HOLISTIC MONITORING AND DIAGNOSIS OF LARGE-SCALE APPLICATION PERFORMANCE DEGRADATION

Allocation: Exploratory/50 Knh
PI: Ravishankar Iyer[1]
Co-PI: Zbigniew Kalbarczyk[1]
Collaborators: Saurabh Jha[1], Benjamin Lim Wen Shih[1]

[1]University of Illinois at Urbana-Champaign

EXECUTIVE SUMMARY

Significant application performance variation has been attributed to "hot-spots" in the high-performance interconnect network, or localized congestion regions arising from applications contending for the same resources within a computer system. Such attribution has largely relied on limited inferential, simulated, or theoretical data. We present a methodology, implemented as a tool, to provide congestion characterizations at runtime for systems with large-scale interconnect networks and to inform diagnostic investigations. We studied Blue Waters high-speed network congestion data to characterize and diagnose likely congestion causes in applications. Our findings include:

- Continuous presence of highly congested links in the network. From our data, in 95% of the operation time we observed congestion regions with a minimum of 20 links and a maximum of 6,904 links. The average congestion duration is 16 minutes and the 95th percentile is 16 hours.
- Limited efficacy of default congestion-mitigation mechanisms. On average, congestion mitigation mechanisms trigger every seven hours but fail to detect 61% of the high-congestion events.

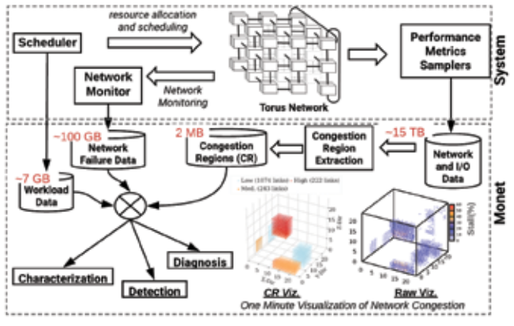

Figure 1: Characterization and Diagnosis workflow for interconnection-networks. The data set size highlighted in red above each block represents the size of the data set collected from Blue Waters.

RESEARCH CHALLENGE

Significant application performance variation has been attributed to network congestion in localized "hot spots" that arise from application contention. Determination of congestion-related performance impact in large-scale HPC production interconnect networks has been largely inferential, based on messaging rates or counters measured within an instrumented application or benchmark. Such measurements may not expose the hot spots, particularly for architectures where an application may be affected by congestion in components not directly accessible from its resource assignments (e.g., the congestion may occur in network components not under the control, nor visible to, the application). Simulations and theoretical models are limited in dynamic detail and often present "steady state" results that do not enable early detection of congestion or analysis of its evolution.

Production characterizations of congestion manifesting as hot spots (as opposed to fully congested, i.e., underprovisioned, networks) can be difficult, since they require systemwide, coherent data on the state of network performance counters at each component that may be limited in the amount of information exposed that can be used for pinpointing and attributing the root cause of congestion. In addition, data must be collected at fidelities necessary to capture the relevant phenomena. Such characterizations can, however, inform designs and acquisition decisions. Runtime evaluations that identify localized areas of congestion and assess congestion duration can be used to trigger Congestion Effect Mitigating Responses (CEMRs), such as resource scheduling, placement decisions, and dynamic application reconfiguration.

In this work, we present a methodology for characterizing congestion in large-scale high-speed 3D Torus networks.

METHODS & CODES

Data Sources. We have used five months of production monitoring data from Blue Waters. We used system-generated network resiliency logs (via netwatch logs), Light-weight Distributed Metric Service (LDMS) data, and generated network performance metrics and Blue Waters application logs. The sizes of the LDMS logs, netwatch logs, and application logs are about 15 TB, 100 GB, and 7 GB, respectively.

Methodology and Tools. We developed *Monet*, a generic framework for supporting congestion characterization and diagnosis in HPC systems. Fig. 1 shows the data collection and analysis pipeline of the *Monet* framework in the context of Blue Waters. *Monet* aggregates and analyzes the network and I/O, resilience, and workload data. The network stall counters are used in the extraction of congestion regions. A congestion region (CR) is a bounding (cuboid) box over all the links that (a) are close to one another (e.g., within a hop distance of three), and (b) have similar stall values. The identified congestion regions are then combined with other data sets (workload data, network failure data, and network performance data) to enable detection, diagnosis, and characterization of network congestion.

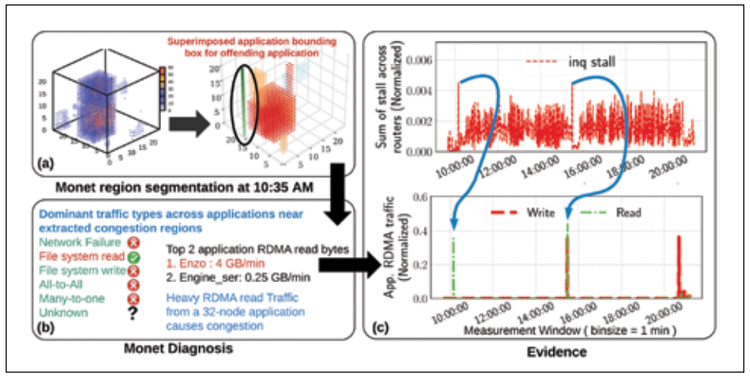

Figure 2: Detection and Diagnosis methodology applied to real scenario: (a) extracting congestion regions (CR), (b) finding anomalous application traffic pattern, and (c) generating evidence.

RESULTS & IMPACT

In this section, we show an example use case in which our analysis methodologies and framework, *Monet*, were used to detect and diagnose the congestion for an example scenario obtained from real data for which the ground truth of the cause was available.

Step 1: Extraction of CR. Fig. 2 shows that our analysis indicated widespread high-level congestion across the system (see the left graph in Fig. 2a). An in-depth analysis of the raw data resulted in identification/detection of congestion regions (see the top-right graph Fig. 2a).

Step 2: Congestion diagnosis. First, the tool correlates the CR-data with application-related network traffic (for all applications that overlapped or were near the congestion regions) and network information to generate candidate factors that may have led to congestion. In this example, there were no failures; hence, this analysis generated only application-related candidate factors A_{CR_i}. Next, we identify anomalous application traffic characteristics by using a median-based outlier detection algorithm.

In our example, as indicated in Fig. 2b, the offending application was Enzo running on 32 nodes allocated along the "Z" direction at location (X, Y, Z) = (0,16,16) (indicated by a black circle in Fig. 2a). At the time of detection, Enzo was reading from the file system at an average rate of 4 GB/min (averaged over the past 30 minutes and with a peak rate of over 70 GB/min), which was 16 times greater than the next-highest rate of read traffic by any other application in that time-window. The tool identified RDMA read bytes/min of the Enzo application as the outlier feature. Hence, Enzo was marked as the anomalous factor that led to the congestion.

Our tool generates evidence for system managers and users by producing timeseries data and statistical distributions of stall and traffic characteristics for the implicated application. The two peaks (marked) in this top plot correspond to the increase in read bytes (normalized to total read bytes during the application run) shown in the bottom plot. Although in this example scenario the Cray congestion mitigation mechanism was triggered, it was not successful in alleviating the network congestion, in part because the congestion was due to I/O traffic rather than message traffic. Instead, the CR size grew over time, impacting several applications.

WHY BLUE WATERS

Blue Waters is one of the few open-science capacity systems that provides a testbed for scaling computations to tens or hundreds of thousands of cores on CPUs and GPUs. It also enables the study of failures and degradations of applications in production petascale systems with its unique mix of XE6 and XK7 nodes. This allows us to understand the performance–fault-tolerance continuum in HPC systems by enabling the investigation of application-level designs for mixed CPU and GPU node systems, and fault isolation in system components to mitigate failures at the application level.

PUBLICATIONS & DATA SETS

Jha, S., et al., Holistic Measurement Driven System Assessment. *Proceedings of the Workshop on Monitoring and Analysis for High Performance Computing Systems Plus Applications* (Honolulu, Hawaii, September 2017).

Jha, S., et al., Characterizing Supercomputer Traffic Networks Through Link-Level Analysis. Submitted (2018).

Jha, S., et al., Characterizing and Diagnosing Congestion in Large-Scale Networks. Submitted (2018).