Toward Efficient Querying of Compressed Network Payloads

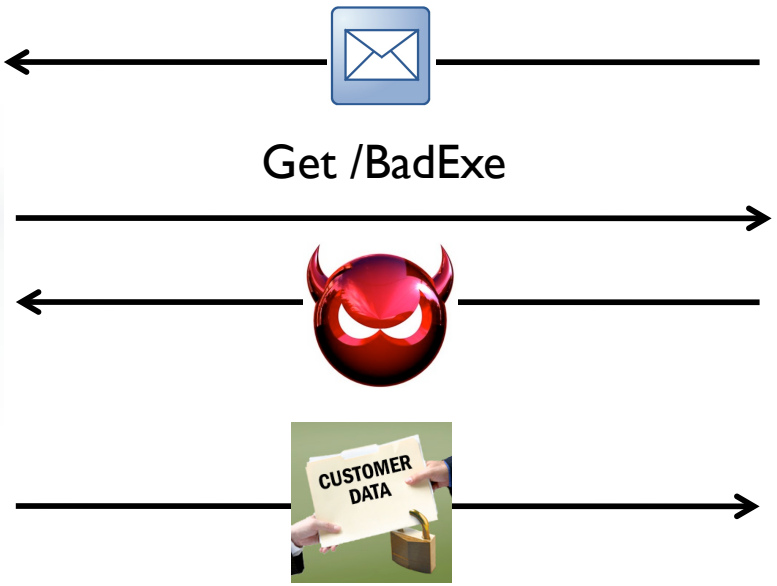
By Teryl Taylor and Fabian Monrose
University of North Carolina at Chapel Hill
Scott E. Coull and John McHugh
RedJack



THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

Motivation







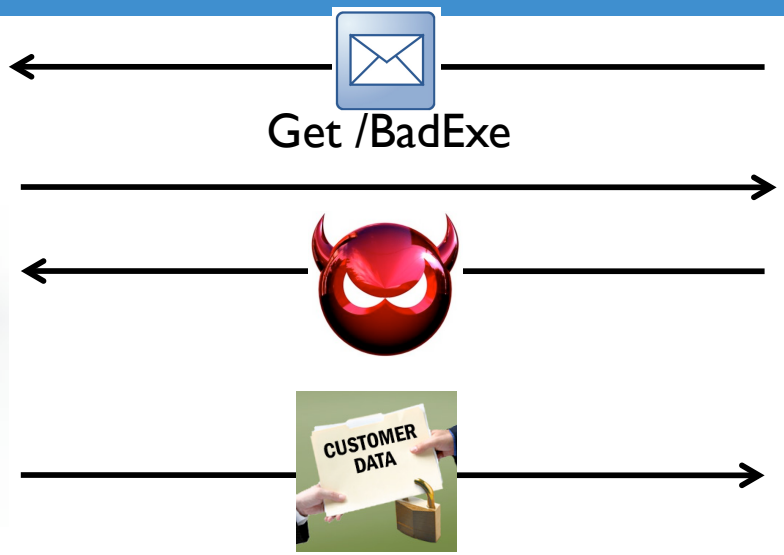
Please Log onto
http://cs.unnc.edu and change your password



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Motivation







Problem: How does an analyst perform efficient forensic analysis using all data available to him/her? Time is of the essence.



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Goal and Challenges

Goal: Build an interactive query system for network traffic which supports payload-based queries.

Challenges:

- Large volume of network traffic: billions of records (TBs of data).
- Network data has high dimensionality – i.e. lots of attributes.
- Payloads are heterogeneous.



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Existing Approaches

- Scan and Filter approaches (e.g. TCPDump).
  - Write optimized ☺.
  - Must scan each record in the dataset ☹.

- Parallel Scan and Filter approaches (e.g. MapReduce).
  - Requires significant computational resource ☹.
  - Google: Data organization is important for speed ☹.

- Relational Databases (row-wise) (e.g. Postgres).
  - Rigid structure ☹.
  - Slow/bloated index generation ☹.



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Approach

- Build a low I/O bandwidth (read-optimized) storage and query framework.

- Utilizing the following principals:
  - Reduce data footprint: aggregation and compression.
  - Only touch data relevant to the query: indexing.
  - Reduce disk I/O: horizontal and vertical partitioning.
  - Support arbitrary application protocols: flexible schemas.



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

```
SELECT SourceIP, Dns.Query.Type
WHERE
  DestPort = 53 AND
  Payload.Dns.Query.Domain = 'www.uunc.com'
```



Locate Data Partitions



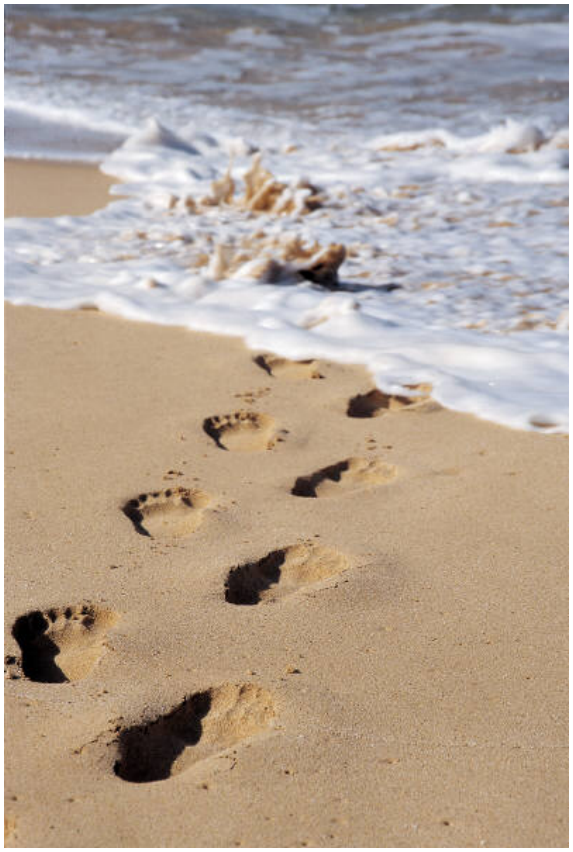
Locate Records



Return Records

Approach

Reduce Data Footprint….



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Record Aggregation on Flows

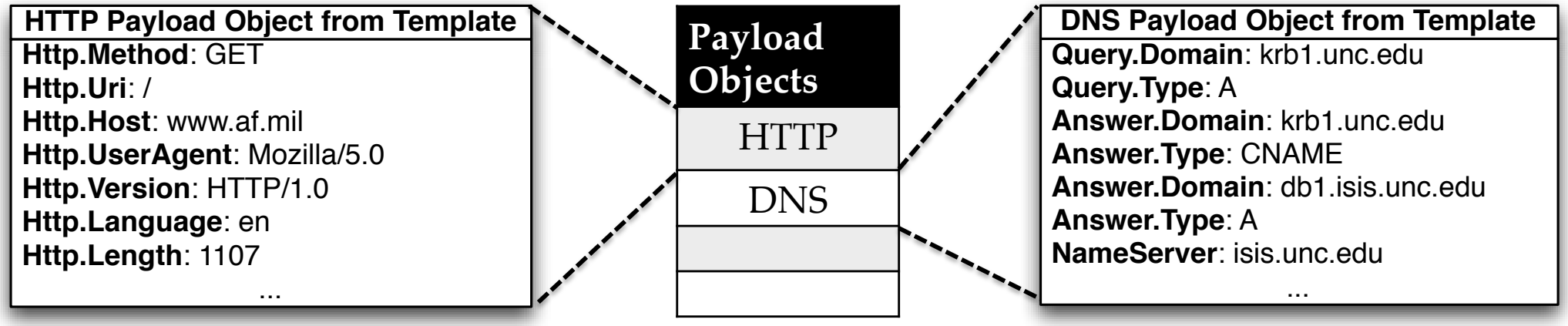
Src IP	Dest IP	Src Port	Dst Port	Protocol
10.10.4.2	13.21.7.6	13	80	TCP
96.21.7.5	43.89.2.1	53	53	UDP

- Low bandwidth I/O ☺.
- Special read-only bitmap indices ☺.



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Payload Summary Objects



- Payload objects support flexible schema.
- Attributes are extracted using deep packet inspection.
- Each object has a template.
- Template instantiated and filled with attributes.



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Joining Flows and Payloads

Column Oriented Flow Records + Serialized Payload Objects

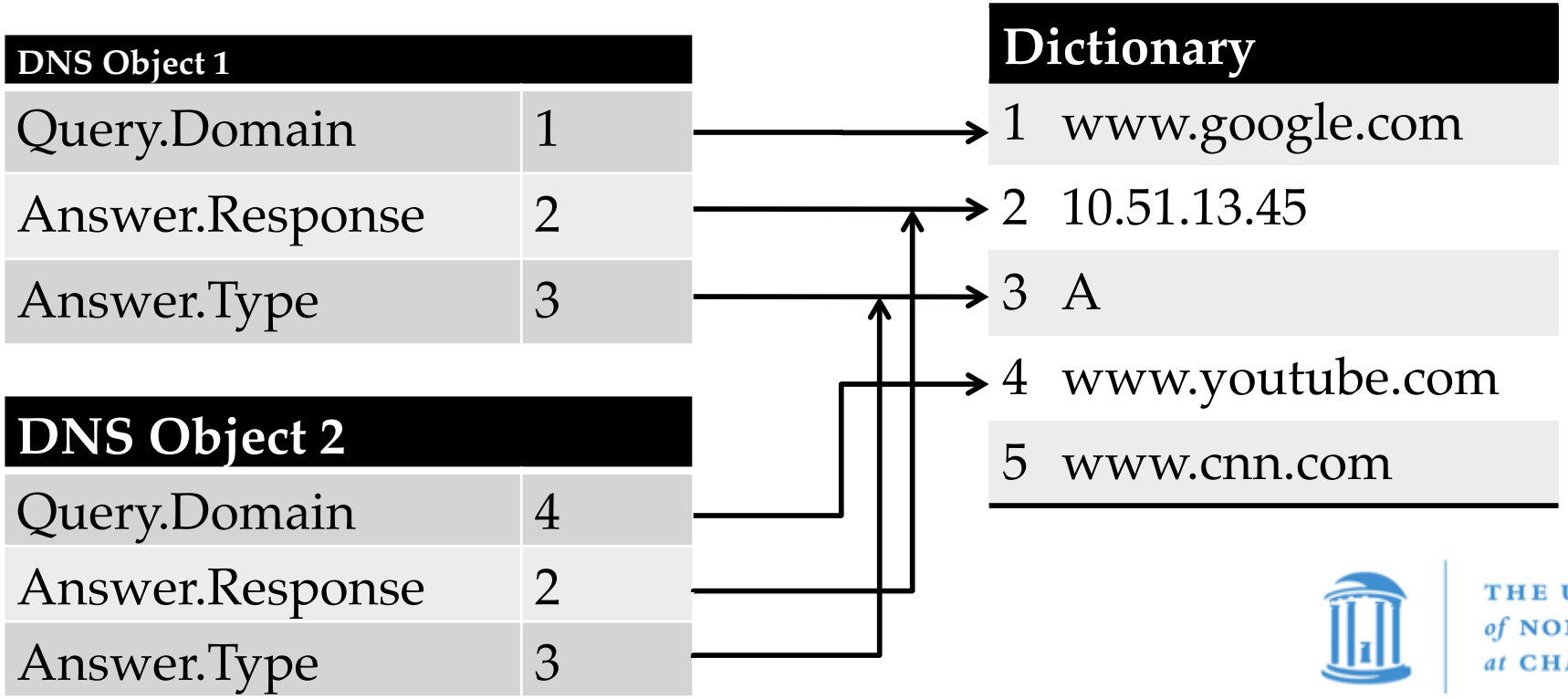
Src IP	Dest IP	Src Port	Dest Port	Protocol	Payload Objects
10.10.4.2	13.21.7.6	13	80	TCP	HTTP
96.21.7.5	43.89.2.1	53	53	UDP	DNS



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Payload Compression

- Two forms of compression
  - Record (object) level dictionary compression.
  - Block-level compression using LZO.



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Approach

Reduce Disk I/O….





THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Partitioning the Data

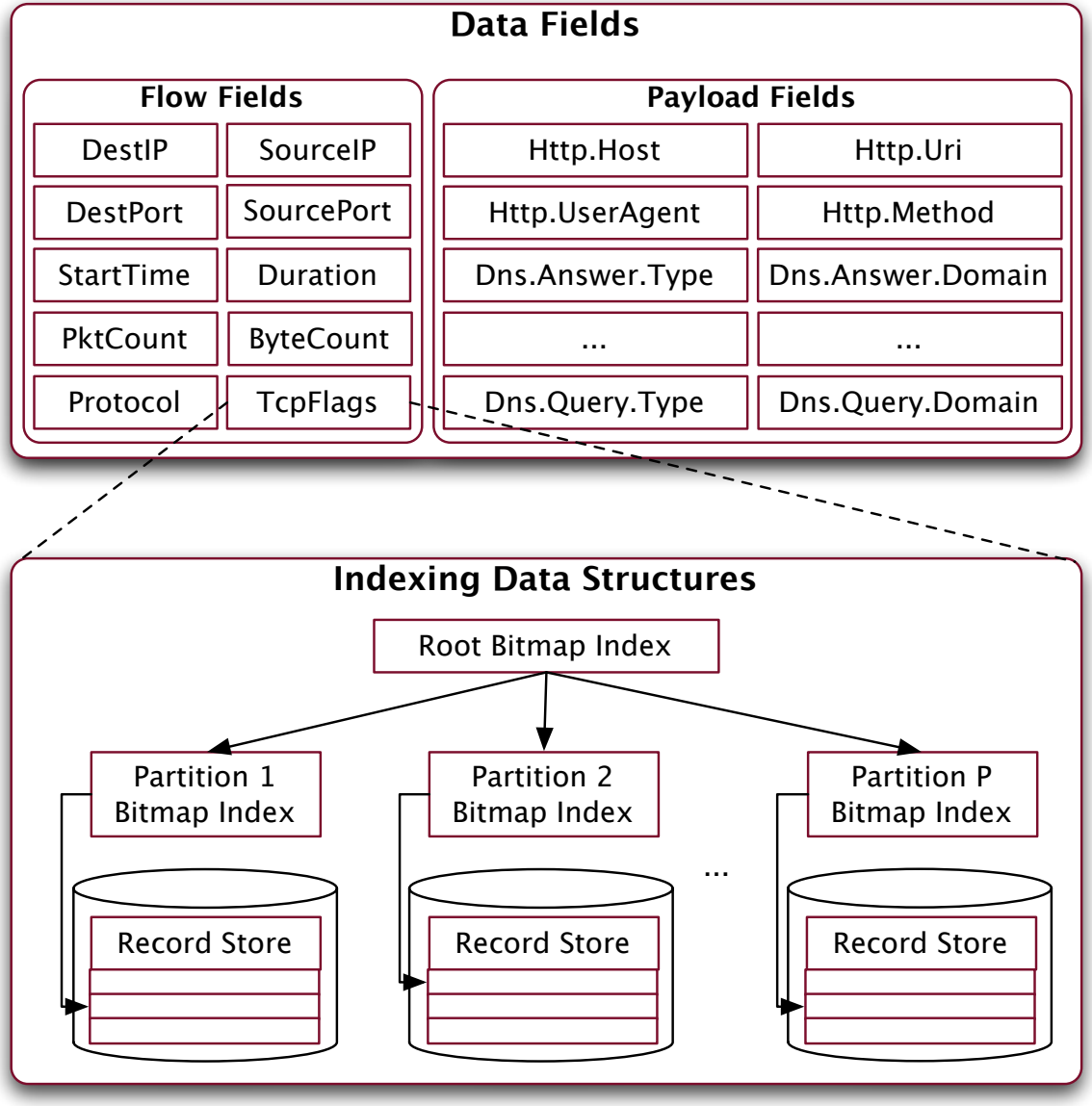
- Records grouped (partitioned) into configurable sizes.
- Records are added to partition as they expire.
- Partitions are independent of each other.

- Advantages:
  - Allows us to access only relevant partitions.
  - Enables data distribution to support query parallelization.



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Select Matching Partitions with Root Indices



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Approach

Only Load Relevant Records….





THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

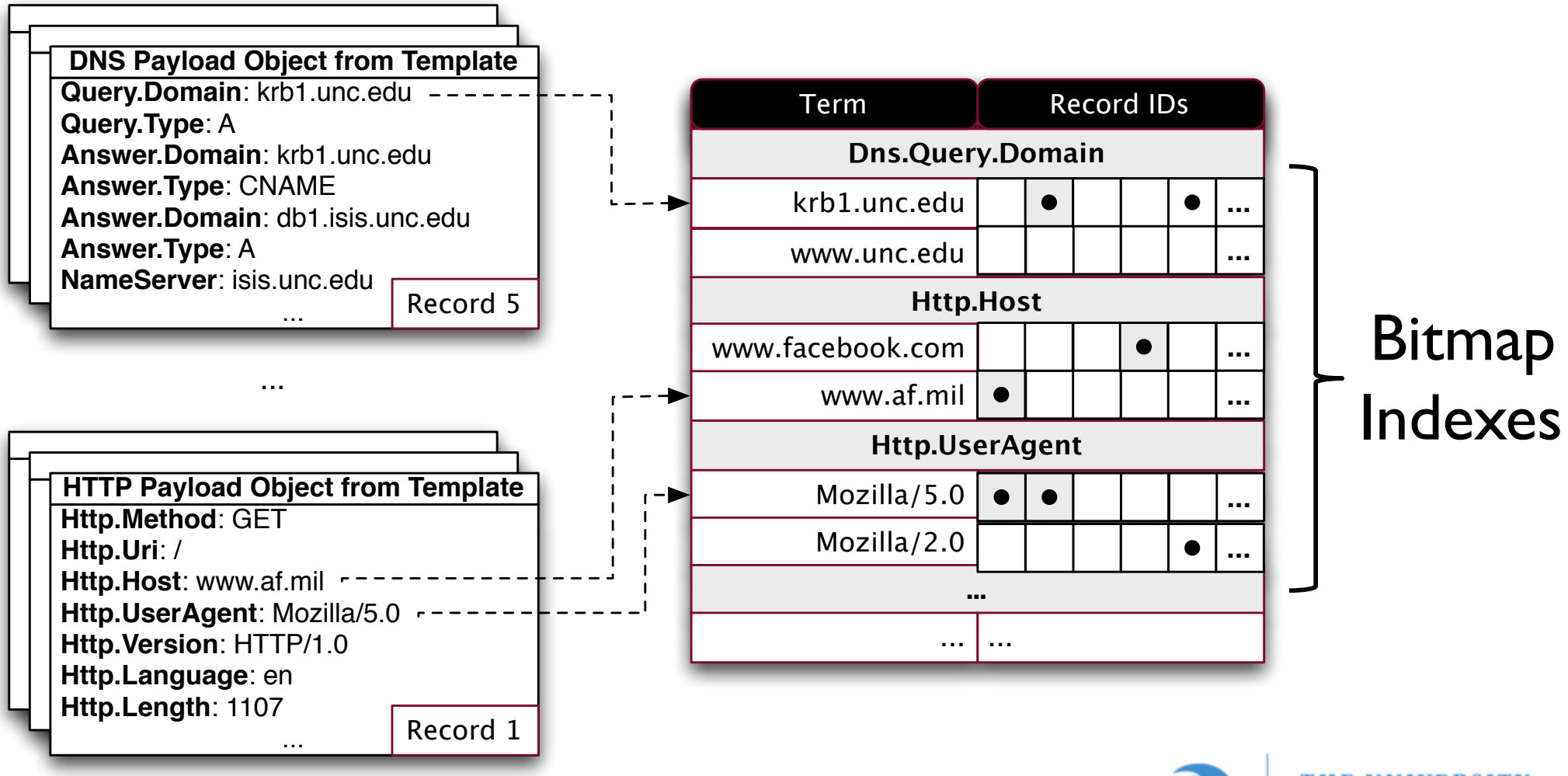
Partition Indices

- Indices important to avoid scan and filter.
- Each flow-based attribute has a bitmap index.

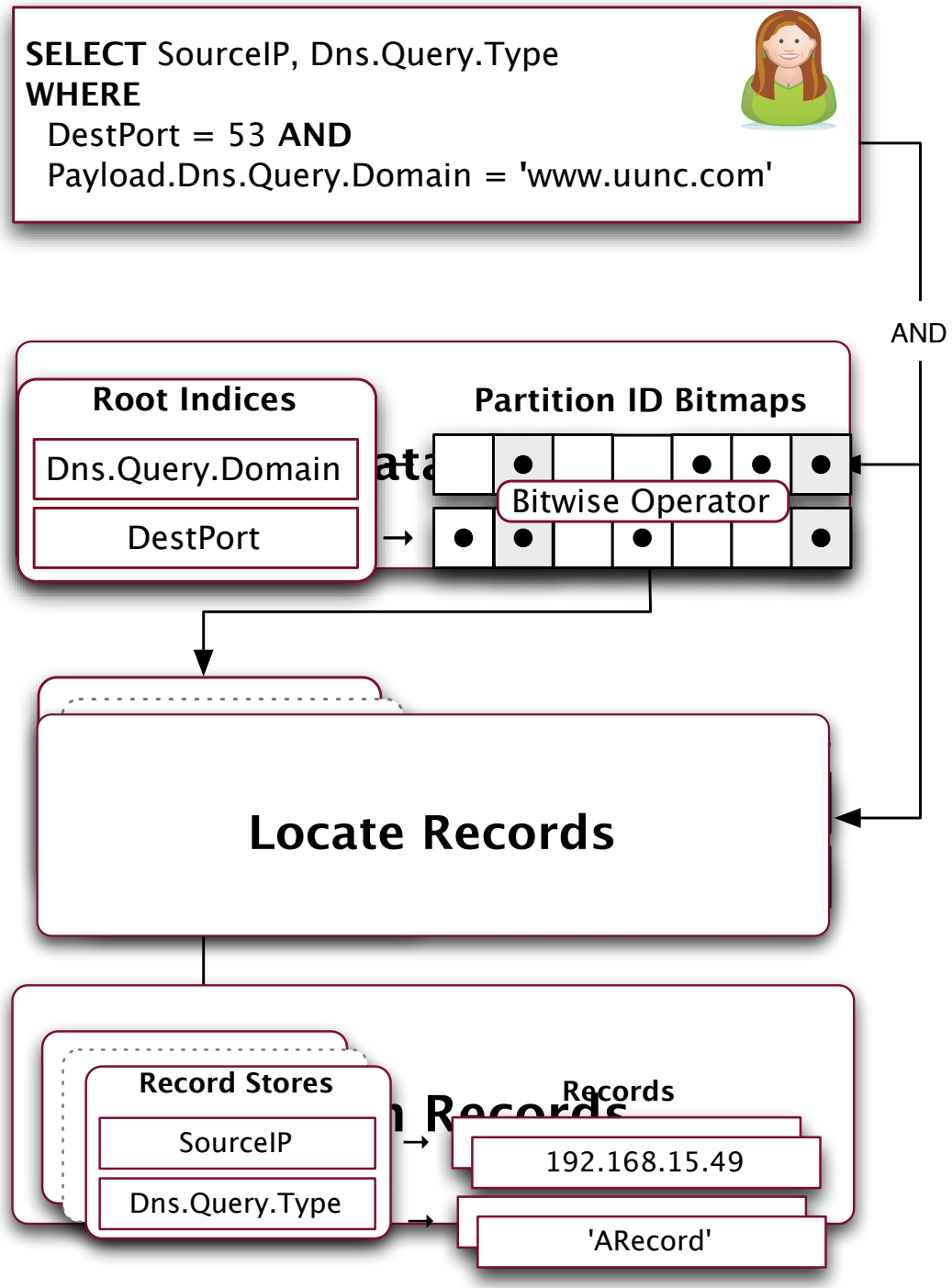


THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Payload Indices



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL


SELECT SourceIP, Dns.Query.Type
WHERE
DestPort = 53 AND
Payload.Dns.Query.Domain = 'www.uunc.com'
AND
Root Indices
Dns.Query.Domain
DestPort
Partition ID Bitmaps
Bitwise Operator
Locate Records
Record Stores
SourceIP
Dns.Query.Type
Records
192.168.15.49
'ARecord'

Evaluation on Campus DNS Data and DNS/HTTP Data

- Dataset 1: Campus DNS – 5 day
  - Number of Connections: 325 million
  - Original Raw Trace: 122 GB
  - Data Store (uncompressed): 155 GB
  - Data Store (dictionary compressed): 83 GB
  - Data Store (dictionary + LZO compressed): 75 GB

- Dataset 2: Campus DNS/HTTP – 2.5 hours
  - Number of Connections: 11.1 million
  - Original Raw Trace: 400 GB
  - Data Store (dictionary + LZO compressed): 12 GB

*Dual Intel Xeon 2.27 GHZ, 12 GBs RAM, single 2TB 7200 RPM local SATA Drive

Evaluation Queries

- Heavy Hitter Queries: return the majority of records.
- Partition Intensive Queries: return a few records from every partition.
- Needle in a Haystack Queries: return a few records.

Reiss et. al, Efficient Analysis of Live and Historical Streaming Data and its application to Cybersecurity, 2006.



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Flow-based Query Comparison

Flow-based Queries	Heavy Hitters	Partition Intensive	Needle Haystack
Postgres	18.1m	9.5m	2.0s
SiLK	1.8m	1.8m	1.8m
Our approach	2.5s	30.5s	0.04s

e.g., SELECT source_IP, destination_IP from all_traffic where protocol=UDP

* DNS dataset

Payload-based Query Comparisons with Relational Joins

Payload-based Queries (joins)	Heavy Hitters	Partition Intensive	Needle Haystack
Postgres	> 2h	> 2h	3s
Our approach	30m	2.5m	0.1s

e.g., SELECT dns.query.type, dns.ns.ttl FROM all_traffic WHERE dns.queryDomain=www.facebook.com AND dns.ns.domain=ns.facebook.com.

* DNS dataset

Single Table Payload-based Query Comparisons

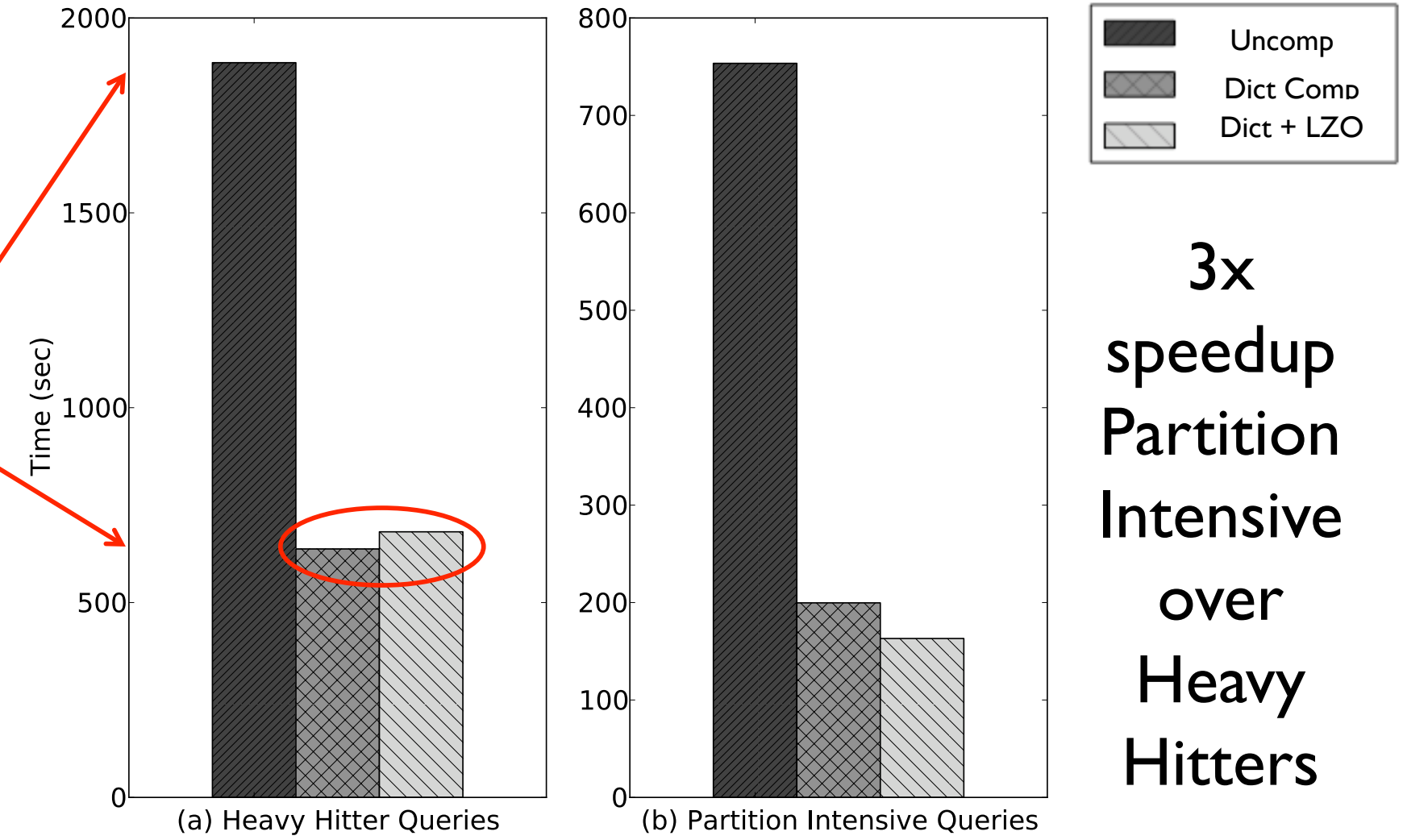
Payload-based Queries (no-joins)	Heavy Hitters	Partition Intensive	Needle Haystack
Postgres	7.6m	9.7m	1.6s
Our approach	9.7m	3.3m	0.1s

e.g., SELECT dns.queryType FROM all_traffic WHERE dns.queryDomain=www.google.com OR dns.queryDomain= www.facebook.com

* DNS dataset

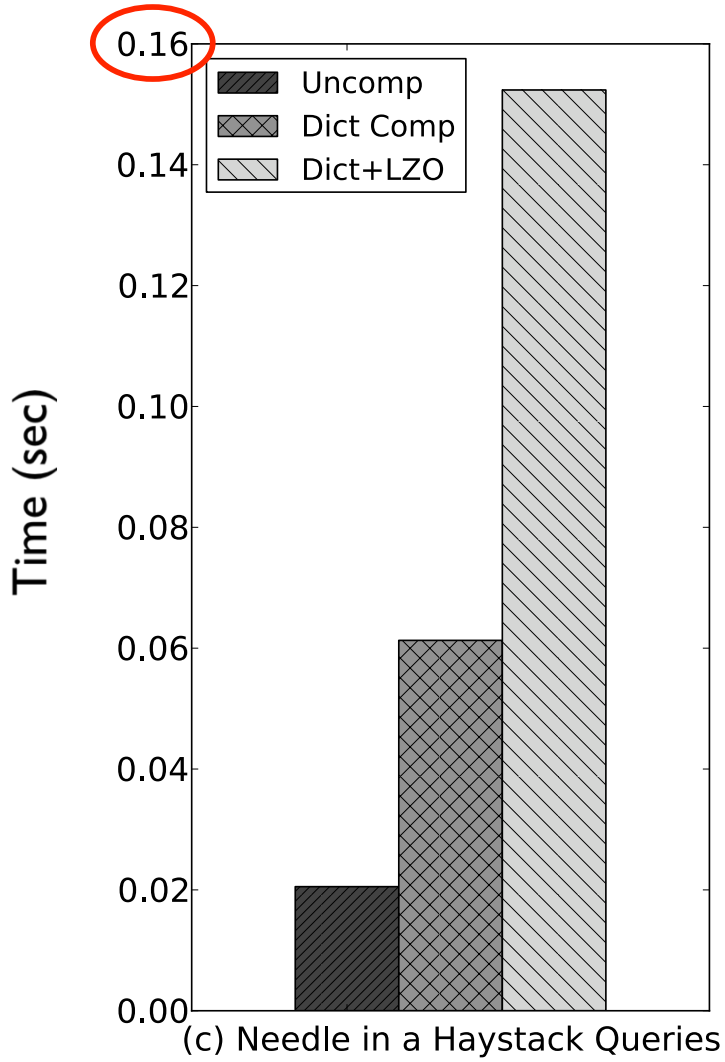
Impact of Payload Compression

>3x Speed Gain



3x speedup Partition Intensive over Heavy Hitters

Impact of Payload Compression



LZO block must be decompressed before query

Summary

- Framework has been used recently to:
  - Finding DGA domains – including Cridex malware.
  - Blacklisted domains.
  - Fast-Flux analysis.
  - Finding correlations between HTTP and DNS requests.

- Sub minute flow query times.
- Sub second needle in a haystack queries.
- Partition Intensive queries on the order of a few minutes.
- Slow linear growth over time.



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Conclusions

- Designed a low bandwidth, random access query framework for network traffic.
- Improved query performance over existing solutions.

- Take Aways:
  - Record-level compression provides significant data reduction and improved query performance.
  - Limit object parsing for performance
  - Data organization important for data reduction,
  - Multi-level indexing improves query performance through I/O reduction.



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL