Cyber Analytics
Applications for
Data-Intensive Computing

Mike Fisk
Los Alamos National Laboratory
Outline:
An Applications Talk (mostly)

- Motivation
- Requirements
- Characteristic Cyber Problems
  - Query
  - Time-series change detection
  - Graph mining
- Our Approach to Map-Reduce Parallelism
Motivation

- National Cyber Infrastructure is vulnerable and regularly penetrated
  - Every major defense contractor, national lab, etc.
  - Intrusions into Google, Adobe, oil sector now publicly acknowledged

- Threats are viral
  - Initial vector grants insider access somewhere in a network
  - Intruder/Insider spreads hop-by-hop through networks and trust relationships between networks
  - Contemporary exploitation is normally at very subtle rates
    - But epidemic “Pearl Harbor” attacks are a constant threat

- Necessitates:
  - Rapid, automatic detection
  - Epidemic-speed dynamic defense
Problem Definition

Definition: A *misuse* of a networked system and include one or more of the following observable acts:
- Penetration (*Intrusion, Exploitation*)
- Remote command & control
- Exfiltration of data
- Denial of availability or integrity (*Attack*)

Problem: Given observed sensor data…
- Detect known attack methods and tools
- Detect unexplained patterns that could be attacks
  - Prioritize response to patterns based on likelihood that it’s an attack
Represented as Temporal Graphs

- Many cyber data sets can (and should) be described as graphs
  - Vertices are hosts, users, etc.
  - Directed edges are communications
    - Discrete packets or flows with durations
  - Events from heterogeneous sensors can be combined in one graph

- A graph construction supports traditional analysis while enabling new analysis
  - Subtle exploitation is often a path through the network
  - Structural characteristics
Data-Intensive Scale & Real-Time

- Rapid data rates
  - LANL (national scales are much larger):
    - 10 gigabit network links being monitored
    - 1 TB/day in general-purpose traffic to the Internet
    - 100 million flows (edges) per day

- Online/streaming decision making
  - Penalty for latency (limited time to catch and stop a worm)
  - Streaming visualization with query-driven context and drill-down
  - Automatic response (since 2003 at LANL)
    - Framework for Responding to Network Security Events (FRNSE)
    - Responses are network quarantine (switch, firewall, DNS)
Exponential Attacks: The battle is over before we know it

25 Jan 2003: *Slammer Worm*
- 75,000 observed infections
- Doubled every 8.5 seconds
- Saturated networks in 3 seconds
- 90% of vulnerable hosts infected within 10 minutes

2009: Conficker worm infects 9M hosts
- Hybrid: network infection as well as removable media
- French Air Force grounded because of inability to access flight plans

2003: Slammer: 75,000 machines in 30 minutes
Application #1: Query & Retrieval

- Fast inject rates
  - Many times just a day/month/year ring buffer
  - Only summarized data stored permanently

- Boolean queries (*tips, signatures, black-lists*)
  - Non-relational, embarrassingly parallel

- Aggregate queries (*trending, features for change detection*)
  - Embarassingly parallel if data partitioned properly
  - Map-shuffle-reduce parallel otherwise

- Relational queries (*coincident events*)
  - Recursive SQL or graph algorithms
  - Parallel requires data replication or lots of communication
    - Graph partitioning optimizes communication
How Big is a Big Query Problem?

- **Transactional Relational Database: 10-100TB**
  - Oracle, DB2, etc.

- **Massively Parallel Processing Databases: >1PB**
  - Greenplum, Netezza, Hadoop/Hive, etc.
    - eBay – 6.5PB (Greenplum)
    - Facebook – 400TB compressed, >10TB/day (Hadoop/Hive)
  - Data partitioned; distributed storage across nodes
  - Optimized for warehousing vs. transaction processing
  - Column vs. Row Storage
  - Weakened consistency

- **Comparison with LANL network data**
  - 6 TB/day, 10TB/year permanent
Problem #2:
Time-Series Anomaly Detection

- Why? Lack of ground truth information
  - Labeled data is rare, synthetic, and not representative
  - Normal data has both malicious and non-malicious activity
  - Moving target (new users, apps, protocols, etc)
  - Some sensors are inscrutable block-boxes which can only be described through experiment

- Focus on Change Detection: anomalies w.r.t. time
  - Kernel-Smoothed Adaptive Thresholds
  - Relative Entropy
  - Hidden Markov Models
  - Machine Learning algorithms

- Feature selection
  - Fundamentals of adversary objectives
  - (Cyclo-)stationary under normal circumstances
  - Local, path, neighborhood, and global properties
Continuous Adaptive Algorithms

Asymmetric EWMA
[Fisk & Gavrilov ‘05]

- Optimized for efficiency over accuracy
  - Memory utilization: 2 floats per model
  - Updates: 1 conditional & 3 FLOPs

- Accuracy tradeoff
  - Predicts upper bounds of periodic behavior, not the periodic behavior itself
    - Bursts at wrong times not detected
Kernel-Smoothed Adaptive Thresholds

- Extensions to [Lambert & Liu ’06]
- Per time-of-day and day-of-week models
  - Supports periodic behavior for certain common periods
  - Smoothed for sparse data (rather than quadratic interpolation)
- Negative binomial model provides sound probability estimates
- Cumulative Sum amplifies consecutive anomalies
- Experiment in optimizing accuracy rather than efficiency
  - SMP & map-reduce versions under development
Problem #3: Non-Local Graph Analysis

Global, regional, and local scale
Global Properties of Graphs

Connected Components introduced in [Collins & Reiter '07]
Cyberspace activity is represented as a cohort of temporal graphs representing different observational perspectives of the same underlying events.
Temporal Coincidence

- Types of Coincidence
  - A node has multiple interesting edges within some time window (but the edges may not be interesting for the same reasons)
  - A number of similarly interesting edges occur within some time window (not necessarily having any nodes in common)
  - There is a path of interesting edges
    \[ v_0 \rightarrow v_1 \text{ at time } t_0, \quad v_1 \rightarrow v_2 \text{ at } t_0 < t_1 < t_0 + k, \quad \ldots \]

- What if we know the observed graph is missing edges with some probability?
- What if we know that edges are false positives with some probability?
- What if there is a pairwise similarity metric for edges?
  - Many attacks are polymorphic but have common elements
Malware Trace Analysis

- Malware has software protection measures built-in
  - Run-time unpacking/decoding
  - Debugger detection

- “Covert debugging”
  - Hypervisor-based instruction trace generation

- Malware analytics challenges
  - Families, lineage
  - Identifying functionality

- Approaches
  - (Sub-)Graph distance metrics, clustering
  - Binding points (library calls, system calls)
Parallel Computation
Computational Approach: File-Oriented Map-Reduce

- Success of the M-R programming model is the ease of constructing parallel & distributed jobs from serial programs
  - Class of problems not requiring continuous use of global shared memory

- Observation: Key → Value tuples perhaps overly abstract
  - Serial programmers can & do deal with more than one tuple/data-point at a time
  - Sort not always necessary
  - Some hierarchical data types (e.g. packets) not well-suited to tuples

- File-Oriented
  - Map files to files, partition files, distribute files, reduce files
  - Existing analytical/programming environments & tools easily used
    - Awk, embedded databases
  - Amortize run-time costs by file rather than by tuple
FileMap: File-Oriented Map-Reduce

[fmisk.github.com/filemap ‘08]

- Thin orchestration layer on top of standard platforms
  - In contrast to monolithic systems such as Hadoop with their own filesystems, security models, etc.
  - Uses remote execution and file copy infrastructure of your choice

- Intermediate result caching
  - Iterative query refinement
  - Redundant queries when multiple people working the same issue

- Out-of-band injest
  - If file appears on a node’s filesystem, it is usable
  - May even be generated locally if the node is a sensor

- Continuous jobs that process new data as it arrives

```
fm store /tmp/*.txt /etext/
fm map -i "/etext/" "sed -f words.sed | fm split -n 100 |> sort | uniq -c"
```
Conclusions

- Cyber security is an evolving application domain that is maturing from labeling edges in graphs to detecting anomalous spatial and temporal patterns.
- Simple queries are large enough to exercise data-intensive, parallel systems.
- Sophisticated (combinatorial) analysis creates further demands.