it exhibits the highest value of P_{Ts}, ~ 85, over the entire day associated with this data. Additionally, some feature connectivity through time, including splits and merges can be seen.

All features terminate at the same time after over 1 hour of continuous congestion when a series of *Congestion Protection Events* occur. This is a Cray-provided software mechanism [12] which seeks to alleviate congestion by throttling injection from all NICS such that the total injection bandwidth is less than that which can be handled by a single node. Because of the drastic nature of this approach, this event is triggered infrequently, based on certain Cray monitored traffic and stall quantities. Here we see that the network experienced significant congestion for at least an hour before the throttling event occurred.

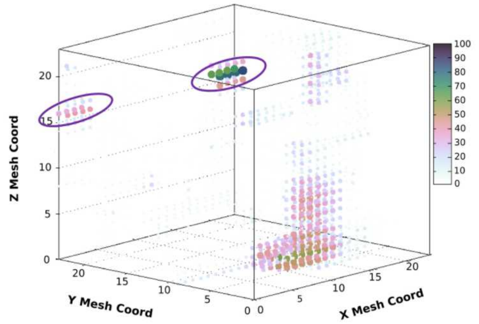


Fig. 10. 3D torus representation of the timeslice in the Blue box in Figure 9 (from [6]). Within the circles, the 8 values at Y=23 and Z=16 correspond to the highest value feature in that box. Note that there is no minimum value for inclusion in this figure. Over 60 such images would need to be viewed to assess the congestion evolution in Figure 9(b) and 1,440 images for Figure 9(a).

VII. Related Work

There are a variety of approaches in analyzing large-scale network congestion. Of the works that use network counters (as opposed to modeling, simulation, or indirect measures such as an application's messaging rate), most utilize counters that are accessible only from within an application's allocation (e.g., [2], [13], [14], [15], [16]), due to the complexities and access issues in collecting global synchronized data. These works cannot provide the complete network analysis we present here.

Previous works of the authors utilize global network counter data in Cray systems (e.g., [6], [17], [18]), but these do not include a longer-range assessment of link-level congestion, as is provided here.

Any number of works represent networks as graphs with vertices connected by edges representing communication. One work [19] used a model of network traffic rates and studied congestion control using graphs in which the vertices were congested links and the edges were the subset of links that data from each source traverses. A topology-centric visualization for a 3D torus, with reduced dimensionality views, was developed by Landge et al [20]. References within include visualizations based on communication traces and patterns, application topologies, etc. Some associated authors developed a topology-centric visualization for a 5D torus, with reduced dimensionality views [10]. None of these are the type of feature graph that we are considering here. Our graph representation is inspired by the feature-based analysis and feature graph work of Koegler et al. (e.g. [21]), which was without specific application to network feature evolution.

VIII. Conclusion

In this paper we have presented a variety of analysis and visualization techniques for understanding congestion in a Supercomputer's High Speed Network (HSN) from a link level perspective. We explored its application and I/O driven origins, and its effects on application performance. We have provided data and application use cases taken from NCSA's large scale (27,648) node Cray XE/XK Blue Waters system. We have described some of the sampling fidelity based limits to understanding imposed by the current data sampling period of 60 seconds.

While the example data and effects were specific to the Cray XE/XK platform, configured in a 3D Torus topology, and the Blue Waters system in particular, it is important to note that the analysis and visualization techniques presented are generally applicable to any system for which link level data on HSN bandwidth and congestion related metrics as well as node level I/O read and write metrics can be made available in a time synchronized fashion.

The techniques presented use link-level statistics to :

- estimate the relationship between P_{Ts} values and application impact
- understand and characterize the distribution of congestion and its relationship with traffic patterns. E.g., such an analysis helped to find the relationship between LNet (I/O) nodes and link-congestion behaviors, i.e., we did not observe any correlation between LNet (I/O) nodes and link congestion in our dataset
- develop a time-centric feature-graph-based visualization for congestion evolution

As follow-on work we plan to increase the data collection fidelity to try to identify at what level we can discriminate between network bursts, and corresponding congestion (if any) associated with file I/O and the normal application communication driven congestion.

Acknowledgment

We thank Larry Kaplan (Cray) and William Kramer, Michael Showerman, Gregory Bauer, and Jeremy Enos (NCSA) for providing raw data and many insightful conversations.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Award Number 2015-02674.

This work is partially supported by NSF CNS 15-13051. This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Application.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

[1] R. Alverson, D. Roweth, and L. Kaplan, "The Gemini system interconnect," in *2010 18th IEEE Symposium on High Performance Interconnects*. IEEE, 2010, pp. 83–87.

[2] K. Pedretti, C. Vaughan, R. Barrett, K. Devine, and S. Hemmert, "Using the Cray Gemini Performance Counters," in *Proc. Cray User's Group*, 2013.

[3] W. J. Dally and B. P. Towles, *Principles and practices of interconnection networks*. Elsevier, 2004.

[4] G. Almasi, "PGAS (partitioned global address space) languages," in *Encyclopedia of Parallel Computing*. Springer, 2011, pp. 1539–1545.

[5] R. Barriuso and A. Knies, "SHMEM users guide for C," Technical report, Cray Research Inc, Tech. Rep., 1994.

[6] A. Agelastos, B. Allan, J. Brandt, P. Cassella, J. Enos, J. Fullop, A. Gentile, S. Monk, N. Naksinehaboon, J. Ogden, M. Rajan, M. Showerman, J. Stevenson, N. Taerat, and T. Tucker, "Lightweight Distributed Metric Service: A Scalable Infrastructure for Continuous Monitoring of Large Scale Computing Systems and Applications," in *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014, pp. 154–165.

[7] Cray Inc., "Managing System Software for the Cray Linux Environment," Cray Doc S-2393-5202axx, 2014.

[8] National Center for Supercomputing Applications, "SPP-2017 Benchmark Codes and Inputs," https://bluewaters.ncsa.illinois.edu/spp-benchmarks.

[9] "Charm++ MiniApps ," http://charmplusplus.org/benchmarks/#amr.

[10] C. McCarthy, K. Isaacs, A. Bhatele, P. Bremer, and B. Hamann, "Visualizing the five-dimensional torus network of the ibm blue gene/q," in *Proc. 1st Workshop on Visual Performance Analysis (VPA)*, 2014.

[11] J. Ellson, E. Gansner, L. Koutsofios, S. North, G. Woodhull, S. Description, and L. Technologies, "Graphviz open source graph drawing tools," in *Lecture Notes in Computer Science*. Springer-Verlag, 2001, pp. 483–484.

[12] Cray Inc., "Managing Network Congestion in Cray XE Systems," Cray Doc S-0034-3101a Cray Private, 2010.

[13] M. Deveci, S. Rajamanickam, V. Leung, K. Pedretti, S. Olivier, D. Bunde, U. V. Catalyurek, and K. Devine, "Exploiting Geometric Partitioning in Task Mapping for Parallel Computers," in *Proc. 28th Int'l IEEE Parallel and Distributed Processing Symposium*, 2014.

[14] T. Groves, Y. Gu, and N. Wright, "Understanding Performance Variability on the Aries Dragonfly Network," in *IEEE Int'l Conf. on Cluster Computing*, 2017.

[15] R. Grant, K. Pedretti, and A. Gentile, "Overtime: A tool for analyzing performance variation due to network interference," in *Proc. of the 3rd Workshop on Exascale MPI*, 2015.

[16] S. Smith, D. Lowenthal, A. Bhatele, J. Thiagarajan, P. Bremer, and Y. Livnat, "Analyzing inter-job contention in dragonfly networks," 2016. [Online]. Available: https://www2.cs.arizona.edu/~smiths949/

[17] J. Brandt, E. Froese, A. Gentile, L. Kaplan, B. Allan, and E. Walsh, "Network Performance Counter Monitoring and Analysis on the Cray XC Platform," in *Proc. Cray User's Group*, 2016.

[18] J. Brandt, K. Devine, and A. Gentile, “Infrastructure for In Situ System Monitoring and Application Data Analysis,” in *Proc. Wrk. on In Situ Infrastructures for Enabling Extreme-scale Analysis and Viz.*, 2015.

[19] D. Hobson-Garcia and T. Hayakawa, “Using congestion graphs to analyze the stability of network congestion control,” in *2009 International Conference on Networking, Sensing and Control*, March 2009, pp. 559–564.

[20] A. G. Landge, J. A. Levine, A. Bhatele, K. E. Isaacs, T. Gamblin, M. Schulz, S. H. Langer, P. T. Bremer, and V. Pascucci, “Visualizing network traffic to understand the performance of massively parallel simulations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2467–2476, Dec 2012.

[21] W. Koegler and W. Kegelmeyer, “FCLib: A library for building data analysis and data discovery tools,” in *Advances in Intelligent Data Analysis VI. IDA 2005. Lecture Notes in Computer Science*, A. Famili, J. Kok, J. Pena, A. Siebes, and A. Feelders, Eds. Springer, 2005, vol. 3646.