Network Traffic Analysis with Query-Driven Visualization

Kurt Stockinger, Kesheng Wu, Scott Campbell, Stephen Lau, E. Wes Bethel, Steve Smith, Brian Tierney, Eli Dart, Jason Lee

Mike Fisk, Eugene Gavrilov, Alex Kent, Christopher E. Davis, Paul Weber, Steve Smith

Rick Olinger, Rob Young, Jim Prewitt, Thomas P. Caudell

Introduction

- Network Traffic Analysis
  - Detect anomalous events in large collections of network connection data generated by software running on borders.
  - Anomalies include scans, compromised hosts, unauthorized use of resources, etc.
  - Use visual, interactive interfaces along with software tools that perform automatic feature detection.
- Query-Driven Visualization
  - Detection consists of finding data that match a given set of criteria.
  - Our approach provides a new capability: highly efficient technology to reduce the duty-cycle in knowledge discovery that is combined with analysis and visualization.
  - The combination increases the likelihood of discovering unexpected anomalies in network connection data.

Network Security Example

- Need to interactively explore millions of connection records to understand the nature of attacks.
- For example, say you notice that your site is being scanned from many addresses on the same subnet.
- You want to know if scanner is:
  - a single attacker
  - a coordinated distributed scan
  - a combination of both
- In general, a distributed scanner indicates a more sophisticated attacker, and something to be more worried about.
- Interactive analysis and visualization allows us to determine the type of attack

Our Results – Summary

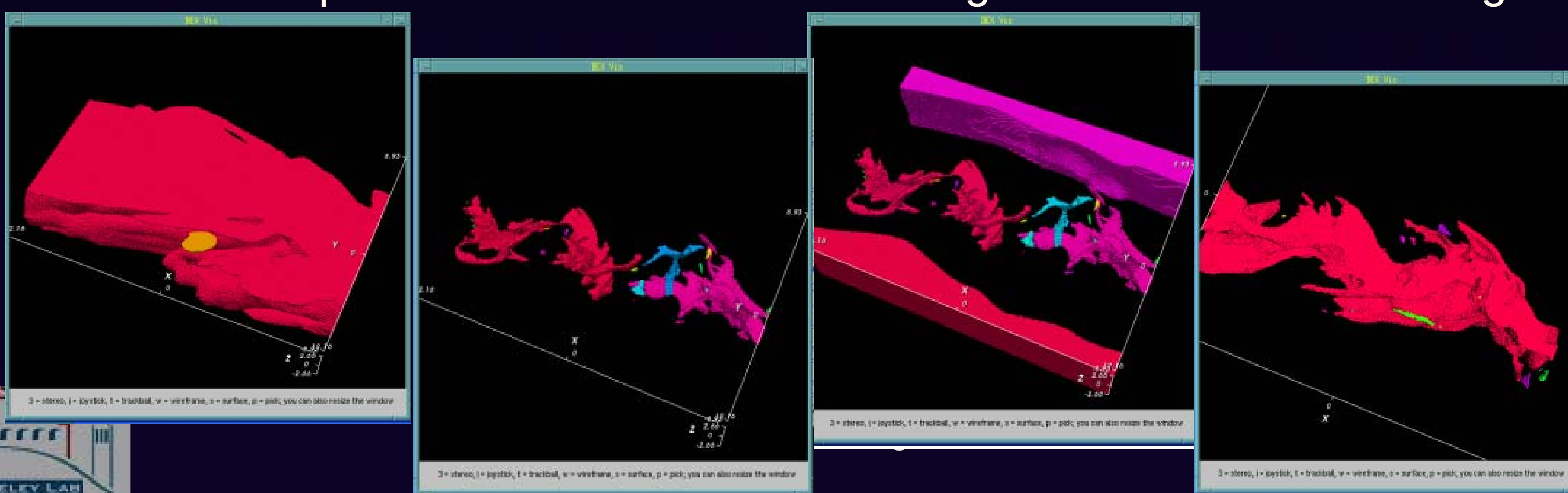
- New (unofficial) “speed and size” achievement.
- Size achievement:
  - Our entry processes 24 weeks’ worth of data (in post-entry runs, we process a full year’s worth of data).
  - In contrast, traditional traffic analysis tools can handle tens of hours’ worth of data.
- Speed achievement:
  - For our entry, a typical query is returned in only 22 seconds on a 12-way parallel system.
  - In contrast, a serial query from an alternate technology answers the same query in about 2200 seconds.

Significance of Results

- Performance result: size and speed.
  - Demonstrated one to two orders of magnitude faster.
  - Demonstrated one to two orders of magnitude larger.
- Impact:
  - Reduce duty-cycle in queries: the basis of discovery.
    - High impact in time-critical situations.
    - A viable set of solutions to address problems of growing data size and complexity.
  - New size and speed capabilities enable faster and broader knowledge discovery.
- Additional significance on following slides.

Robustness/Flexibility and Novelty

- Indexing and query technology is applicable to a wide range of uses – not just network traffic analysis.
  - High energy physics data analysis (CHEP 2004).
  - Combustion (below), astrophysics, …
  - Accelerating the visualization pipeline (IEEE Visualization 2005).
- Novelty
  - New capabilities result when combining best-of-breed technologies.



Performance and Scalability

- Serial performance: one order of magnitude faster.
  - Computational complexity a function of the number of items returned by the query, not the number of items searched.
- Parallel operation: another order of magnitude.
  - Our entry shows parallel execution across a 12-way system.
  - Parallel speedup efficiency of about 80%.
  - More in-depth scalability experiment presently underway

The Rest of the Presentation

- Network Traffic Analysis – Kurt Stockinger (LBNL)
  - What features did we detect in live network data and the analytic discourse to realize those discoveries.
- FastBit Indexing for efficient Data Queries – Kurt Stockinger (LBNL)
  - Finding data quickly relies on scientific data management technology for indexing and querying large data sources.
- Feature Identification and Visualization – Steve Smith (LANL and LBNL)
  - Feature identification: several automated tools help find features in large and complex data.
  - Visualization complements data management and analysis technologies and is the primary communication vehicle between software, data and human analysts.

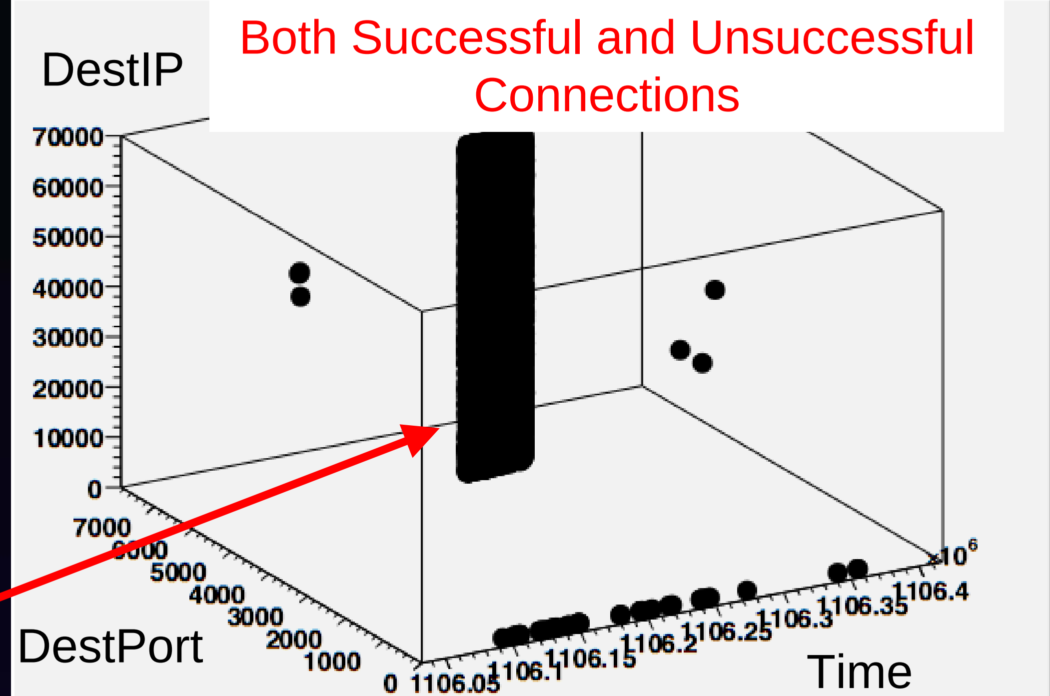
The Network Analysis Story

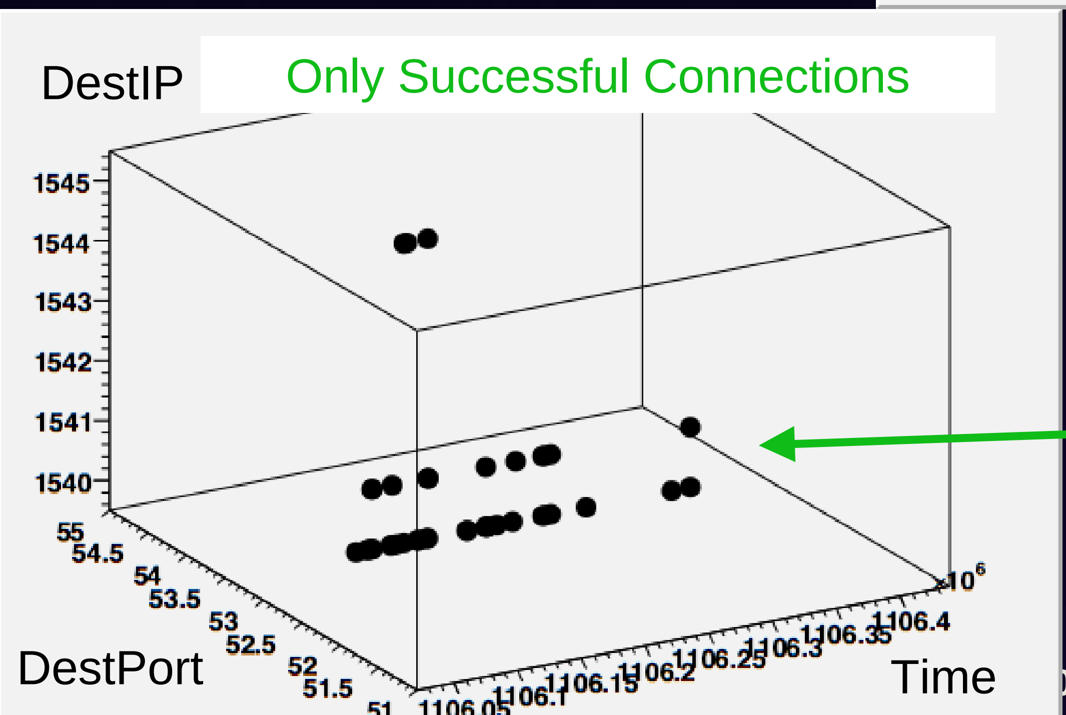
Scott Campbell, Eli Dart, Brian Tierney, Jason Lee
Kurt Stockinger

Lawrence Berkeley National Laboratory

Scan Analysis Example

- From IDS logs, we see that there might be a distributed scan coming from subnet 134.95.X.Y
- To verify this, we plot DestIP vs. DestPort vs. time
- Note large scan of port 6101

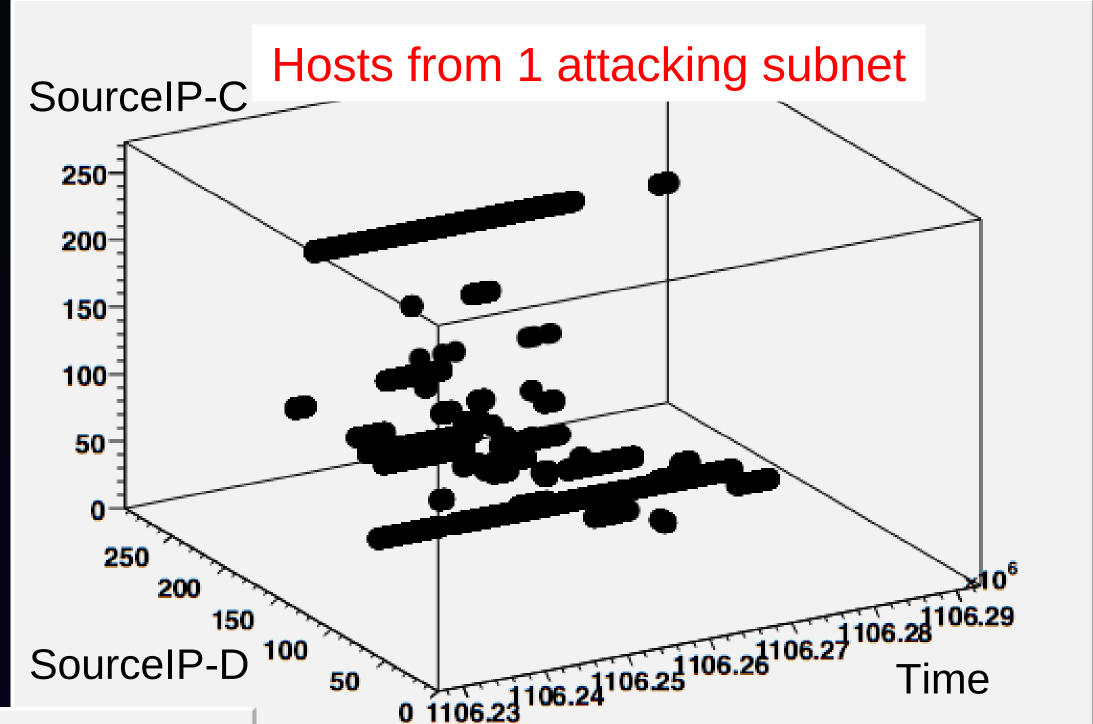


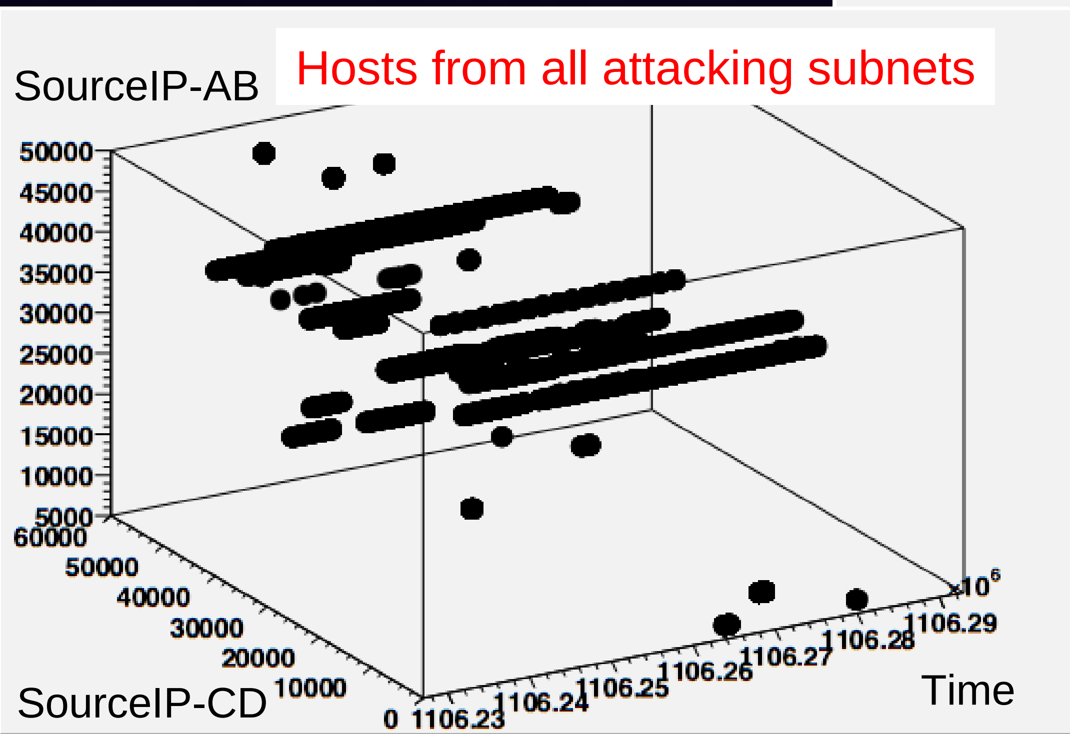


- Q: Were any of the connections form the attacking subnet successful ?
- A: Some of the traffic from that subnet is legitimate

Scan Analysis Example

- Plot the source of the attacks from *just the attacking subnet* to port 6101:
- Several hosts are part of the distributed attack (coordinated distributed scan)

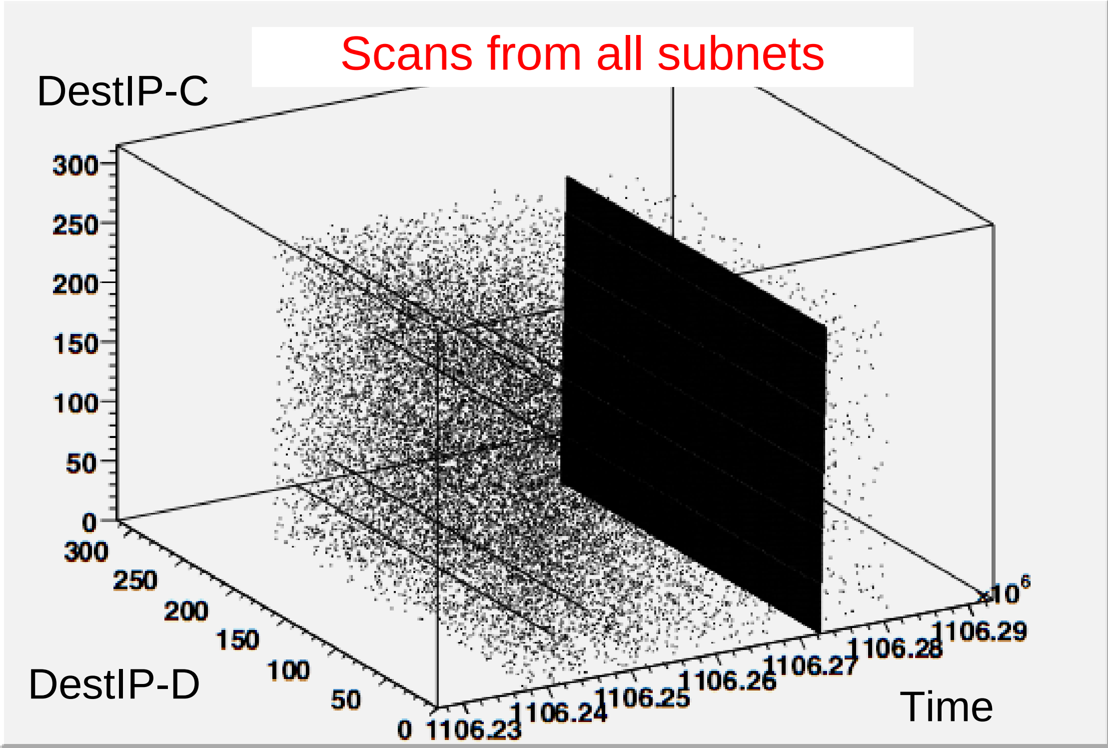


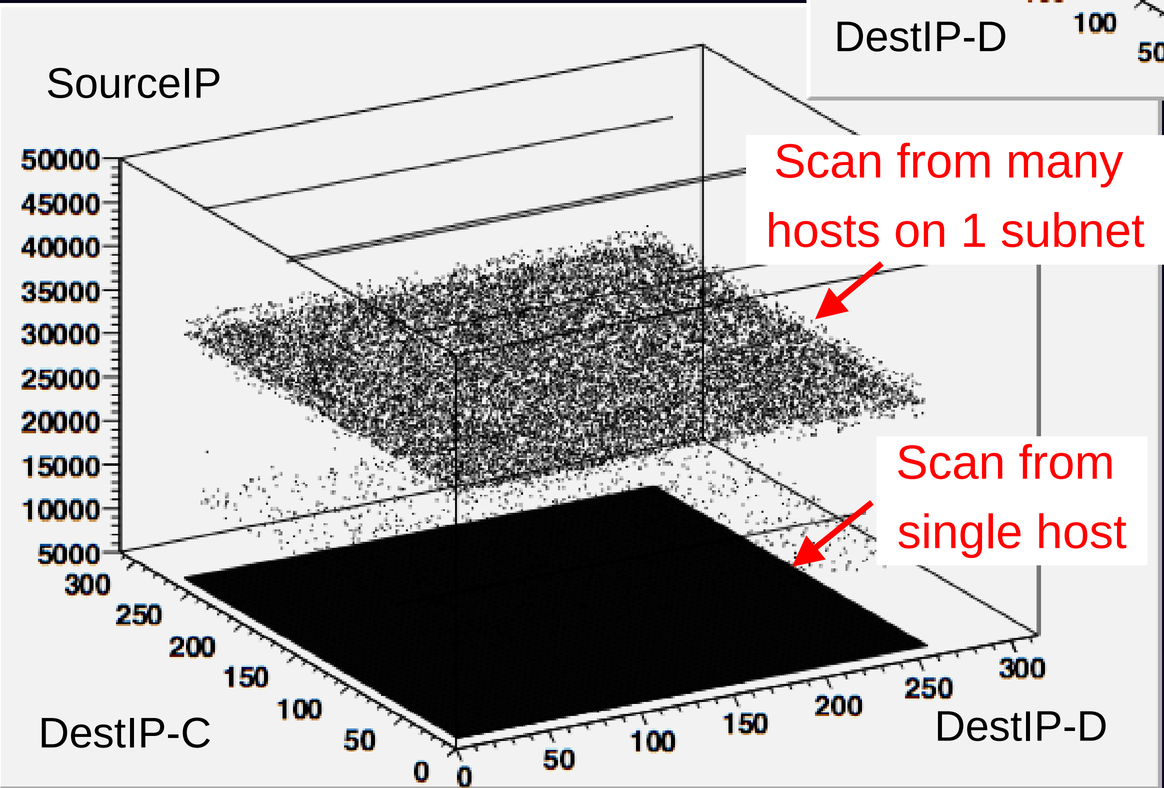


- Plot all scanners of port 6101 from *all subnets:*
- In addition to the coordinated distributed scan, there are also several other unrelated scans occurring in parallel

Scan Analysis Example

- Plot all scanners of port 6101 from *all* subnets:
- Besides the distributed attack from the 1 subnet, there are scanners coming from other subnets as well





- Plot of the source of the attacks:
- There are 2 separate large attacks:
  - from single host
  - from distributed hosts

FastBit Indexing for Efficient Data Queries

Kurt Stockinger, Kesheng (John) Wu

Lawrence Berkeley National Laboratory

Why Bitmap Indices?

- Goal: efficient search of *multi-dimensional* read-only (append-only) data:
  - E.g. temp < 104.5 AND velocity > 10^7 AND density < 45.6
- Commonly-used indices are designed to be updated quickly
  - E.g. family of B-Trees
  - Sacrifice search efficiency to permit dynamic update
- Most multi-dimensional indices suffer *curse of dimensionality*
  - E.g. R-tree, Quad-trees, KD-trees, …
  - Don’t scale to large number of dimensions (< 10)
  - Are efficient only if all dimensions are queried
- Bitmap indices
  - Sacrifice update efficiency to gain more search efficiency
  - Are efficient for multi-dimensional queries
  - Query response time scales linearly in the actual number of dimensions in the query

Bitmap Indices

a) list of attributes b) equality encoding c) range encoding

	$\pi_A(R)$	E^9	E^8	E^7	E^6	E^5	E^4	E^3	E^2	E^1	E^0	R^8	R^7	R^6	R^5	R^4	R^3	R^2	R^1	R^0
1	3	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	0	0	0
2	2	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	0	0
3	1	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	0
4	2	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	0	0
5	8	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	2	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	0	0
7	9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
9	7	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
10	5	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0
11	6	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
12	4	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	0	0	0	0

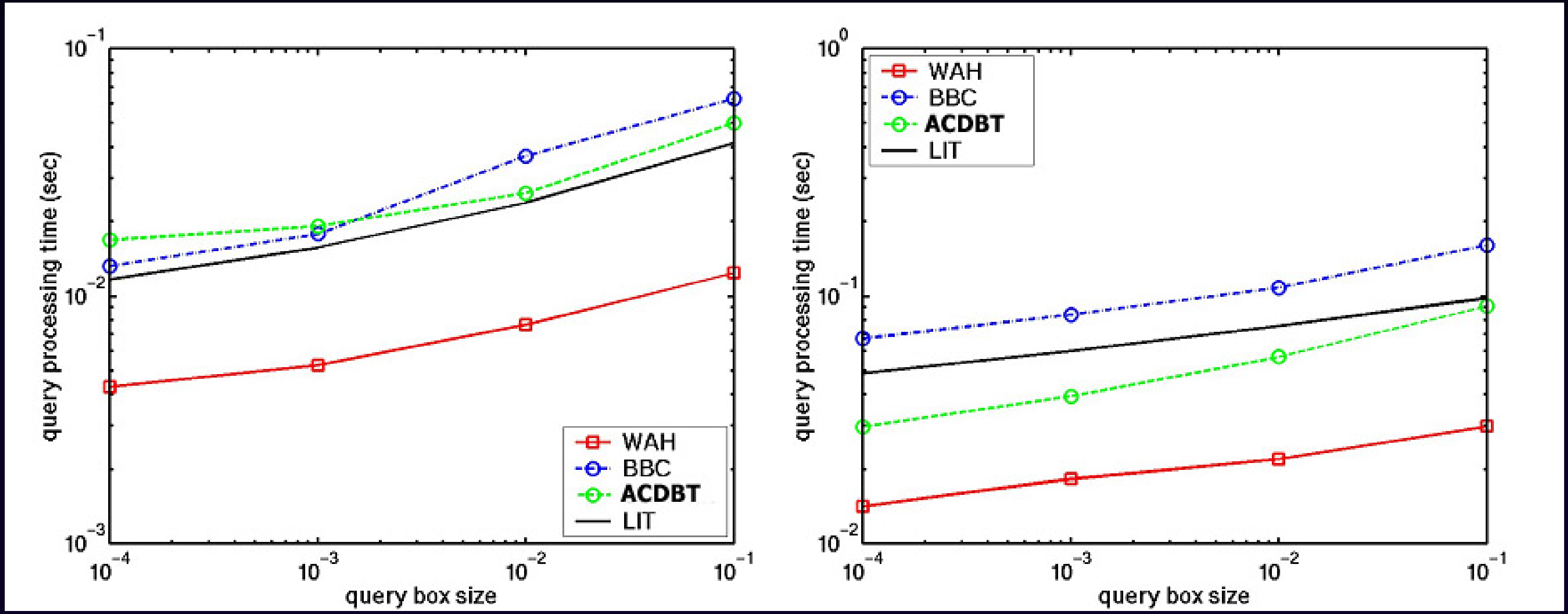
(a) (b) (c)

Equality encoding compresses very well

Range encoding optimized for one-sided range queries, e.g. temp < 3

Query Performance of FastBit

- FastBit uses WAH (Word-Aligned Hybrid) compression technique developed at Berkeley Lab
- FastBit significantly outperforms existing solutions



2D

5D

High Energy Physics Data

Network Traffic Analysis

- Monitor incoming and outgoing network traffic

  TCP Session Summaries:

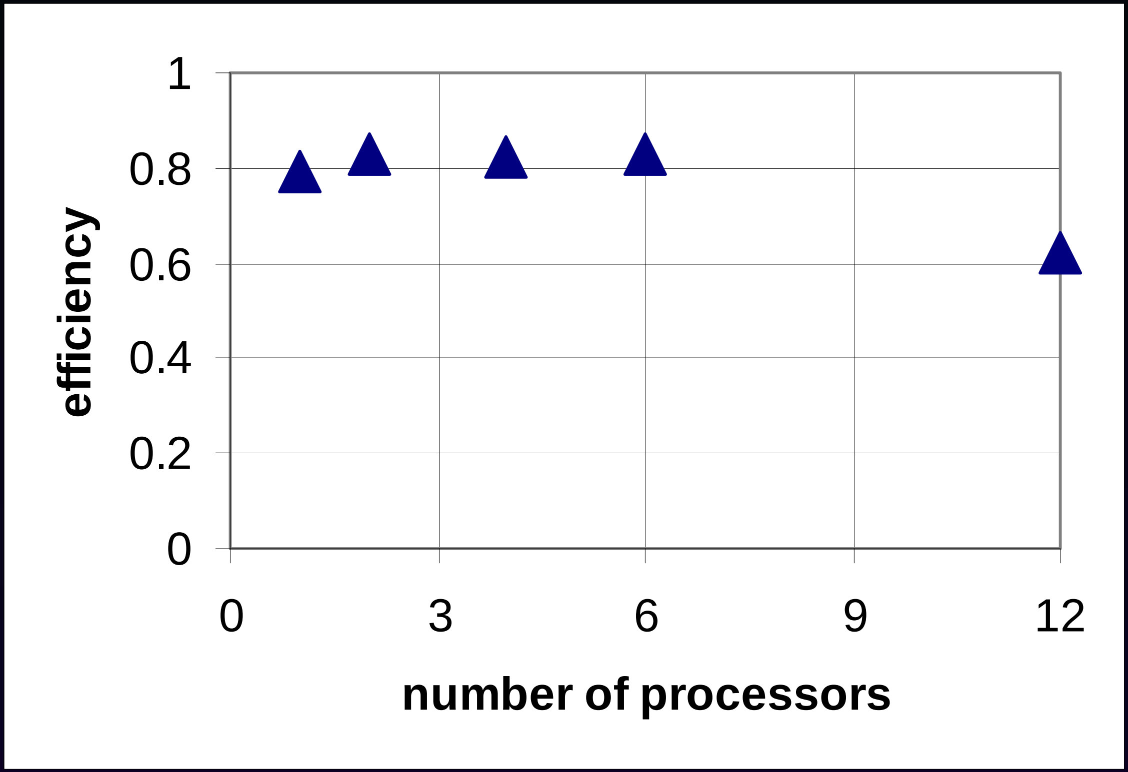
StartTime	EndTime	Protocol	SrcIP	DstIP	SrcPort	DstPort	Bytes	Packets
4.176	52.357	TCP	10.129.13.1	172.28.11.5	61089	80	6000	13
4.183	10.729	TCP	192.168.14.1	172.28.82.1	30274	80	508	3

- Goals:
  - Parallel visual data analysis
  - High-speed forensics
  - Large scale profiling
- Software Technology:
  - ROOT: data storage and analysis package developed by CERN
  - FastBit integrated with ROOT data analysis environment:
    - Large-scale parallel query processing

Experimental Setup

- FastBit + ROOT + MPI
- SGI Onyx
  - 12 SMP Processors (total size of shared memory: 12 GB)
  - Shared, parallel file system running IRIX
- Data:
  - 1.1 billion records
  - 24 weeks of network traffic data: ~241 GB
  - Size of compressed bitmap indices: 73 GB

Parallel Efficiency of the FastBit Query Engine



- Parallel efficiency is 80% in most cases.
- Using all 12 processors causes some contention with the OS, which degrades the parallel efficiency to 60%.
- Evaluating 3-dimensional query takes 22.8 sec with FastBit-ROOT
- (ROOT: sequential scan 2,467 sec, parallel: 206 sec)

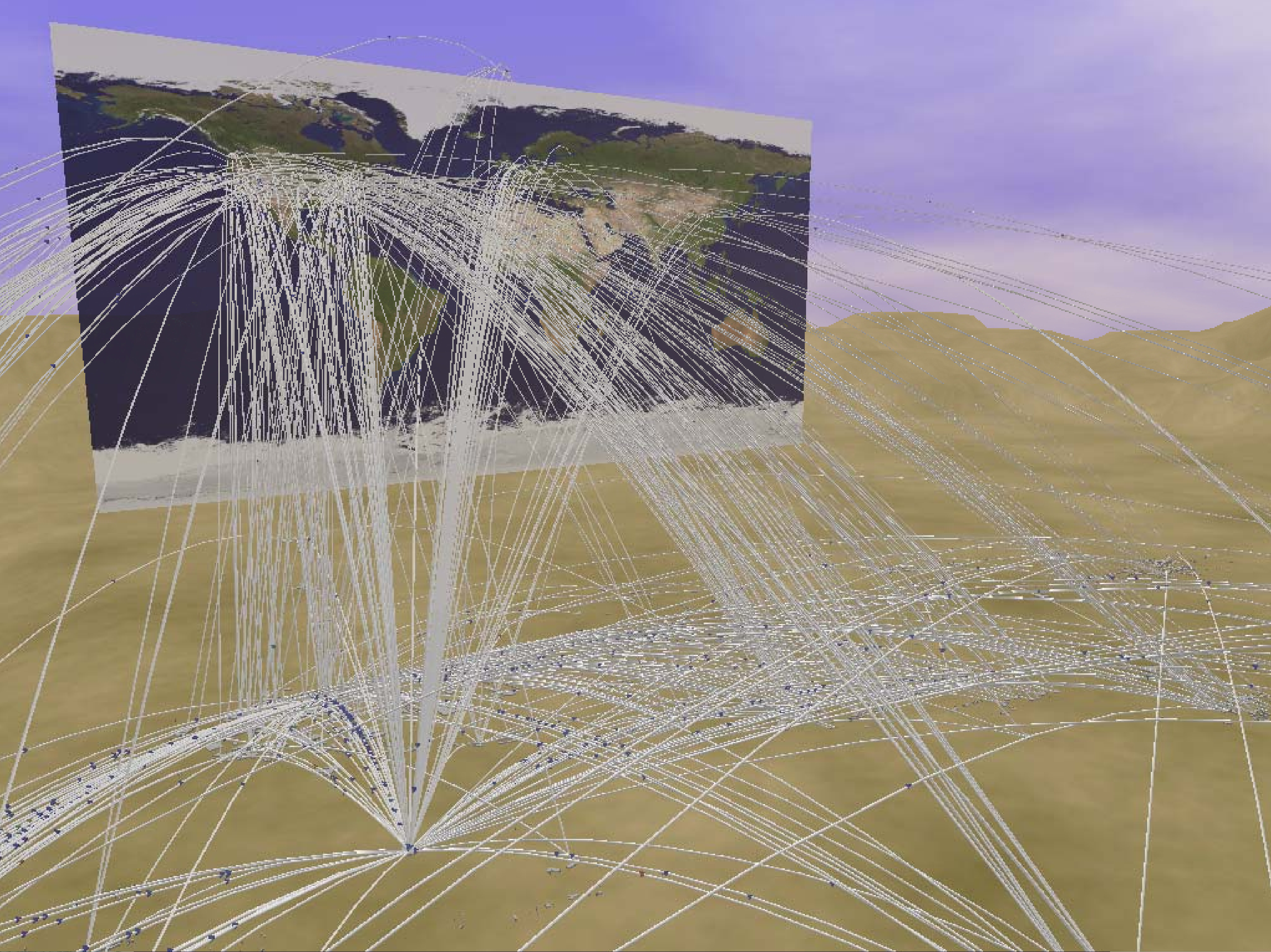
Automated Feature Detection and Interactive Visualization

Mike Fisk[1], Eugene Gavrilov[1], Alex Kent[1], Christopher E. Davis[1,2], Rick Olinger[2], Rob Young[2], Jim Prewitt[2], Paul Weber[1], Thomas P. Caudell[2], Steve Smith[1,2,3]

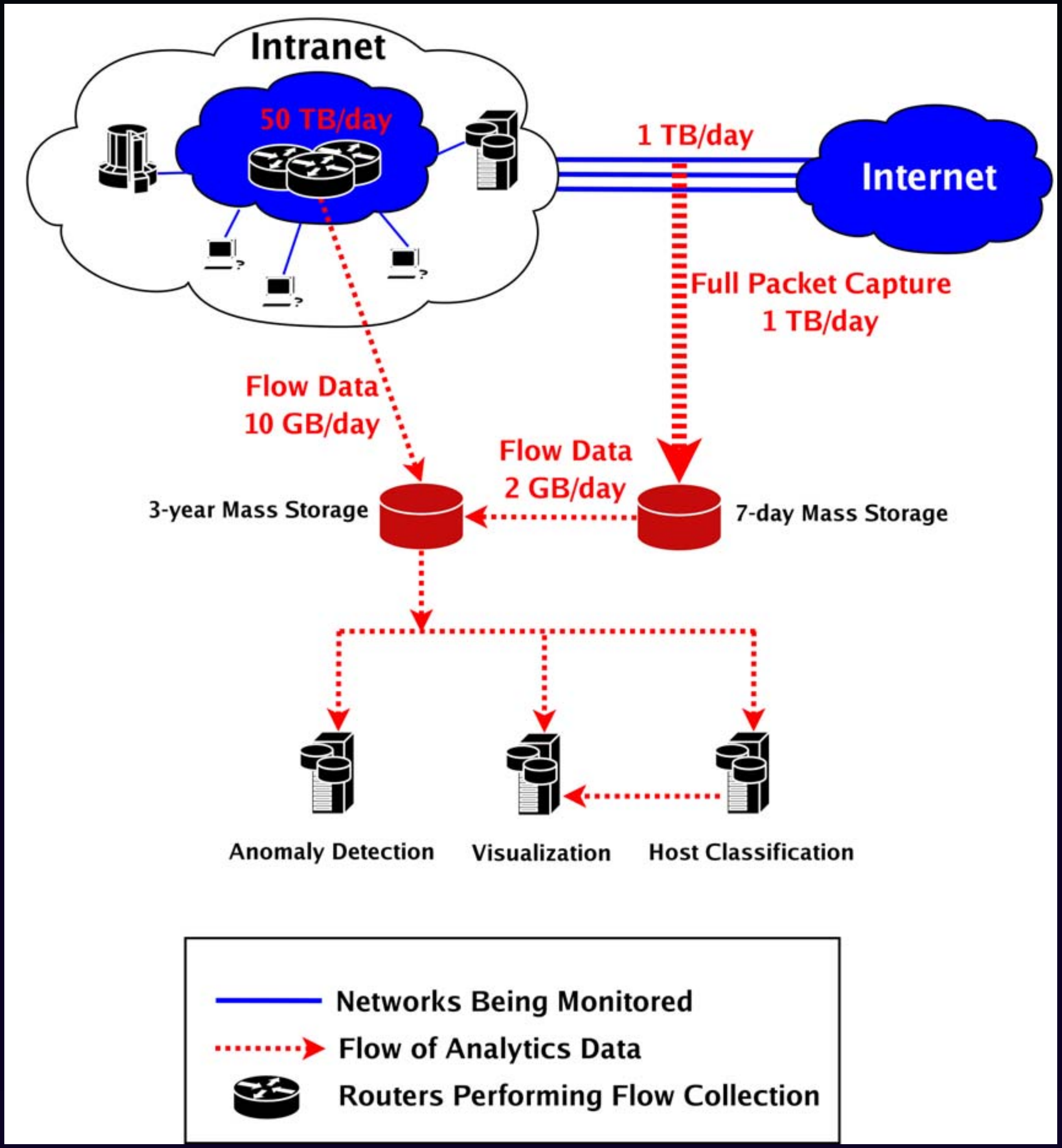
[1]Los Alamos National Laboratory

[2]University of New Mexico

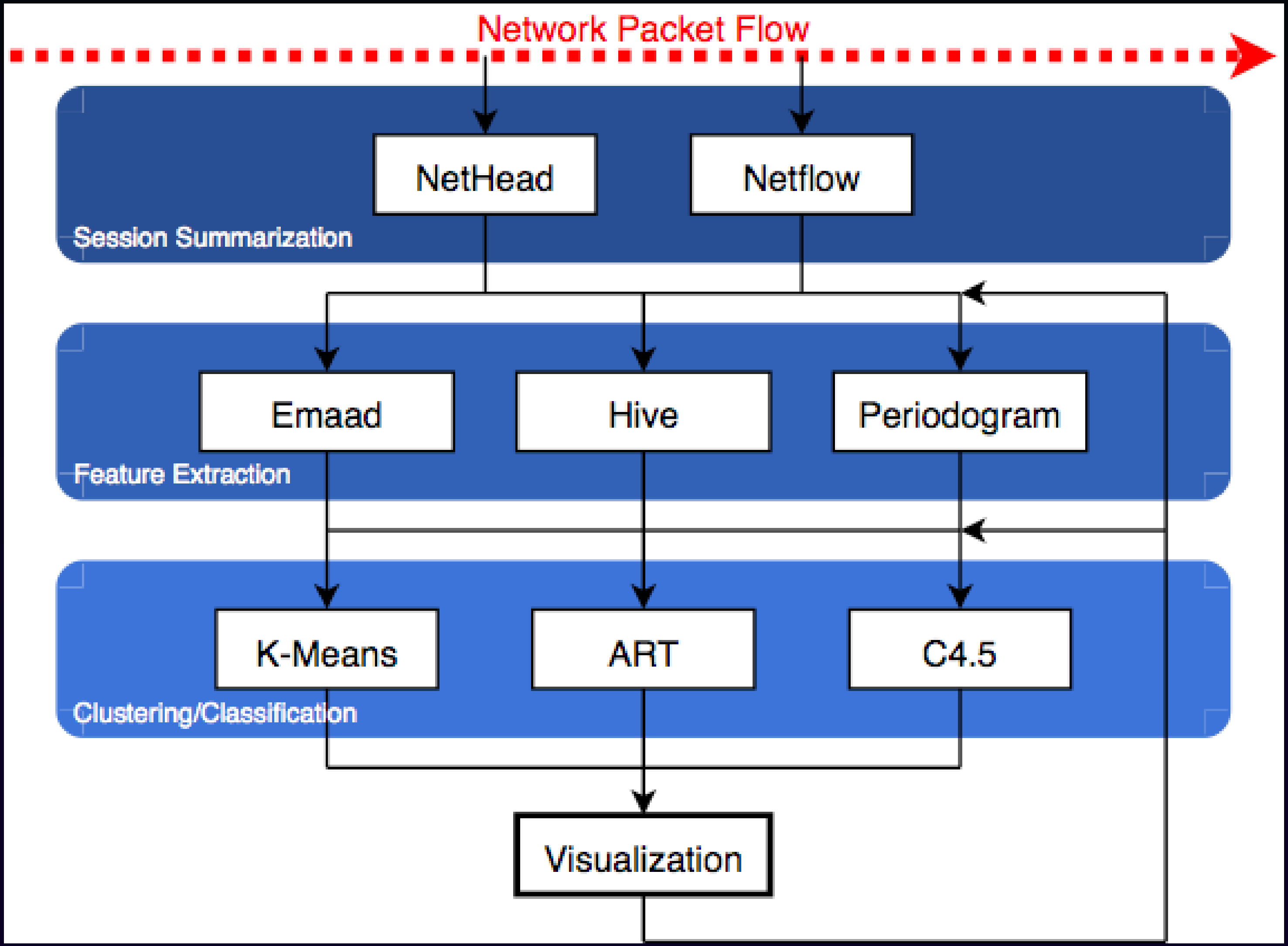
[3]Lawrence Berkeley National Laboratory



System Architecture @ Los Alamos

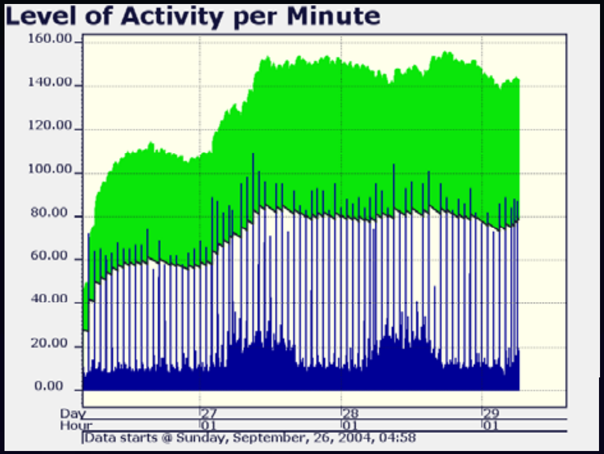


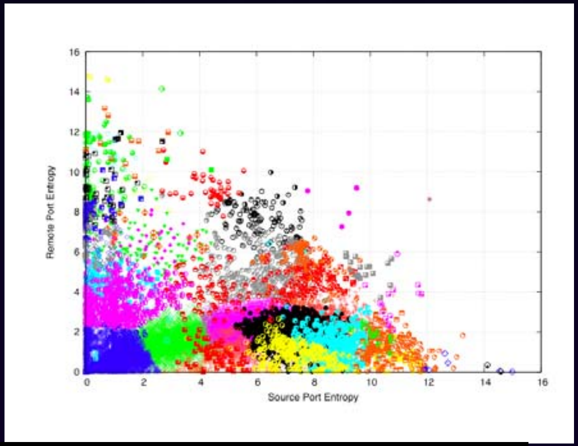
Analytical Process @ Los Alamos

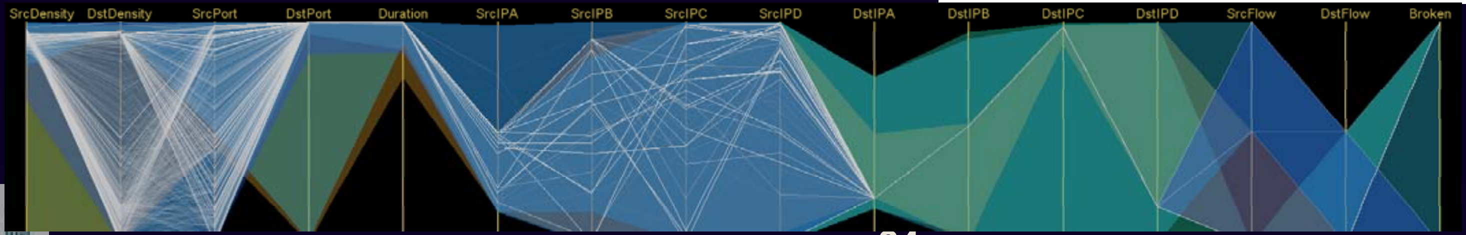


Feature Extraction and Classification

- Time-based
  - Exponential moving average
  - Clustering of time segments
- Session based
  - Clustering of session summaries
- Host based
  - Clustering of host statistics



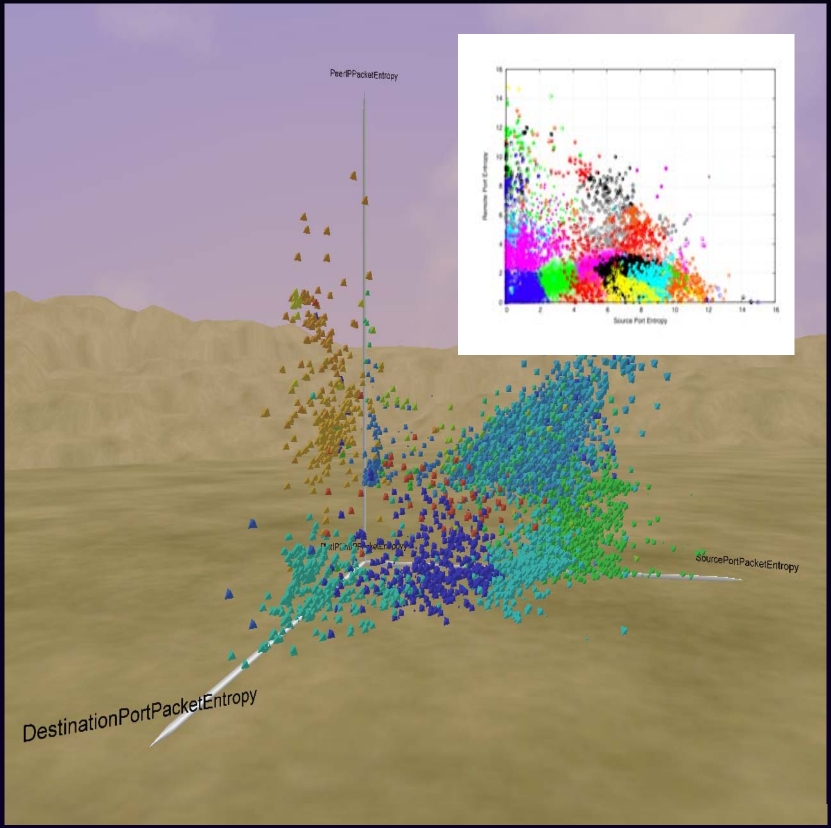




Characterization by Host Statistics

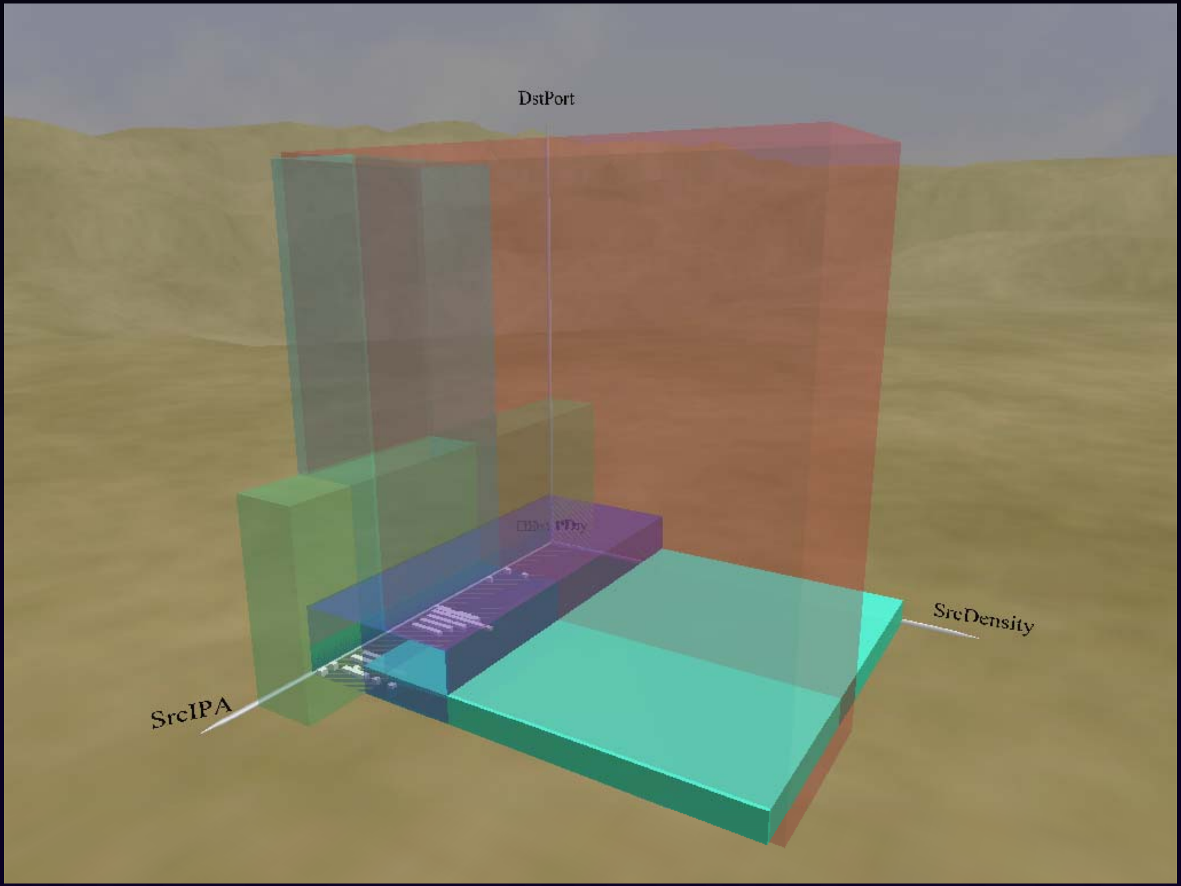
- Measures
  - Mean
  - Standard deviation
  - Shannon entropy
- K-means clustering
- Expert classification and labeling of clusters

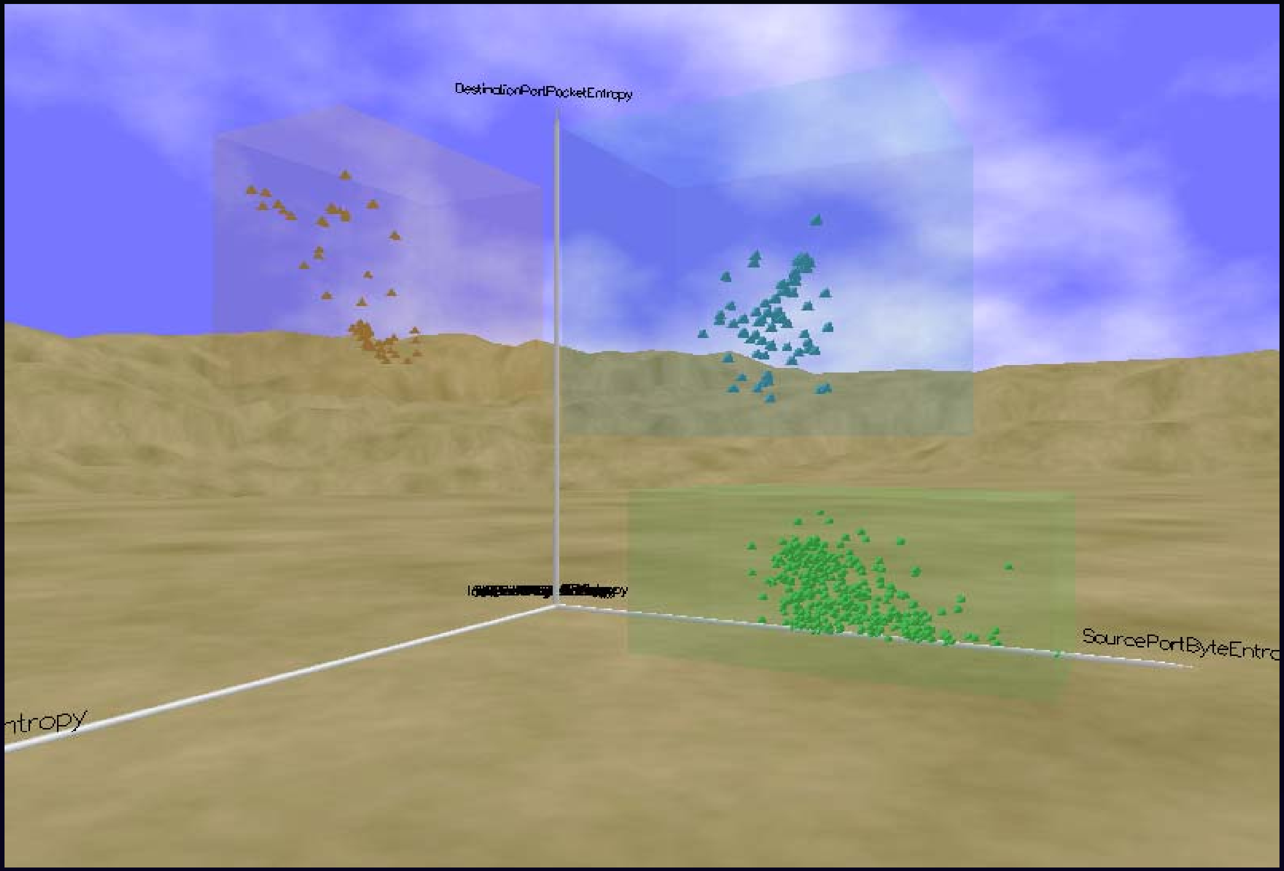
  (*normal client, e-mail server, web server, scanner, etc*.)
- Calculate hyperboxes from clusters for use as range queries over larger data sets.

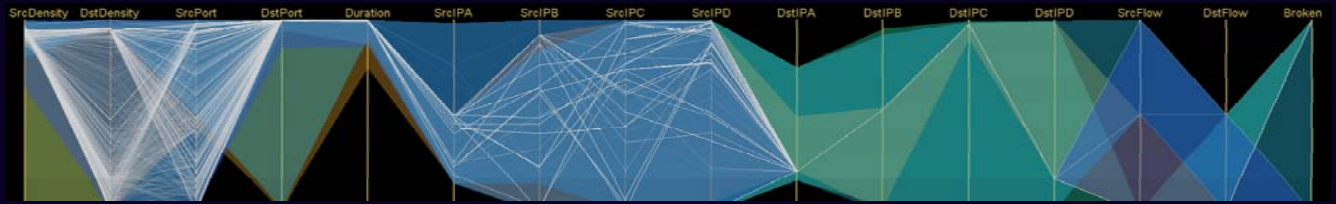


Clustering of Session Summaries

- Adaptive Resonance Theory (ART) neural network
- Hyperbox template generation and testing.







Hyperboxes describe high dimensional range queries

QuickTime™ and a
H.264 decompressor
are needed to see this picture.

Query-driven Visual Analytics Summary

- Ultra-scale data sets (24 Weeks - 10^9 records - 25 dimensions)
- Query-driven analysis
  - Exploratory network analysis
  - Automated clustering
- FastBit - High Performance Data Analysis:
  - Combined with ROOT (developed at CERN)
  - Parallel query evaluation of large data sets
  - Visualization of network traffic to identify malicious hosts etc.
  - 3-dimensional query on 1.1Billion records takes 22.8 sec with FastBit-ROOT

    (ROOT: sequential 2,467 sec, parallel: 206 sec)
- High dimensional data and queries demand novel visualization

Come to our Live Demos

FastBit Performance

Visual Browsing and Retrieval

Feature Extraction

Clustering and Classification

Novel Visualization

The End