Towards a Real-Time Classification Mechanism for the Causes of Data Loss

Phillip M. Dickens
Department of Computer Science
University of Maine

ABSTRACT

Given the critical nature of communications in computational Grids it is important to develop efficient, intelligent, and adaptive communication mechanisms. An important milestone on that path is the development of classification mechanisms that can distinguish between the many factors that can cause data loss in cluster and Grid environments. In this paper, we describe our work in developing such a mechanism and its integration into a high-performance communication system for computational Grids. The classification algorithms are based on the analysis of what may be termed packet-loss-signatures that describe the patters of packet loss in the current transmission window. We present experimental data showing that the classification system can execute as part of the communication system with reasonable overhead, and that it can accurately distinguish between causes of packet loss at loss rates as low as 0.03%.

Introduction

Computational Grids create large-scale distributed systems by connecting geographically distributed computational and data-storage facilities via high-performance networks. Such systems, which can harness and bring to bear tremendous computational resources on a single large-scale problem, are becoming an increasingly important component of the national computational infrastructure. At the heart of such systems is the high-performance communication infrastructure that allows the geographically distributed computational elements to function as a single (and tightly-coupled) computational platform. Given the importance of Grid technologies to the scientific community, research projects aimed at making the communication system more efficient, intelligent, and adaptive are both timely and critical.

An important milestone on the path to such next-generation communication systems is the development of a classification mechanism that can distinguish between the many factors that can cause data loss in current cluster/Grid environments. The idea is to use the classification mechanism to respond to data loss in a way that is appropriate for the particular set of system dynamics responsible for such loss. Given such a capability, the communication system can take full advantage of the underlying bandwidth when system conditions permit, can back off in response to observed (or predicted) contention within the network, and can accurately distinguish between these two situations.

This research is addressing the issue of identifying the root cause(s) of data loss as observed by a high-performance data transfer system during the course of its execution. The approach we are pursuing is to analyze what may be termed *packet-loss signatures,* which show the distribution (or pattern) of those packets that successfully traversed the end-to-end transmission path and those that did not. These signatures are collected by the receiver and delivered to the sender upon request. Thus the packet-loss signatures are essentially large selective-acknowledgment packets, and are so named based on our belief (with support from previous research results [14, 15]) that different classes of error mechanisms have different "signatures". We are applying complexity theory to the problem of learning the underlying structure (or lack thereof) of these signatures, and studying the relationship between such underlying structure and the system conditions responsible for its generation. Our research has shown that complexity measures capture quite well the underlying system dynamics, and that understanding such dynamics provides significant insight into the cause(s) of observed data loss.

Currently, the packet-loss signatures are collected and stored during the data transfer but are analyzed offline. This research addresses the issue of integrating the classification mechanism into an existing high-performance data transfer system, and the calculation of complexity values as the data transfer is progressing. The longer-term goal of this work is to use, in a highly adaptive and efficient data transfer system, information related to the root cause(s) of data

loss. However, this paper focuses on techniques to build and integrate such classifiers; adaptations based on this knowledge will be the focus of forthcoming papers.

The testbed for this research is FOBS[1]: a high-performance data transfer system for computational Grids developed by the author[2][9-12, 25]. FOBS is a UDP-based data transfer system that provides reliability through a selective-acknowledgment and retransmission mechanism. As noted above, it is precisely the information contained within the selective-acknowledgment packets that is collected and analyzed by our classification mechanism.

Three important factors, whose combination is unique among high-performance data transfer mechanisms for computational Grids, make FOBS an excellent testbed for this research. First, FOBS is an application-level protocol. Thus the congestion-control algorithms can collect, synthesize, and leverage information from a higher-level view than is possible when operating at the kernel level. Second, the complexity measures can be obtained as a function of a *constant* sending rate. Thus the values of the variables collected are (largely) unaffected by the behavior of the algorithm itself. Third, FOBS is structured as a feedback control system. Thus the external data (e.g., the complexity measures) can be analyzed at each control point, and this data can be used to determine the duration of the next control interval and the rate at which data will be placed onto the network during this interval. We do not discuss further the design, implementation, or performance of FOBS here. The interested reader is directed to [11-13] for detailed discussions on these issues.

In this paper, we focus on distinguishing between contention for network resources and contention for CPU resources. This distinction is important for two reasons. First, contention for CPU cycles can be a major contributor to packet loss in UDP-based protocols such as FOBS. This happens, for example, when the receiver's socket-buffer becomes full, additional data bound for the receiver arrives at the host, and the receiver is switched out and thus unavailable to pull such packets off of the network. To illustrate this issue, consider a data transfer with a sending rate of one gigabit per second and a packet size of 1024 bytes. Given this scenario, a packet will arrive at the receiving host every 7.9 micro-seconds, which is approximately the amount of time required to perform a context switch on the TeraGrid systems [1] used in this research (as measured by Lmbench [19]).

The second reason this distinction is important is that data loss resulting from CPU contention is completely outside of the network domain and should not be interpreted as growing network condition. This opens the door for new, less aggressive responses, for this class of data loss.

It is important to point out that we used contention at the NIC as a proxy for network contention for two reasons: First, while it was relatively easy to create contention at the NIC level, it was extremely difficult to create contention within the 40 gigabit per second networks that connect the facilities on the TeraGrid. Second, there is little difference in the complexity measures obtained when data loss is caused by contention at an intermediate router and by contention at a NIC.

This paper makes two important contributions. First, it demonstrates a simple classification mechanism that is quite powerful in its ability to distinguish between various causes of packet loss. Second, and to the best of our knowledge, this represents the first time a classification mechanism has been integrated into a data transfer system and executed dynamically. This paper should be of interest to a large segment of the Grid community given the interest in and importance of exploring new approaches by which data transfers can be made more intelligent and efficient.

The rest of the paper is organized as follows. In Section 2, we discuss related work. In Section 3, we discuss the complexity analysis used in our research and show how such techniques can be applied to the packet-loss signatures. In Section 4, we describe the implementation of the classification mechanism and its integration into FOBS. In Section 5, we discuss the experimental design, and present the results of these experiments in Section 6. In Section 7, we provide our conclusions and outline future work.

2 Related Work

The issue of distinguishing between categories of losses has received significant attention within the context of TCP for hybrid wired/wireless networks (e.g., [3-6, 18, 24]). The idea is to distinguish between losses caused by network congestion and losses caused by errors in the

[1] Fast Object-Based data transfer System

[2] In collaboration with D.W. Gropp (Argonne National Laboratory).

wireless link, and to trigger TCP's aggressive congestion control mechanisms only in the case of congestion-induced losses. This ability to classify the root cause of data loss, and to respond accordingly, has been shown to improve the performance of TCP in this network environment [3, 18, 23]. These classification schemes are based largely on simple statistics on observed round-trip times, observed throughput, or the inter-arrival time between ACK packets [5, 7, 18]. Debate remains, however as to how well techniques based on such simple statistics can classify loss [18]. Another approach being pursued is the use of Hidden Markov Models where the states are characterized by the mean and standard deviation of the distribution of round-trip times [18]. Hidden Markov Models have also been used to model network channel losses and make inferences about the state of the channel [20].

Our research has similar goals, although we are developing a finer-grained classification system to distinguish between contention at the NIC, contention in the network, and contention for CPU resources. Also, we believe that complexity measures may prove to be a more robust classifier than (for example) statistics on round-trip times and could be substituted for such statistics within the mathematical frameworks established in these related works. Similar to the projects discussed above, we separate the issue of classification of root cause(s) of data loss from the issue of implementing responses based on such knowledge.

Research into other application-level alternatives to TCP is also related (e.g., [2, 21, 22]). However, none of these projects attempt to determine the root cause(s) of observed packet loss that is a major focus of our research.

3 Diagnostic Methodology

The packet-loss signatures can be analyzed as time series data with the objective of identifying diagnostics that may be used to characterize causes of packet loss. A desirable attribute of a diagnostic is that it can describe the dynamical structure of the time series. The approach we are taking is the application of *symbolic dynamics* techniques, which have been developed by the nonlinear dynamics community and are highly appropriate for time series of discrete data. This approach to classifying causes of packet loss works because of the differing timescales over which such losses occur. For example, packet loss due primarily to network-based causes such as router contention or contention at the NIC is likely to show temporal structure over a wide variety of timescales reaching down to the spacing between packets. A platform-based cause such as CPU contention at the host upon which the data receiver is executing will more likely be associated with a narrower range of longer timescales (e.g., the size of the time slice allocated to the receiver in a time-sharing system).

In symbolic dynamics [17], the packet-loss signature is a sequence of symbols drawn from a finite discrete set, which in our case is two symbols: 1 and 0. One diagnostic that quantifies the amount of structure in the sequence is *complexity*. There are numerous ways to quantify complexity. In this discussion, we have chosen the hierarchical approach of d'Alessandro and Politi [8], which has been applied with success to quantify the complexity and predictability of time series of hourly precipitation data [16].

The approach of d'Alessandro and Politi is to view the stream of 1s and 0s as a language and focus on subsequences (or *words*) of length **n** in the limit of increasing values of **n** (i.e., increasing word length). First-order complexity, denoted by $\mathbf{C^1}$, is a measure of the richness of the language's vocabulary and represents the asymptotic growth rate of the number of *admissible words* of fixed length **n** occurring within the string as **n** becomes large. The number of admissible words of length **n**, denoted by **Na(n)**, is simply a count of the number of distinct words of length **n** found in the given sequence. For example, the string **0010100** has **Na(1)** = 2 (0,1), **Na(2)** = 3 (00,01,10), **Na(3)** = 4 (001, 010, 101, 100). The *first-order complexity* (C^1) is defined as

$$C^1 = \lim_{n->\infty} (\log 2\ Na(n)) / n \qquad (1)$$

The first-order complexity metric characterizes the level of randomness or periodicity in a string of symbols. A string consisting of only one symbol will have one admissible word for each value of **n**, and will thus have a value of $C^1=0$. A purely random string will, in the limit, have a value of $C^1=1$. A string that is comprised of a periodic sequence, or one comprising only a few periodic sequences, will tend to have low values of C^1.

As noted, a hierarchy of complexity values is defined in [8]. The next level of the hierarchy is

a quantity termed C^2 that captures the fact that random strings are of lower complexity than strings that have rules governing their creation. We do not discuss this quantity here because we have not yet integrated it into our classification mechanism.

4 Implementation of Classifiers

The classification mechanism uses a sliding window of length **n** to search for all of the admissible words of length **n** in the signature. If, for example, it is searching for words of length **n = 3**, then the first window would cover symbols 0-3, the second window would cover symbols 1-4, and so forth. Recall that the symbols are either 1 or 0, and represent the received/not-received status of each packet in the transmission window. As each word is examined, an attempt is made to insert it into a binary tree whose branches equal to either 1 or 0. Inserting the word into the tree consists of following the path of the symbol string through the tree until either (1), a branch in the path is not present or (2), the end of the symbol string is reached. If a branch does not exist, it is added to the tree and the traversal continues. In such a case, the word has not been previously encountered and the number of admissible words (for the current word size) is incremented. Similarly, if the complete path from the root of the tree to the end of the symbol string already exists, then the word has been previously encountered and the count of admissible words is unchanged. Given the count of admissible words of length **n**, it is straight-forward to calculate the complexity value for that word length.

The integration of the classification mechanism into FOBS is made possible (or at least reasonably straight-forward) by the fact that FOBS is a synchronous communication protocol. That is, the sender and receiver synchronize after all of the data in the current transmission window has been successfully received. The receiver then writes the data to disk (assuming it is a file transfer) while the sender reads from disk the data to be sent in the next transmission window. The complexity values are also computed during this time and provided to the controller. The controller, in turn, uses the complexity values (and other variables discussed below) to determine the size of the next transmission window and the rate at which the data will be sent during the window.

5 Experimental Design

A set of experiments were conducted to answer two fundamental questions: First, are the statistics associated with each type of data loss different enough to allow for classification based on those statistics? Second, what is the cost of performing the classification dynamically?

All experiments were conducted on the TeraGrid [1]: a high-performance computational Grid that connects various supercomputing facilities via networks capable of operating at speeds up to 40 gigabits per second. The two facilities used in these experiments were the Center for Advanced Computing Research (CACR, located at the California Institute of Technology), and the National Center for Supercomputing Applications (NCSA, located at the University of Illinois, Urbana). The host platforms at both facilities were IA-64 Linux clusters where each compute node consisted of dual Intel Itanium2 processors. The compute nodes were 1.3 GHz at CACR and 1.5 GHz at NCSA. The operating system at both facilities was Linux 2.4.21-SMP. Each compute node had a gigabit Ethernet connection to the TeraGrid network.

The experiments were designed to capture a large set of complexity measures under known conditions. In one set of experiments, the data receiver executed on a dedicated processor within CACR, and additional compute-bound processes were spawned on this same processor to create CPU contention. As the number of additional processes was increased, the amount of time the data receiver was switched out similarly increased. Since the data receiver was not available to take packets off of the network during the times it was switched-out, there was a direct relationship between CPU load and the resulting packet loss rate. We were interested in analyzing the structure of the bitmaps as a function of both the root cause of data loss (i.e., contention for CPU or NIC resources) and the loss rate. We therefore varied the number of additional processes to obtain a wide range of loss rates.

To investigate loss patterns caused by contention for NIC resources, we initiated a second (background) data transfer. The data sender of the background transfer executed on a different node within NCSA, and the receiver executed on the second processor within the same node as the primary data receiver. Since both processors of a given node share the same NIC, we were able to generate contention at the NIC without causing contention for CPU cycles

with the two receivers. Initially, the combined sending rate was set to the maximum speed of the NIC (one gigabit per second), and contention for NIC resources was increased by increasing the sending rate of the background transfer. The packet loss experienced by both data transfers was a function of the combined sending rate, and this rate was also set to provide a wide range of loss rates.

In both sets of experiments, the complexity values were computed at each control point. Thus we were able to determine the cost of the classification based on the difference between the sending rate and the actual throughput. All experiments were performed late at night when there was little (if any) network contention. The sending rate was held constant at 800 megabits per second to accurately gauge the cost of the classification mechanism. Similar to the technique of loss pairs [18], we maintained a parallel data transfer (with the same send rate) that traversed the same network path except for the last hop. This approach was taken to determine the impact of contention within the network path as a cause of data loss. Similarly, we used hardware counters to track the state of the network internal to the data receiver and to track other CPU events such as process creation, paging, and context switches.

6 Experimental Results

Figure 1 shows the differences in complexity values as a function of the root cause of data loss. As can be seen, the complexity measures tend to diverge very quickly with increasing loss rates. This is very encouraging in terms of using complexity measures as a classifier for causes of data loss.

Figure 2 focuses on the complexity values associated with very low loss rates. As can be seen, the complexity measures are quite close until the loss rate reaches approximately 0.003. From this we conclude that complexity measures in and of themselves are not powerful enough to serve as a classifier at very low loss rates. Thus a focus of our current research is the development of other variables that can be used either instead of or in conjunction with complexity measures at low loss rates.

Figure 3 points out an interesting anomaly in the data where the complexity measure associated with CPU contention is significantly higher than any other such value. We examined the log of the hardware metrics collected during each transmission window to see if it could shed light on what caused such a high complexity value. It turned out that there were *twenty processes* created during that particular window resulting in approximately 6000 context switches. The purpose of the processes is unknown, but they were external to the data transfer. Given that there was no loss of data due to network contention, it is assumed that all of the extra CPU activity was responsible for the increased complexity. This shows that significant changes in CPU activity, as shown by the variables collected by the hardware counters, may also be quite helpful in ruling in or out various causes of packet loss.

The final metric of interest was the cost of dynamically analyzing the packet-loss signatures. As noted, the sending rate was a constant 800 megabits per second. The overall throughput was measured at 670 megabits per second. Thus the classification mechanism has an overhead of approximately 16%.

7 Conclusions and Future Research

In this paper, we have shown that complexity measures of packet-loss signatures can be highly effective as a classifier for causes of packet loss over a wide range of loss rates. Also, it was shown that the divergence of complexity measures, and thus the ability to discriminate between causes of packet loss, increases rapidly with increasing loss rates. However, for loss rates less than approximately 0.003, it was shown that complexity measures in and of themselves are not powerful enough to discriminate between causes of packet loss. Thus our current research efforts are focused on identifying other metrics or statistical models that can be effective at very low loss rates.

We also showed that it is quite feasible to integrate our classification mechanism with an existing high-performance data transfer system, and to compute the complexity values as the data transfer is progressing. While this computation represents an overhead of approximately 16%, it still represents a very good tradeoff when contention for CPU resources is a major source of data loss.

The ability to classify the temporal dynamics of packet loss behavior (as expressed by the packet-loss signatures) offers two significant advantages. First, such classification allows the control mechanisms to apply corrective actions based on the particular cause of packet loss. For example, the control mechanisms may be able to migrate the data receiver, rather than drastically

reducing the sending rate, when the root cause of packet loss is determined to be contention for CPU (rather than network) resources. Second, if the underlying dynamics has structure, it may be possible to construct simple predictors that allow the data transmitter to shape its behavior in such a way as to increase the probability that a sent packet is received successfully. These are enticing possibilities, and the exploration, evaluation, and integration of these techniques to the problem of large-scale data transfers represents the focus of our current research activities.

References

1. The TeraGrid Homepage http://www.http://www.teragrid.org
2. Allcock, W., Bester, J., Breshahan, J., Chervenak, A., Foster, I., Kesselman, C., Meder, S., Nefedova, V., Quesnel, D. and Tuecke, S. Secure, Efficient Data Transport and Replica Management for High-Performance Data_Intensive Computing. In the Proceedings of *IEEE Mass Storage Conference*, (2001).
3. Balakrishnan, S., Padmanabhan, V., Seshan, S. and Katz, R. A Comparison of Mechanisms for Improving TCP Performance Over Wireless Links. In *IEEE/ACM Transactions of Networking*, *5(6)*. Pages 756-769. 1997.
4. Balakrishnan, S., Seshan, S., Amir, E. and Katz, R. Improving TCP/IP performance over wireless networks. In the Proceedings of *ACM MOBICON*, (November 1995).
5. Barman, D. and Matta, I. Effectiveness of Loss Labeling in Improving TCP Performance in Wired/Wireless Networks. In the Proceedings of *ICNP'2002: The 10th IEEE International Conference on Network Protocols*, (Paris, France, November 2002).
6. Biaz, S. and Vaidya, N. Discriminating Congestion Losses from Wireless Losses using Inter-Arrival Times at the Receiver. In the Proceedings of *IEEE Symposium ASSET'99*, (Richardson, TX, March, 1999).
7. Biaz, S. and Vaidya, N. Performance of TCP Congestion Predictors as Loss Predictors, Texas A&M University, Department of Computer Science Technical Report 98-007, College Station, Texas.
8. D'Alessandro, G. and Politi, A. Hierarchical Approach to Complexity with Applications to Dynamical Systems. In *Physical Review Letters*, *64* (14). Pages 1609-1612. April, 1990.
9. Dickens, P. FOBS: A Lightweight Communication Protocol for Grid Computing. In the Proceedings of *Europar 2003*, (2003).
10. Dickens, P. A High Performance File Transfer Mechanism for Grid Computing. In the Proceedings of *The 2002 Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*. (Las Vegas, Nevada, 2002).
11. Dickens, P. and Gropp, B. An Evaluation of Object-Based Data Transfers Across High Performance High Delay Networks. In the Proceedings of *the 11th Conference on High Performance Distributed Computing*, (Edinburgh, Scotland, 2002).
12. Dickens, P., Gropp, B. and Woodward, P. High Performance Wide Area Data Transfers Over High Performance Networks. In the Proceedings of *The 2002 International Workshop on Performance Modeling, Evaluation, and Optimization of Parallel and Distributed Systems.*, (2002).
13. Dickens, P. and Kannan, V. Application-Level Congestion Control Mechanisms for Large Scale Data Transfers Across Computational Grids. In the Proceedings of *The International*

Conference on High Performance Distributed Computing and Applications, (2003).

14. Dickens, P. and Larson, J. Classifiers for Causes of Data Loss Using Packet-Loss Signatures. In the Proceedings of *IEEE Symposium on Cluster Computing and the Grid(ccGrid04)*, (2004).
15. Dickens, P., Larson, J. and Nicol, D. Diagnostics for Causes of Packet Loss in a High Performance Data Transfer System. In the Proceedings of *Proceedings of 2004 IPDPS Conference: the 18th International Parallel and Distributed Processing Symposium*, (Santa Fe, New Mexico, 2004).
16. Elsner, J. and Tsonis, A. Complexity and Predictability of Hourly Precipitation. In *Journal of the Atmospheric Sciences*, *50* ((3)). Pages 400-405. 1993.
17. Hao, B.-l. *Elermentary Symbolic Dynamics and Chaos in Dissipative Systems*. World Scientific, 1989.
18. Liu, J., Matta, I. and Crovella, M. End-to-End Inference of Loss Nature in a Hybrid Wired/Wireless Environment. In the Proceedings of *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'03)*, (Sophia-Antipolis, France, 2003).
19. The BitMover Homepage. http://www.bitmover.com/lmbench/
20. Salamatian, K. and Vaton, S. Hidden Markov Modeling for Network Communication Channels. In the Proceedings of *ACM SIGMETRICS 2001 / Performance 2001*, (Cambridge, Ma, June 2001).
21. Sivakumar, H., Bailey, S. and Grossman, R. PSockets: The Case for Application-level Network Striping for Data Intensive Applications using High Speed Wide Area Networks. In the Proceedings of *Super Computing 2000 (SC2000)*.
22. Sivakumar, H., Mazzucco, M., Zhang, Q. and Grossman, R. Simple Available Bandwidth Utilization Library for High Speed Wide Area Networks. In *Submitted to Journal of SuperComputing*.
23. Tsaoussids, V. and Matta, I. Open Issues on TCP for Mobile Computing. In *Journal of Wirelesss Communications and Mobile Computing- Special Issue on Reliable Transport Protocols for Mobile Computing*, *2(1)*. February, 2002.
24. Vaidya, N. and Biaz, S. Discriminating Congestion Losses from Wireless Losses Using Inter-Arrival Times at the Receiver. In the Proceedings of *IEEE Symposium ASSET'99*, (March, 1999).
25. Vinkat, R., Dickens, P. and Gropp, B. Efficient Communication Across the Internet in Wide-Area MPI. In the Proceedings of *The 2001 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*. (Las Vegas, Nevada, 2001).

Figures:

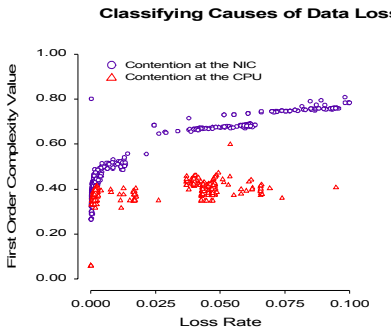

Figure 1. This figure shows the first-order complexity measures associated with NIC contention and CPU contention. The loss rate is varied from 0% to 10%.

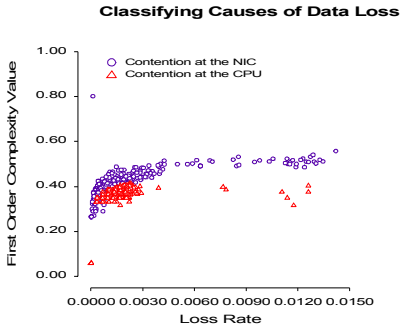

Figure 2. This figure shows the first-order complexity measures associated with NIC contention and CPU contention at very low loss rates. The loss rate is varied from 0% to 1.5%.

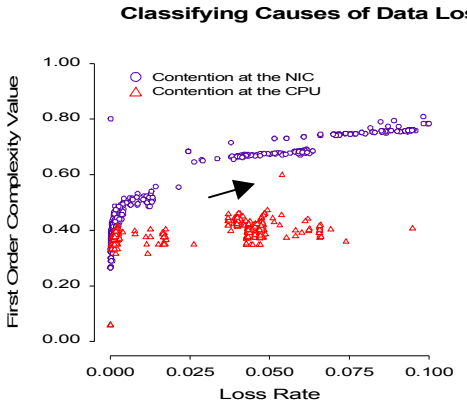

Figure 3. This figure points out a very high complexity value that is not associated with NIC contention.