Lawrence Livermore National Laboratory

Hierarchical Bloom Filters: Accelerating Flow Queries and Analysis

January 8, 2008

FloCon 2008

Chris Roblee

DOE Computer Incident Advisory Capability (CIAC)

Lawrence Livermore National Laboratory, P. O. Box 808, Livermore, CA 94551

This work performed under the auspices of the U.S. Department of Energy by
Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

UCRL-PRES-236738

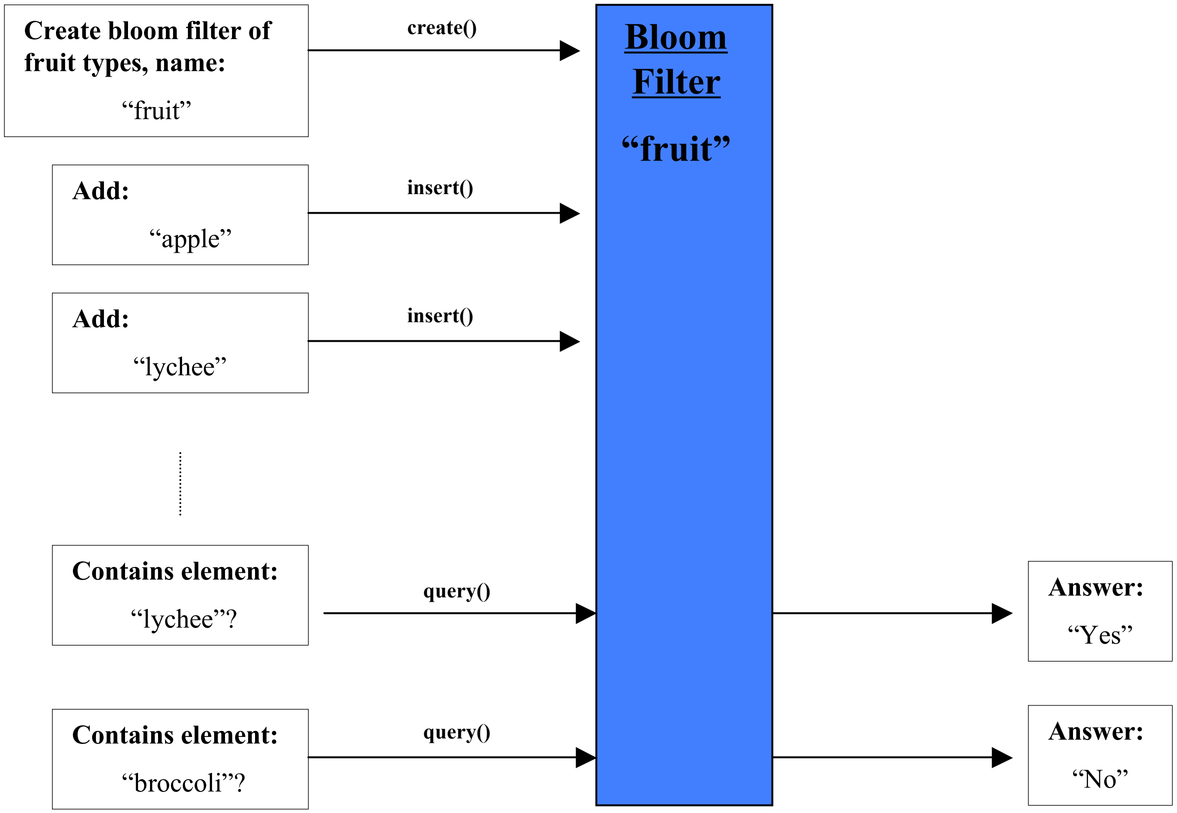
Overview

- Introduction to Bloom Filters
- Overview of CIAC’s Bloom Filter-Based indexing System
- Approach's Applicability for CIAC & other CERTs
- Performance on Actual Flow Data
- Applications of Approach in Conjunction With Analytical Tools
  - Facilitating incident detection and analysis with flow visualization tools.

A Very Brief Introduction to Bloom Filters

Introduction to Bloom Filters

- High-level Functionality – trivial



http://www.eecs.harvard.edu/~michaelm/NEWWORK/postscripts/BloomFilterSurvey.pdf
http://en.wikipedia.org/wiki/Bloom_filter

Introduction to Bloom Filters

- **The Concept**
  - Efficient, probabilistic data structure, providing extremely light-weight string lookups, or “approximate membership queries”.
  - Invented by Burton Bloom in 1970 to optimize spellchecking.
  - Trade-off small probability of **false positives** for massive gains in **space and time efficiency**.
  - Popular for various large-scale network applications (e.g., web caches, query routing).

References:

http://www.eecs.harvard.edu/~michaelm/NEWWORK/postscripts/BloomFilterSurvey.pdf

http://en.wikipedia.org/wiki/Bloom_filter

How Bloom Filters Work

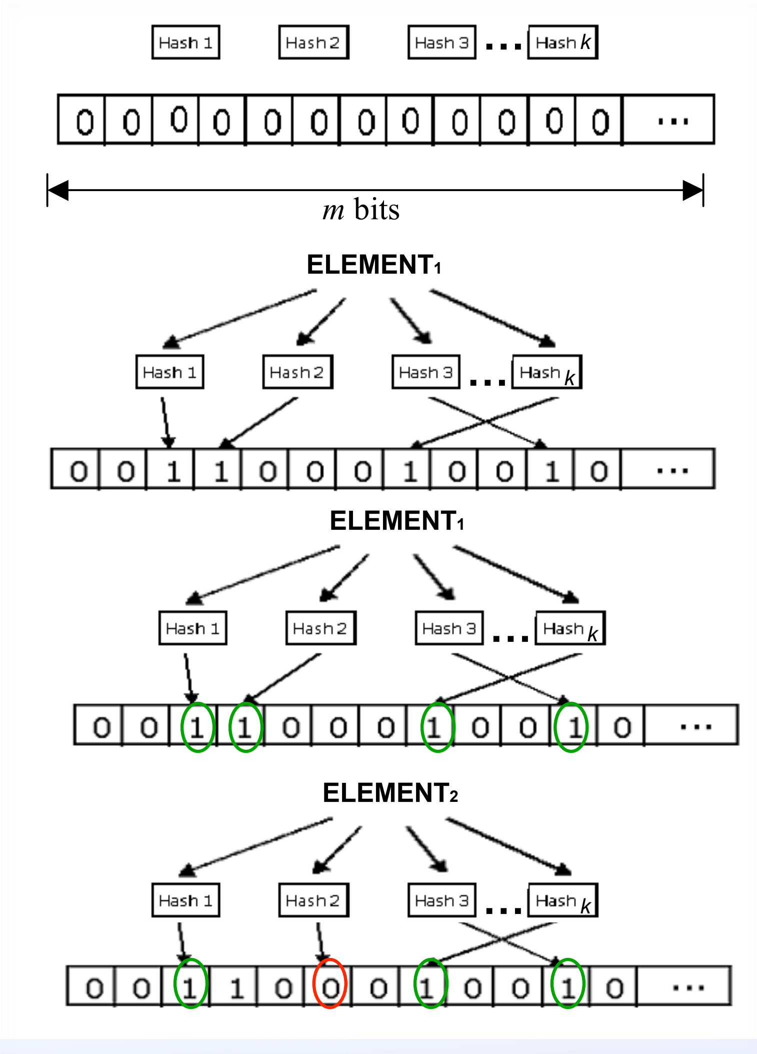
1. <u>Empty</u> bloom filter is a bit array of *m* ‘0’- bits.

2. Introduce *k* different hash functions, each maps key value to one of m array positions.

3. <u>Insert</u> element by feeding it to each hash function, to obtain *k* array positions. Set these bits to ‘1’.

4. <u>Query</u> element (check its existence) by re-feeding into each hash function, and checking corresponding bit positions. If all bits are ‘1’, then element is either in the filter or it’s a <u>false positive</u>.

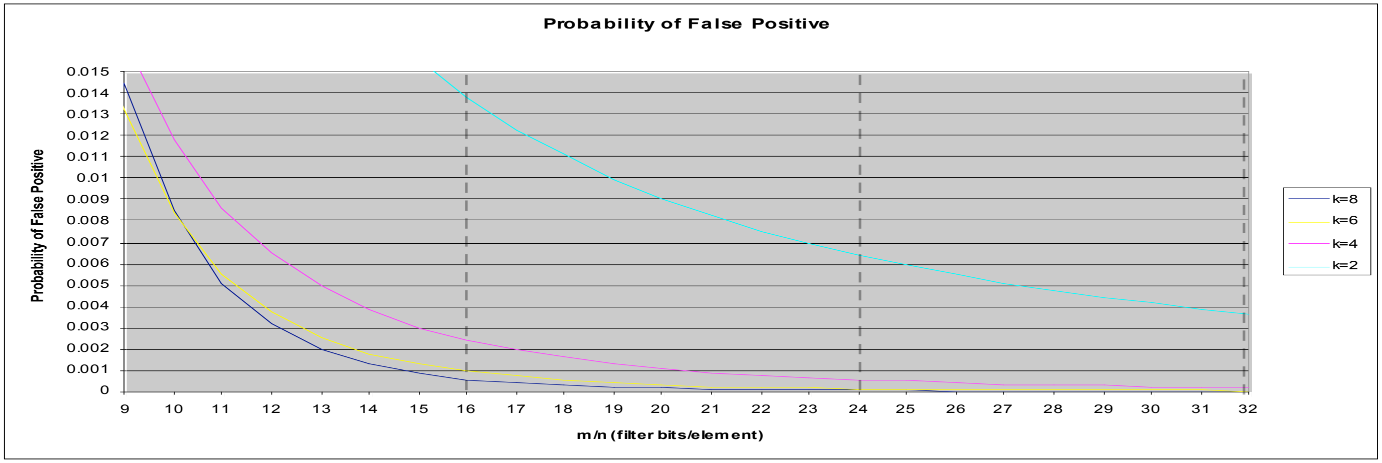
5. If bit positions of hashes of an element contain a ‘0’, then that element is <u>definitely</u> <u>not</u> in filter (no false negatives).



Introduction to Bloom Filters

- False Positives
  - Probability of false positive for a populated bloom filter is:

$$p(FP \approx \left(1 - e^{-kn/m}\right)^k$$



- k - number of hash functions used
- n – number of elements inserted
- m – size of bloom filter (bit array)

Bloom Filters - Summary

- Quick test of element membership:
  - 0 likelihood of false negatives
  - Tunable false positive rates
- Probability of collisions proportional to the number of elements in set & inversely proportional to filter size.
- Enforce maximum false positive threshold by tuning filter size:
  - Often require as little as one byte per element

Functionality

- Significant space and time advantages over many standard, deterministic indexing structures:
  - Self-balancing trees
  - Tries
  - Hash-Tables
  - Arrays, Linked Lists
- Query time is O(k), independent of number of items in set.
- Many open source implementations available.

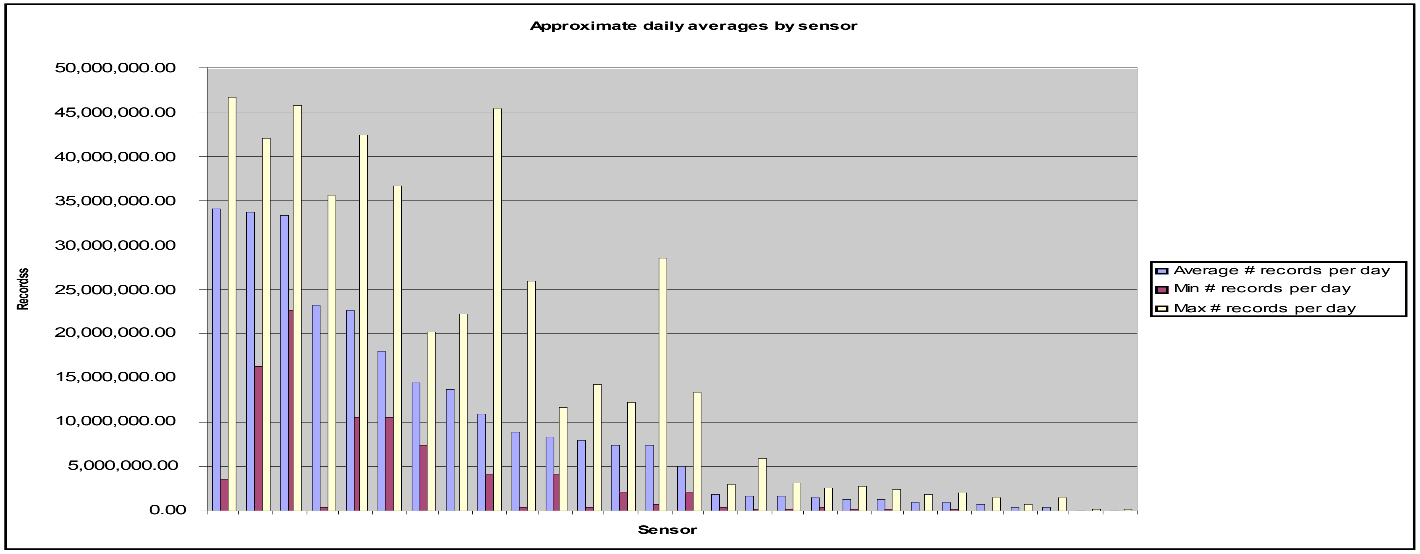
Practicality

Inexpensive, easy to deploy and maintain

Bloom Filters: Operational Viability for CIAC and the CERT Community

CIAC's Flow Collection Review

- CIAC collects massive volumes of biflow data from 29 sensors across the DOE complex:
  - 300-500 million biflows daily (~4600/s)
  - ~14GB/94GB compressed/uncompressed daily



CIAC’s Flow Collection Review

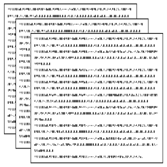
- biflow feed:
  - Session summary
  - Fields:
    - Date/Time & Duration
    - Source/Destination IP and Port
    - Protocol Information
    - Bidirectional Byte and Packet Counts
    - Bidirectional Protocol Options
    - Subset of TCP/ICMP flags

Example Biflow Record

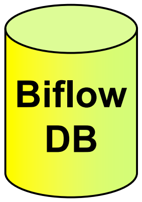
```
1171066191.997532,20070210000951.997532,site3,flo30,6,192168081021,192,168,81,21,IT,010000001008,10,0,1,8,US,53,1024,0,0,0.0000,0,0,54,0,1,0,0,0,0,0,0,60,0,60,0,,,14,00,+14,0,0,0,0
```

CIAC Analysis - Legacy Search Methodologies

- File *grep*
  - Search sensors and hours for range of interest (e.g., “site3, site12, site21 from 10/1/06 through 12/31/06”).
  - Requires reading/decompressing and combing through GBs of data (from disk) for every day searched.

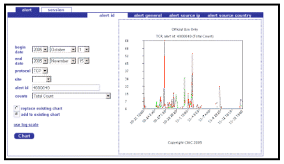


- RDBMS - Oracle
  - SQL+
  - Perl/JDBC
  - Typically limited* to past ~25 days of bi-directional sessions (~15%)



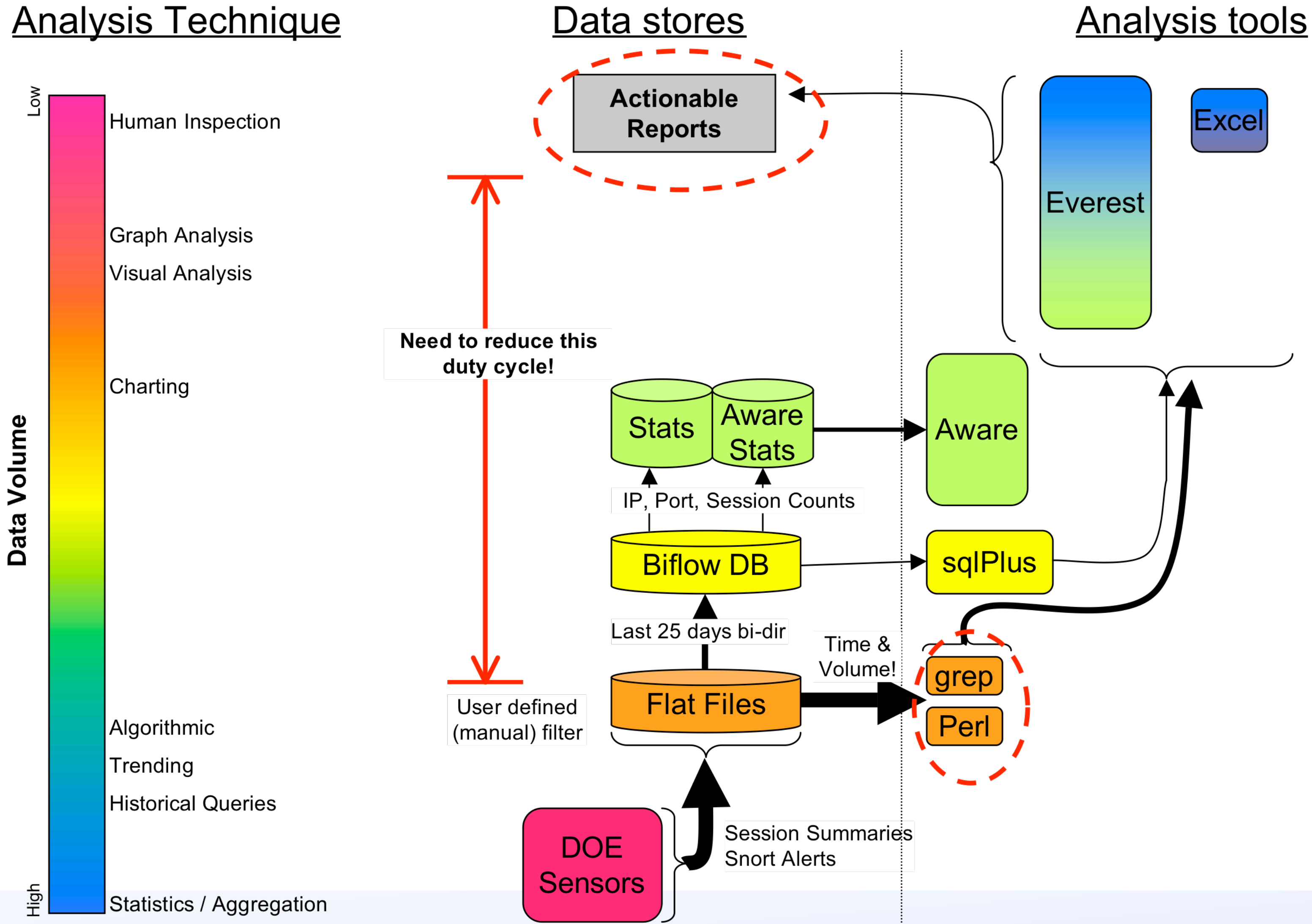
Biflow DB

- AWARE web portal
  - High-level charting and statistics (session counts, etc.)



<u>Many mission-critical searches can take several hours or days to complete</u>

Current CIAC Analysis Data Flow



Watch and Warn Query Needs and Issues

- Rapidly search all flow data over long periods of time:
  - Analysts typically search on IP address:
    - Watch list (suspicious, known-bad, etc.)
    - Nodes of interest
    - Compromised internal nodes
  - Various time (hours, days, months) and space (single site, all sites) scales.
  - Require quick turnaround (minutes) to respond to site requests:
    - e.g. "Have you seen these IPs at my site in the past 3 weeks?"

- IP-based searches often yield relatively small result sets:
  - "Interesting" IP might only have been seen in 30 site-hours, whereas 21,600 hours (~1 DOE-month) might have been searched.

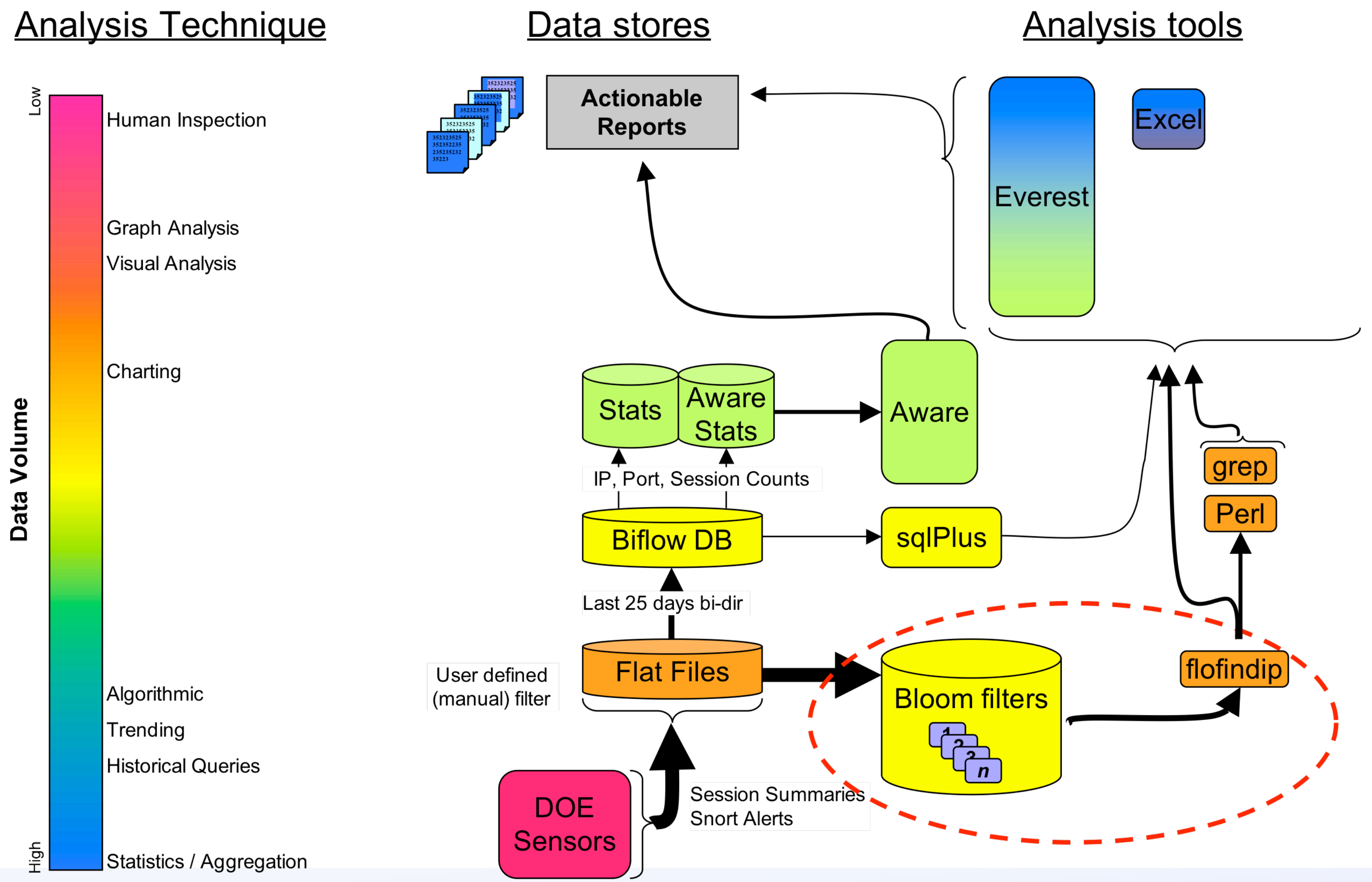
    ➔ 99.9% wasted duty cycle!
  - Need to reduce the search space (raw flow files) through better cataloging of data as it arrives.

Bloomdex:
CIAC’s Bloom Filter-based Indexing System for Network Flow Analysis

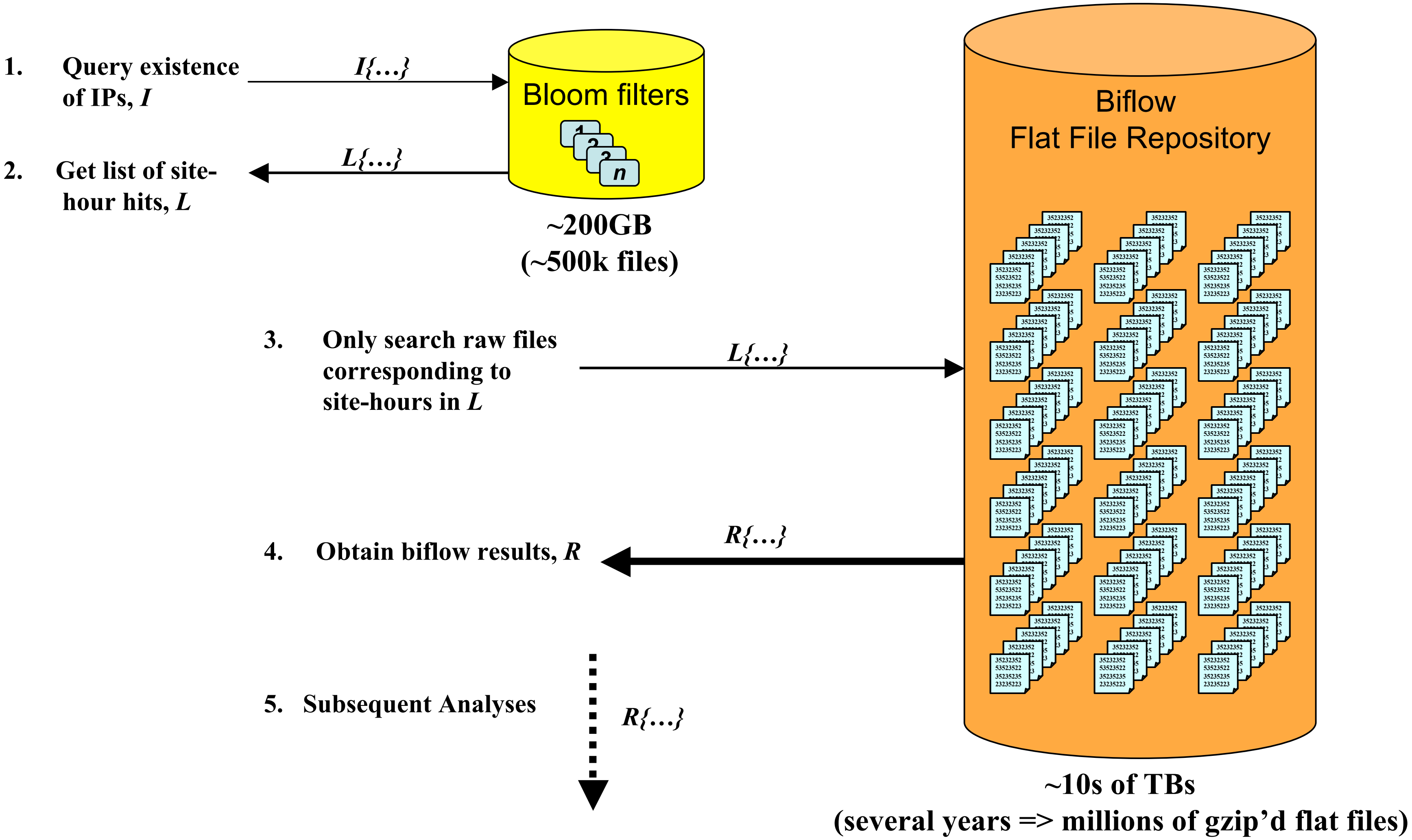
Solution: Bloomdex

- ***Bloomdex***
  - A hybrid hierarchy/file-based Bloom filter system to index CIAC's biflow records.
  - Currently indexed by source or destination IP.
  - Index partitioned by:
    - Site-month (e.g., "SITE8 11/2006")
    - Site-day (e.g., "SITE8 11/5/2006")
    - Site-hour (e.g., "SITE8 11/5/2006 13:00")
  - Uses intuitive directory tree structures and multi-scale bloom filters to accelerate IP-based searches.
  - max(FP rate) ≈ 2×10^{-4} → **3 bytes of storage per unique IP**

Blooomdex - CIAC Analysis Data Flow



Reducing the Biflow Search Space



1. Query existence of IPs, *I*
2. Get list of site-hour hits, *L*
3. Only search raw files corresponding to site-hours in *L*
4. Obtain biflow results, *R*
5. Subsequent Analyses

Bloom filters

~200GB
(~500k files)

Biflow
Flat File Repository

~10s of TBs
(several years => millions of gzip'd flat files)

Bloomdex: Performance Profile

Bloomdex: Comparative Performance Profiles

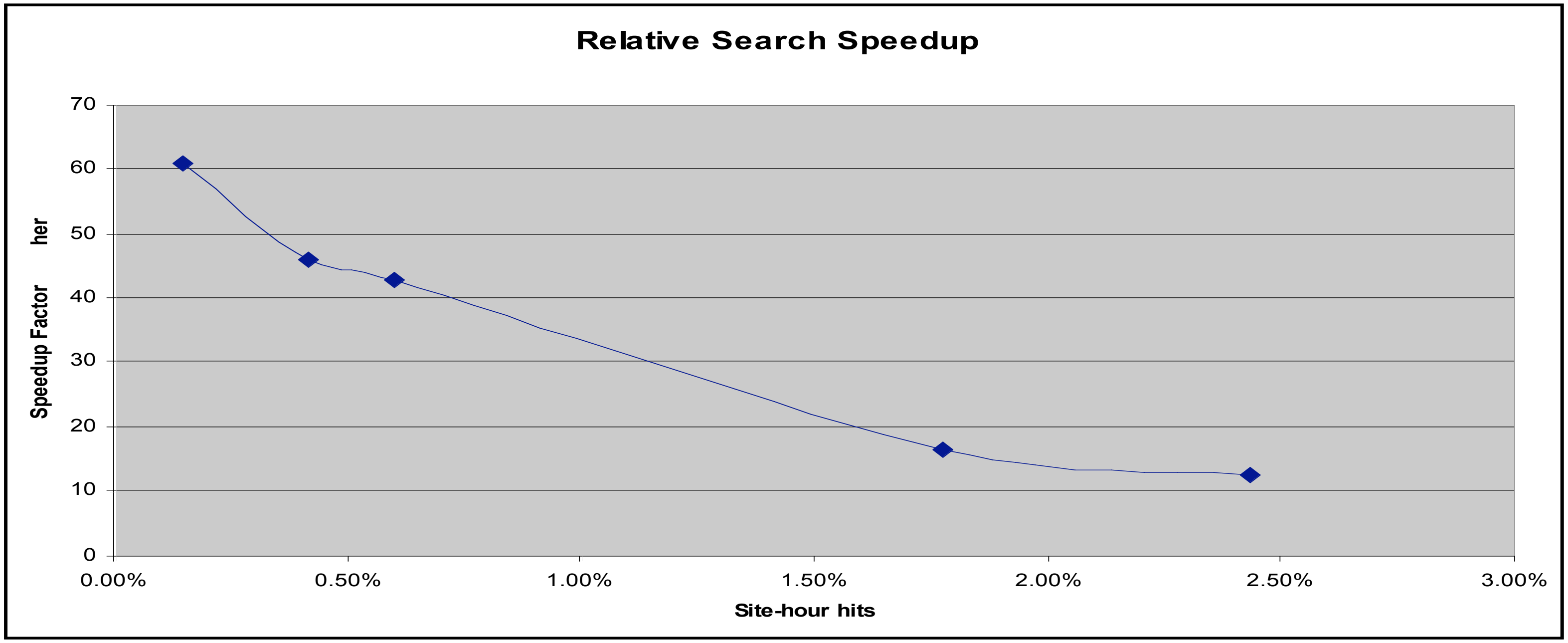
Typical analyst IP-based queries:

IPs searched	Date-range searched	Site-hours searched	Site-hour hits	% Site-hour hits	Session hits	Raw biflow file hits	Search time (conventional)	Search time (bloomdex)	Relative Speedup
8	12/13/06 - 1/9/07	19,140	466	2.43%	10,594	600	**16.29 hours**	**1.3 hours**	**12.5**
13	10/15/06 - 1/17/07	65,888	1,166	1.77%	158,345	1,667	**57.52 hours**	**3.45 hours**	**16.7**
13	1/22/07 - 1/29/07	4,959	31	0.60%	78	39	**4.16 hours**	**5.82 minutes**	**42.9**
4	1/1/07 - 1/2/07	725	3	0.41%	3	3	**21.5 minutes**	**28 seconds**	**46.1**
9	1/23/07 - 1/24/07	725	1	0.14%	1	1	**41.7 minutes**	**41 seconds**	**61**

- Expect >10x speedup
- Strong dependency on site-hour hit ratio
- Future optimizations to search tools could make it even faster

Bloomdex: Performance Profile

- Comparative Performance:



□ **Strong relationship between speedup and site-hour hit ratio**

□ **Ideal for searches on sparsely-occurring IPs**

Bloomdex: Performance Profile

- Bloom filter generation performance:

  - **Average site-day filter generation rate:**
    - ~ 33/hour = 792/day (current incoming rate: 29/day)

  - **Average site-hour filter generation rate:**
    - ~ 390/hour = 9360/day (current incoming rate: 696/day)

<u>Will scale well to 100+ sites (cheaply)</u>

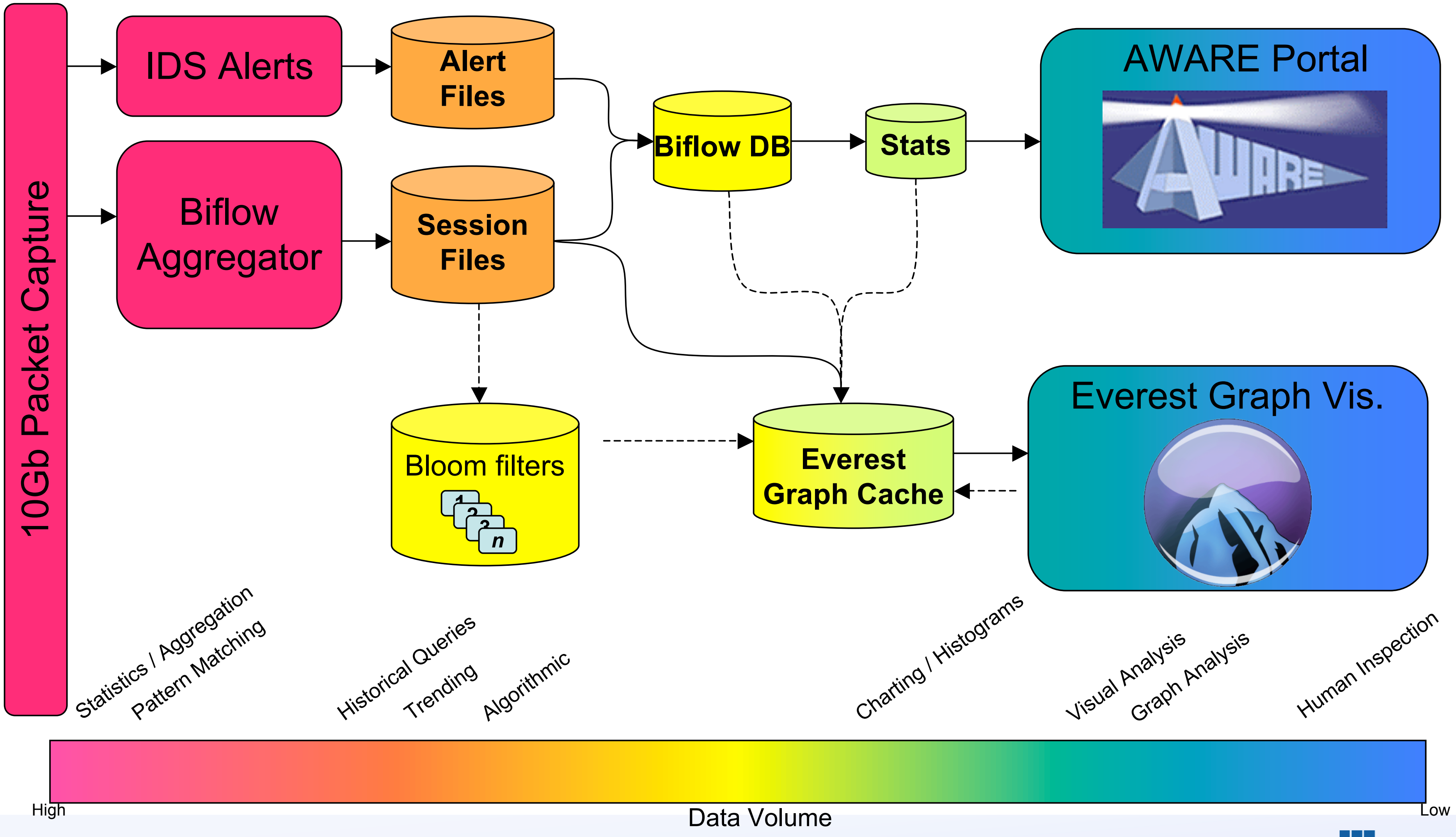
Bloomdex: Status

- **Coverage**
  - 2.5 years of biflow records indexed.
- **Storage footprint**
  - 3 bytes per unique IP at the site-hour, site-day and site-month levels.
  - Bloom filters currently using ~200GB of shared storage.
- **Exploring additional space and performance-based optimizations**
  - Other dimensions (e.g., port, ip-port, srcip-dstip pairs)
  - Counting Bloom filters
  - Different hashing functions
  - Parallelization

Bloomdex:

Analyst Workflow Integration

Analyst Workflow Integration



Facilitating Incident Analysis with Bloomdex and Everest Flow Visualization

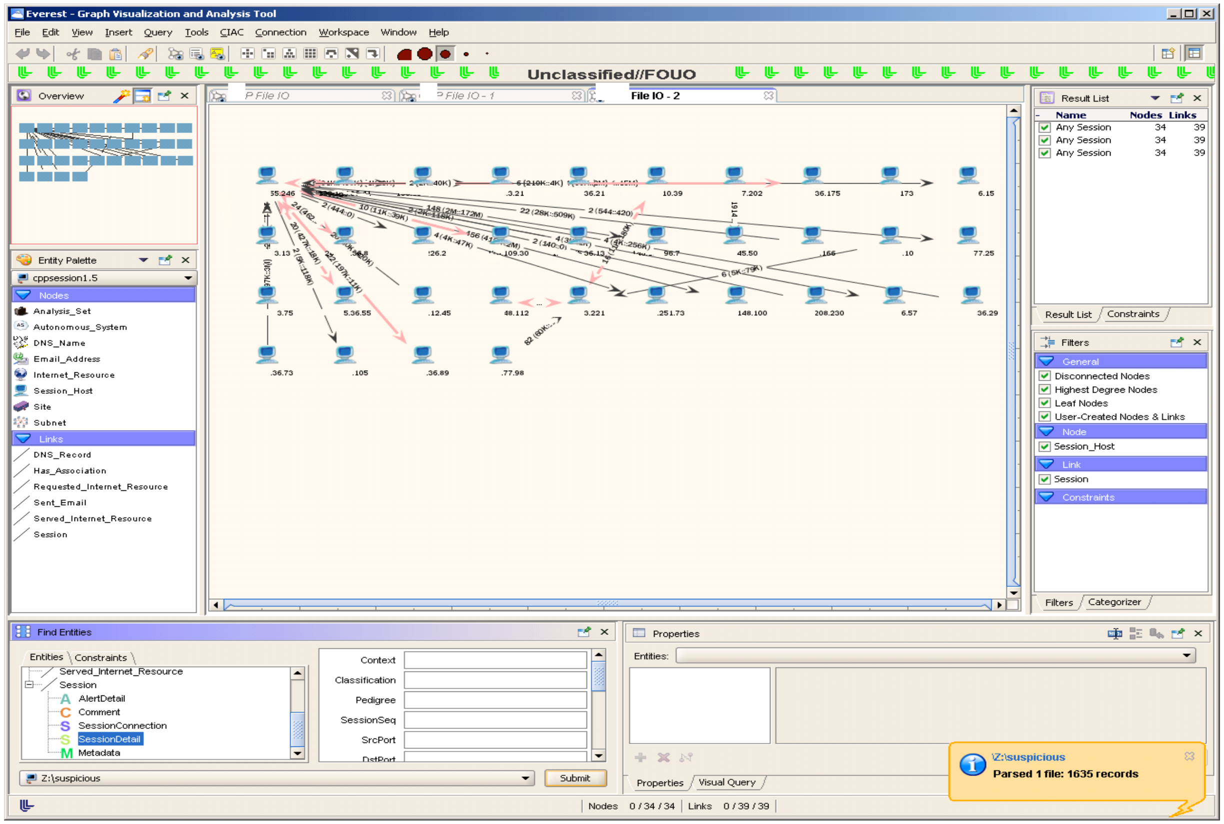
- **Example Use Scenario:**

1. Site reports compromise
   - Supplies <u>4 suspect IPs</u> to CIAC.

2. CIAC queries biflow data for suspect IPs using *Bloomdex* query tool:
   - Search all sensors over a sufficient time range (perhaps a full year).
   - Quickly identify several other sites with hosts exhibiting similar behaviors.
   - <u>Analysis set narrowed down to just 1,635 sessions.</u>

Analysis Using Bloomdex and Everest (2)

3. Launch Everest graph visualization tool, point to *Bloomdex* output file containing result set (1,635 biflow records).
4. Issue general query to generate session graph:



Everest - Graph Visualization and Analysis Tool

Unclassified//FOUO

Result List

Name	Nodes	Links
☑ Any Session	34	39
☑ Any Session	34	39
☑ Any Session	34	39

Filters

General
- ☑ Disconnected Nodes
- ☑ Highest Degree Nodes
- ☑ Leaf Nodes
- ☑ User-Created Nodes & Links

Node
- ☑ Session_Host

Link
- ☑ Session

Constraints

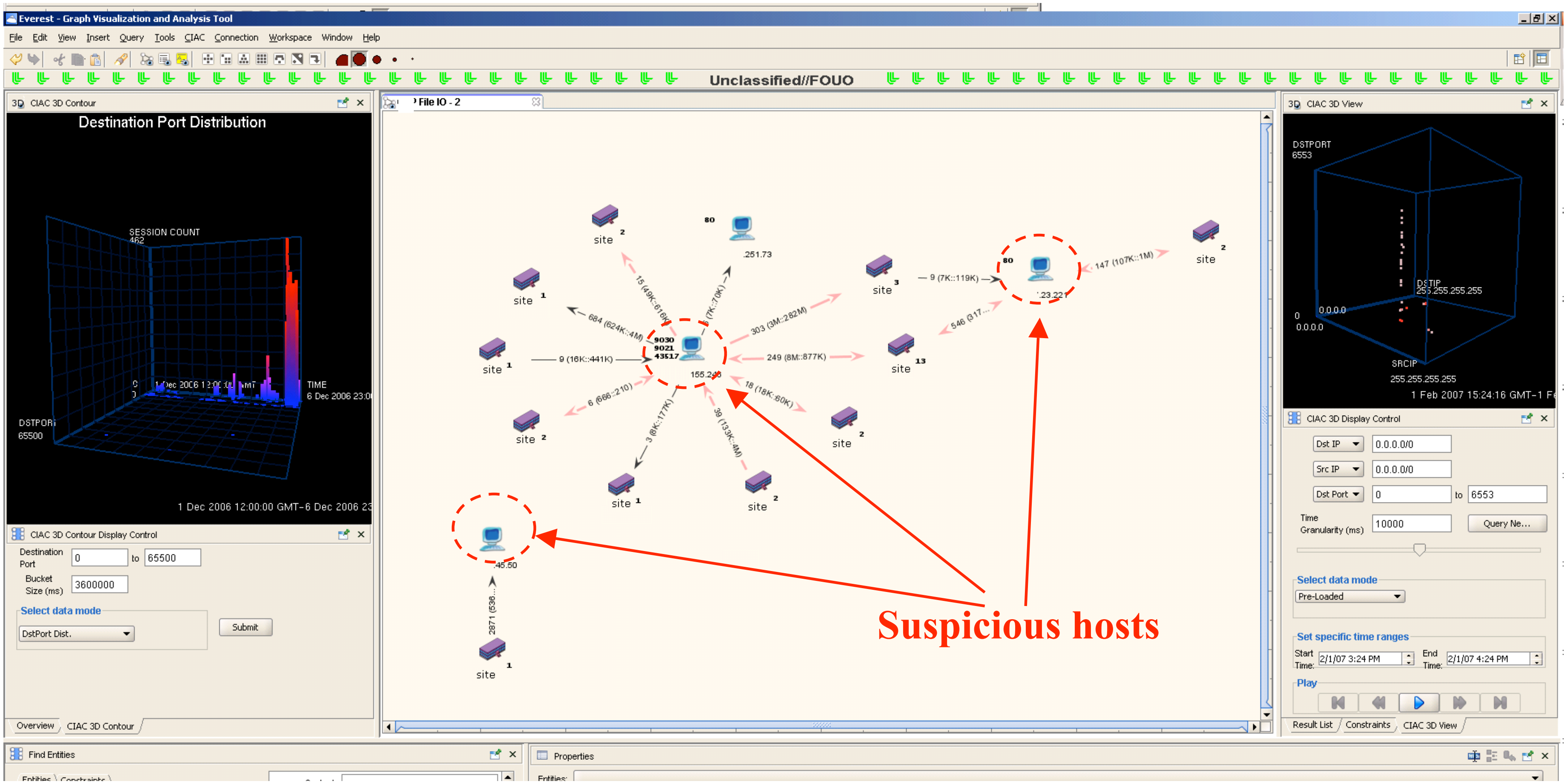
Z:\suspicious

Parsed 1 file: 1635 records

Nodes 0 / 34 / 34 Links 0 / 39 / 39

Analysis Using Bloomdex and Everest (3)

5. Perform drill-down or aggregate analysis



CIAC 3D Contour Display Control

Destination Port: 0 to 65500

Bucket Size (ms): 3600000

Select data mode: DstPort Dist.

Submit

CIAC 3D Display Control

Dst IP: 0.0.0.0/0

Src IP: 0.0.0.0/0

Dst Port: 0 to 6553

Time Granularity (ms): 10000

Query Ne...

Select data mode: Pre-Loaded

Set specific time ranges

Start Time: 2/1/07 3:24 PM

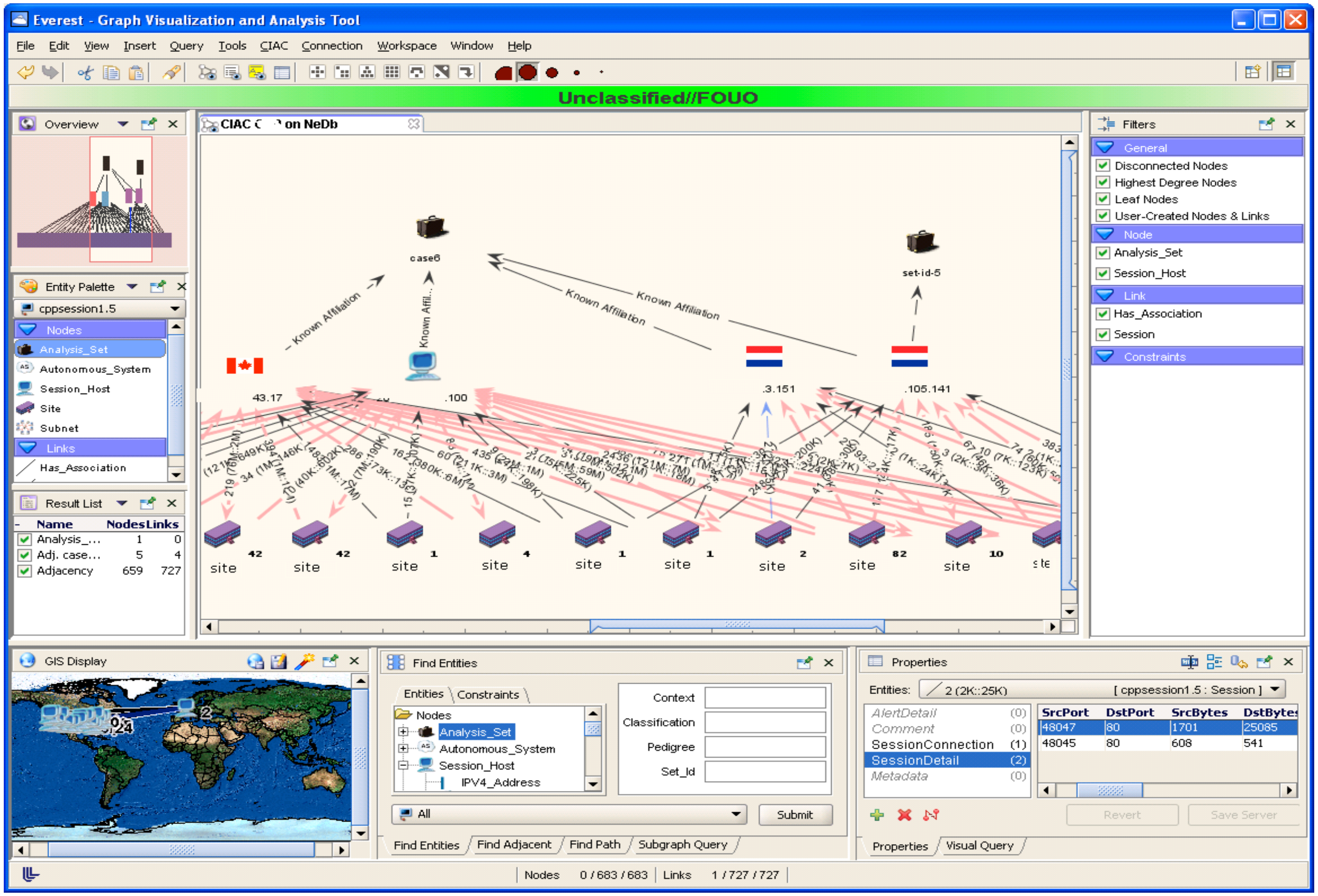
End Time: 2/1/07 4:24 PM

Play

Suspicious hosts

Analysis Using Bloomdex and Everest (4)

6. Perform in-depth or summary analysis



Filters

General
- ☑ Disconnected Nodes
- ☑ Highest Degree Nodes
- ☑ Leaf Nodes
- ☑ User-Created Nodes & Links

Node
- ☑ Analysis_Set
- ☑ Session_Host

Link
- ☑ Has_Association
- ☑ Session

Constraints

Result List

	Name	Nodes	Links
☑	Analysis_...	1	0
☑	Adj. case...	5	4
☑	Adjacency	659	727

Find Entities: Entities / Constraints; Nodes: Analysis_Set, Autonomous_System, Session_Host, IPV4_Address

Context:
Classification:
Pedigree:
Set_Id:

All — Submit

Properties

Entities: 2 (2K::25K) [cppsession1.5 : Session]

AlertDetail	(0)
Comment	(0)
SessionConnection	(1)
SessionDetail	(2)
Metadata	(0)

SrcPort	DstPort	SrcBytes	DstBytes
48047	80	1701	25085
48045	80	608	541

Revert — Save Server

Nodes 0 / 683 / 683 | Links 1 / 727 / 727

Conclusion

- **The *Bloomdex* suite enables significantly faster turnaround times on analyst IP-based queries:**
  - It does this by drastically narrowing the search space through Bloom filter pre-queries.
  - Facilitates use of other analytic tools, such as Everest.
  - Provides significant space savings.
  - **Very straightforward and inexpensive to deploy and maintain.**

- **Future**:
  - Utilize compressed bitmap indexes as an integrated indexing/retrieval solution.

Questions

cdr [at] llnl.gov