Network Traffic Analysis With Query Driven Visualization
SC 2005 HPC Analytics Results

Kurt Stockinger[α], Kesheng Wu[α], Scott Campbell[σ], Stephen Lau[σ],
Mike Fisk[Ψ], Eugene Gavrilov[Ψ], Alex Kent[Ψ],
Christopher E. Davis[ΨΩ], Rick Olinger[Ω], Rob Young[Ω], Jim Prewett[Ω],
Paul Weber[Ψ], Thomas P. Caudell[ΩΨ], E. Wes Bethel[α], Steve Smith[ΨαΩ]

Lawrence Berkeley National Laboratory (LBNL)
[α]High Performance Computing Research Department (HPCRD/LBNL)
[σ]National Energy Research Sciences Center (NERSC/LBNL)
[Ψ]Los Alamos National Laboratory (LANL)
[Ω]University of New Mexico (UNM)

Our analytics challenge is is to identify, characterize, and visualize anomalous subsets of large collections of network connection data. We use a combination of HPC resources, advanced algorithms, and visualization techniques. To effectively and efficiently identify the salient portions of the data, we rely on a multi-stage workflow that includes data acquisition, summarization (feature extraction), novelty detection, and classification. Once these subsets of interest have been identified and automatically characterized, we use a state-of-the-art high-dimensional query system to extract data subsets for interactive visualization. Our approach is equally useful for other large-data analysis problems where it is more practical to identify interesting subsets of the data for visualization than to render all data elements. By reducing the size of the rendering workload, we enable highly interactive and useful visualizations. As a result of this work we were able to analyze six months worth of data interactively with response times two orders of magnitude more quickly than with conventional methods.

Motivating Example: Figure 1 shows an example of thousands of network connections (white lines) that one of our automatic clustering methods identifies as being correlated. Each of the shaded regions delimits a hyperbox for subsequent high dimensional queries. Since the entire dataset may represent billions of connections, it is critical to have good methods for defining and executing subset queries over multiple dimensions.
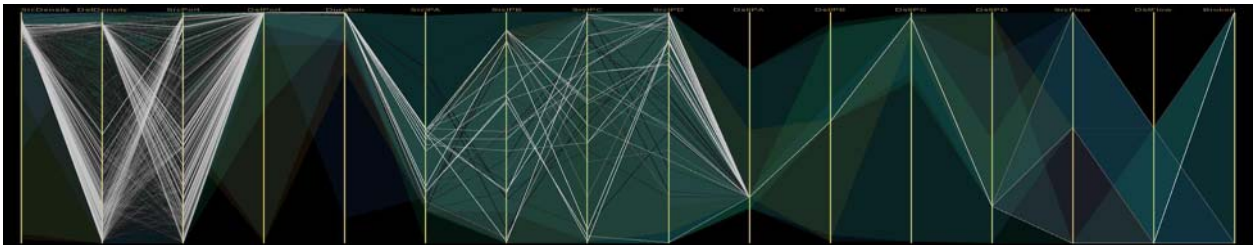

Figure 1. Parallel Coordinates View of Clusters of Network Traffic

Data: In the past year, on the order of 500 Terabytes (TB) crossed the boundary between the Internet and unclassified networks at LANL, NERSC, and LBNL. Data collection tools at each of these boundaries collect summary information of approximately 10 billion distinct connections, or 1 TB, per year. In addition, router-based information saved for every subnet internal to the LANL unclassified network totals 46 billion records and 2.5 TB per year, representing several Petabytes of network traffic. Post-processing of this data for analysis and indexing can increase the size several times. In the past, this data could only be analyzed as a whole by large batch processing jobs or in small segments, usually 6 to 24 hours at a time. The raw data consists of summary information on each session, and consists of start time, duration, protocol, source and destination byte counts, packet counts, IP addresses, port numbers, and flags describing completeness of the connection. Subsequently, some of these fields are further decomposed (octets of IP addresses, for example) and statistical properties derived on a per-host basis**.**

Data Collection and Management: Los Alamos uses a hybrid data collection, storage and retrieval system containing components for collecting flows from routers (cflowd), summarizing packet captures into flows (Nethead), extracting statistical features from flows (Hive), performing anomaly detection (Emaad), and maintaining accessibility of all data for ad-hoc query and forensic analysis (SMACQ, DiSARM, MySQL). We have implemented multiple techniques for storing and querying datasets of different sizes. Our System for Modular Analysis and Continuous Queries (SMACQ) allows ad-hoc, relational queries of un-indexed data. We also efficiently generate binary MySQL table files without requiring MySQL to parse the data itself, resulting in an off-the-shelf relational database system capable of keeping up with the extremely high burst data rates associated with network traffic. For even larger, historical datasets, we use our DiSARM system for B+-tree indexed access to flat files.

Lawrence Berkeley has developed a unique indexing, storage and retrieval system known as FastBit that uses extremely efficient compressed bitmap indices. FastBit has been used very effectively for managing huge collections of scientific data. For this challenge, we adapted this system, which has already integrated with the ROOT data analysis and graphing system developed at CERN, for the purpose of network data analysis. FastBit's performance has been shown to exceed that of similar systems for multi-dimensional queries, scaling linearly with size of the data set and sub linearly with the number of dimensions. With the help of the ROOT developers, we parallelized the index evaluation and data retrieval. This allowed us to increase ROOT-FastBit's performance by another order of magnitude.

Parallel Subselection with FastBit: We tested our system on 24 weeks of network traffic data consisting of 1.1 billion records of 25 fields each. The total size of raw data was 241 Gigabytes (GB), and the size of the compressed bitmap indices was 73GB (which is only 30% of the raw data). Note that the typical size of one of the most commonly used indices in database systems, the B+-tree, is often three to four times larger than the raw data. This underscores the storage efficiency of our compressed bitmap index implementation. All the measurements were performed on a 12-processor SGI Onyx system with a disk array capable of supporting 670 MB/s throughput.

Using FastBit, a typical three-dimensional query, such as "select IPS_B, IPS_C, IPS_D where IPS_B < 100 and IPS_C < 100 and IPS_D = 128" can be answered much more efficiently than other searching systems. On the test system using all 12 pocessors, FastBit uses an average of 22.8 seconds to answer the above query. In contrast, a system commonly used for analysis of large scientific datasets (with a known efficient configuration) requires 2,467 seconds to answer the same query on one processor. Assuming a perfect speedup, it would take 206 seconds on 12 processors. FastBit is 9 times faster in this case. If a typical database system were used, it would have to essentially read all records to answer the above query.

FastBit is more efficient than other searching system for a number of reasons. The compressed bitmap indices used in FastBit are compact. The compressed bitmap indices are also efficient for answering multiple attribute queries. On the particular test machine, FastBit additionally benefits from the fact that many indices can fit into the main memory.

The figure below shows the scaled parallel efficiency of FastBit on multiple processors. In most cases, we observe a parallel efficiency of about 80%. When attempting to use all 12 processors, we compete with the OS tasks and observe a parallel efficiency of about 60%.
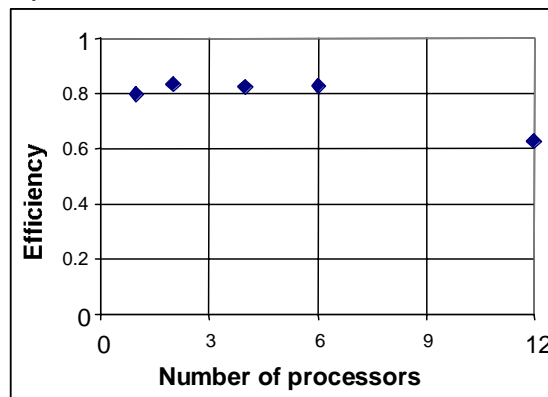

Figure 2: The scaled parallel efficiency of FastBit measured on a 12-processor SGI Onyx system.

Automatic Classification and Novelty Detection: Using machine-learning techniques, we are able to classify hosts and flows and to identify a much smaller subset of the overall data for further analysis. For example, by using an incremental clustering algorithm over 10 statistical features extracted by Hive over a month's worth of daily host values, we have generated 27 clusters. On subsequent days, we then use a *k*-means algorithm to assign each host to one of these clusters. Three of the clusters account for nearly 89% of the data. Expert analysis has verified that these clusters represent innocuous behavior in this feature space and can be excluded from our analysis. Our visual analysis therefore focuses on the remaining 11% of data. We have also implemented Decision Tree algorithms to train against this filtered data in order to develop a set of concise query terms necessary to reproduce the same sub-selection in efficient query engines.

UNM, working with LANL, has developed Adaptive Resonance Theory based neural networks to find clusters in the temporal domain as well as in feature space. Each cluster is described by a range in each of the dimensions of the raw data. By training these algorithms on "normal" or "abnormal" subsets of the whole data set, the resulting templates describe ranges in high dimensions or hyperboxes.

Los Alamos's Emaad system is an unsupervised detector of anomalous per-host behavior. Emaad uses efficient exponential functions to estimate the first and second moments of time-series data. By design, Emaad lends itself to measuring time-series data with periodic spikes and on/off behaviors, as well as constant behavior. Thus, it operates as efficiently as a simple exponential moving average anomaly detector, but with dramatically fewer false positives for these common patterns of network traffic. Emaad yields a prioritized list of hosts for further analysis. This list is used to automatically build a query that extracts additional contextual information in order to visualize the anomalous hosts' traffic in context.

Visualization Tools: We use ROOT's basic plotting capabilities to verify and review the results of the FastBit. ROOT provides a range of simple and effective histogram, point and surface plotting in 2 and 3 dimensions that we use for visual analysis of aspects of the network traffic.

The SpaceShield, HyperSpace Viewer, and HyperCube Viewer tools implemented in the Flatland/Flux environment were designed specifically to provide a first-person immersive experience in 3D scenes. Scene complexity must be maintained below tens of thousands of simultaneous objects to maintain smooth animation. The underlying visual idioms of these tools are that of a threatening hemispherical space over a defended annular space (SpaceShield) and of a hypercube, which is viewed as a 3D projection and can be rotated through the major axes or, if desired, rotated arbitrarily off-axis.

The two methods, ROOT's graphs and plots along with Flatland's immersive, metaphorical environments, complement each other well. The former is very utilitarian and familiar while the latter has the potential of improving cognitive coupling through intuitive familiarity and larger conceptual real estate.

Visual Analytics with ROOT-FastBit: The daily output from network Intrusion Detection systems typically identifies between tens and hundreds of potentially hostile IP addresses. Given the tremendous volume of connection data, answering a simple question such as 'have we ever seen this IP address before?' can take significant time and resources. As the dimensionality of queries increases, the time and effort required may grow beyond what would be acceptable for day to day use. In the following example, two IP addresses from the same class B address space are identified as hostile by routine traffic monitoring. By working backwards from the original data, it is possible to identify previously unknown hostile hosts by filtering on a combination of failed connections and the source network in the range of the suspect IP addresses, then plotting the last two octets of the connections as a 2D histogram.
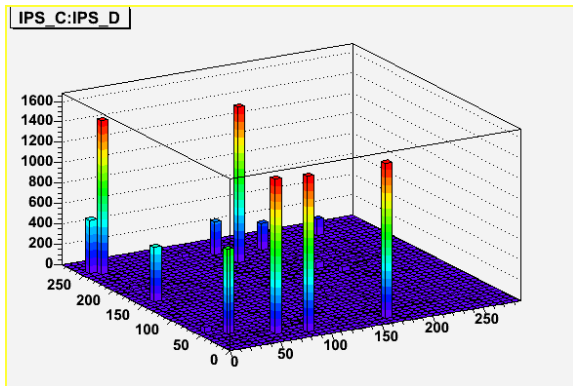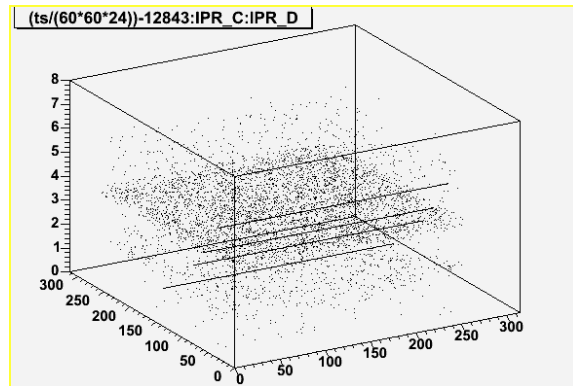
Figure 3. Suspicious Hosts

Figure 4. Scatterplot of Class B Network

In Figure 3 we see spikes identifying hosts that have engaged in this suspicious behavior. In order to associate this activity with a type of scanning, we can look at the distribution of destination hosts over a discrete period of time. In Figure 4 we see that there are several linear scans of individual subnets (the thin lines along the x-axis), and two entire class-B scans during this time period. This shows the behavior of the hosts identified in Figure 3.The most significant feature here is the immediate access to detailed connection data for long periods of time. This is a practical and useful result that may lend itself to FastBit being folded into the production environment at NERSC.

Immersive Network Traffic Monitoring and Analysis with Flatland/Flux Based Tools: A number of information visualization tools have been implemented in the Flatland/Flux environment developed at UNM and LANL. In Figure 5 we see the SpaceShield (on the left) and HyperBox Viewer (on the right).   In Figure 6 we see the HyperCube Viewer.
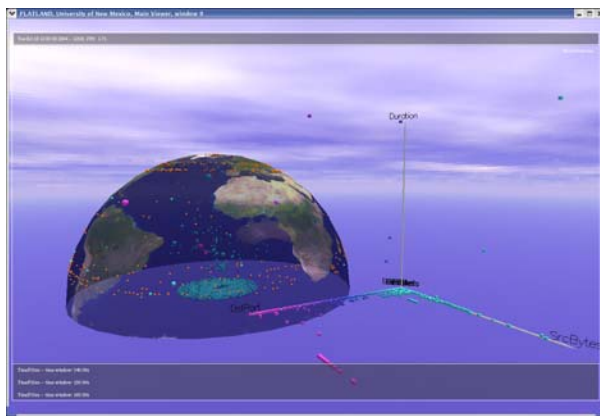

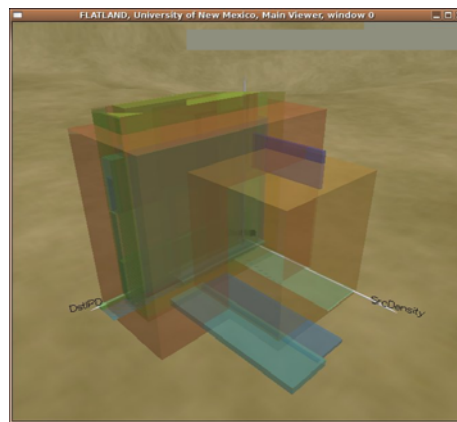Figure 5. SpaceShield and HyperSpace Viewer

Figure 6. HyperCube Viewer

The SpaceShield (Figure 5) shows external hosts either by their IP addresses or their locations in geographic coordinates. Internal hosts are distributed around a ring in the center and connections between them are shown as animated glyphs encoding port number, amount of data and duration of sessions. The playback rate can be adjusted to review long periods of time quickly or to analyze very short duration events. This is a particularly useful view for analyzing scans, denial of service attacks, and traffic from bad neighborhoods. This example captures the end of a very large IP scan, seen as many green glyphs at the center of the ring.

The HyperSpace Viewer allows the user to project the 11 dimensional data coordinates into 3 dimensions. This is a richly interactive tool for discovering correlations in collections of data between multiple dimensions.   The SpaceShield and HyperSpace Viewer are linked and show the same data in different views. The scan traffic from the Space Shield is shown near the axes and another unusual feature in pink is seen in the near ground.

The HyperCube Viewer works like the HyperSpace Viewer but was augmented to show the hyperbox templates from the ART clustering algorithm.   Each cube here is only one 3 dimensional facet of a much higher dimensional hyperbox which could be the basis for a high-dimensional query. Each hypercube encloses a set of connections that

are similar in some way. Expert analysts are experimenting with these clusters to try to identify an intuitive understanding of how these data are correlated.

These animated visualizations can best be understood by viewing the accompanying video.

Summary: Existing tools for analyzing network data are challenged by the size and complexity of large institutional networks. We have demonstrated a new ability to ingest network traffic from large networks and to interactively analyze, retrieve, and view that data. We have used advanced machine-learning techniques to automatically identify subsets of the data that are in need of further, interactive analysis and to categorize data for analysts. Using advanced query engines such as FastBit on HPC hardware allows near-interactive rate access (<60 second response time) to large network data sets. With efficient visualizations on the backend, the query results can be interactively manipulated and reviewed.