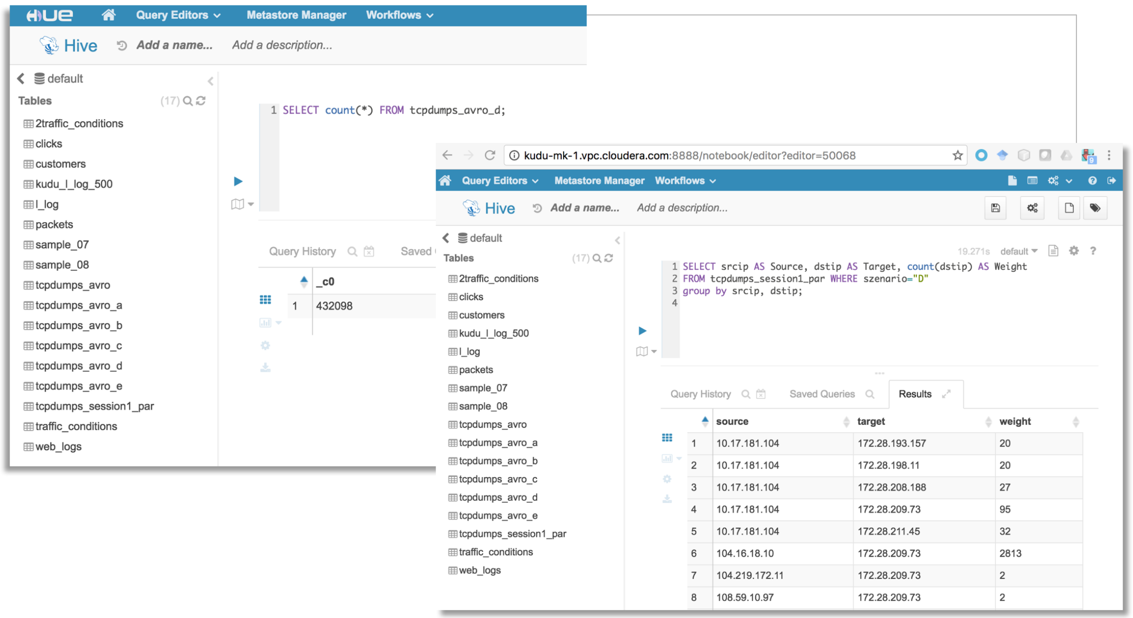
Network Traffic Analysis & Cluster Analysis

Exploring Hadoop Clusters using Free Tools



- 2traffic_conditions
- clicks
- customers
- kudu_l_log_500
- l_log
- packets
- sample_07
- sample_08
- tcpdumps_avro
- tcpdumps_avro_a
- tcpdumps_avro_b
- tcpdumps_avro_c
- tcpdumps_avro_d
- tcpdumps_avro_e
- tcpdumps_session1_par
- traffic_conditions
- web_logs

```
SELECT count(*) FROM tcpdumps_avro_d;
```

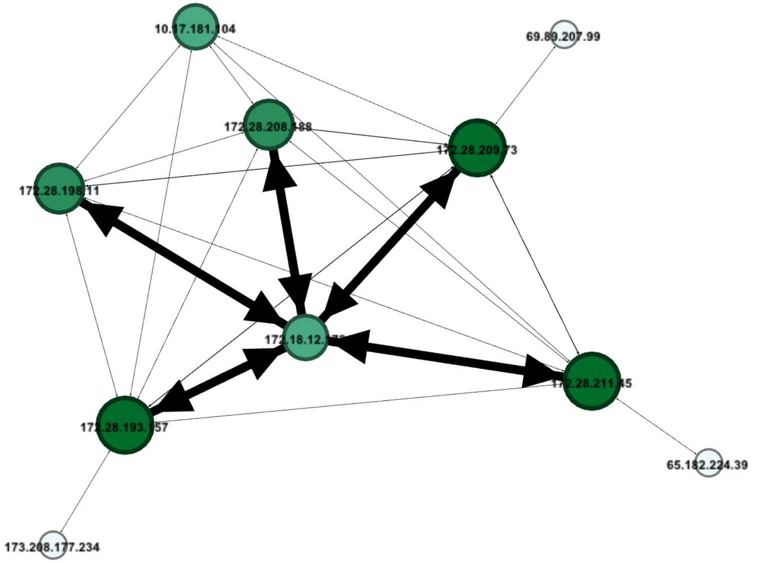
	_c0
1	432098

kudu-mk-1.vpc.cloudera.com:8888/notebook/editor?editor=50068

- 2traffic_conditions
- clicks
- customers
- kudu_l_log_500
- l_log
- packets
- sample_07
- sample_08
- tcpdumps_avro
- tcpdumps_avro_a
- tcpdumps_avro_b
- tcpdumps_avro_c
- tcpdumps_avro_d
- tcpdumps_avro_e
- tcpdumps_session1_par
- traffic_conditions
- web_logs

```
SELECT srcip AS Source, dstip AS Target, count(dstip) AS Weight
FROM tcpdumps_session1_par WHERE szenario="D"
group by srcip, dstip;
```

	source	target	weight
1	10.17.181.104	172.28.193.157	20
2	10.17.181.104	172.28.198.11	20
3	10.17.181.104	172.28.208.188	27
4	10.17.181.104	172.28.209.73	95
5	10.17.181.104	172.28.211.45	32
6	104.16.18.10	172.28.209.73	2813
7	104.219.172.11	172.28.209.73	2
8	108.59.10.97	172.28.209.73	2



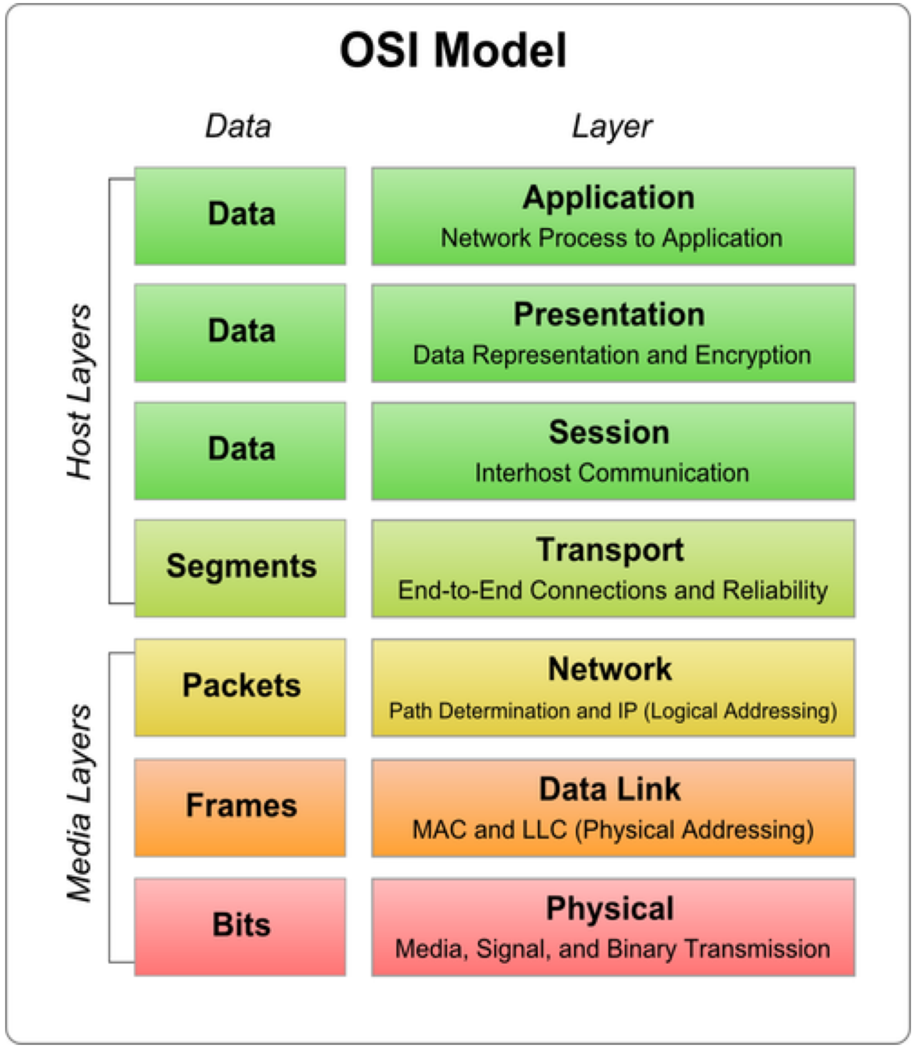
Background and Goals:

- **Apache Spot** was started recently
  - DNS, Netflow, PCAP data is analyzed
  - The goal is to identify:
    ”suspicous connections”
    or:
    “dangerous activity”.

- What is suspicious?
  - Apache Spot uses a topic-model approach, to classify traffic.

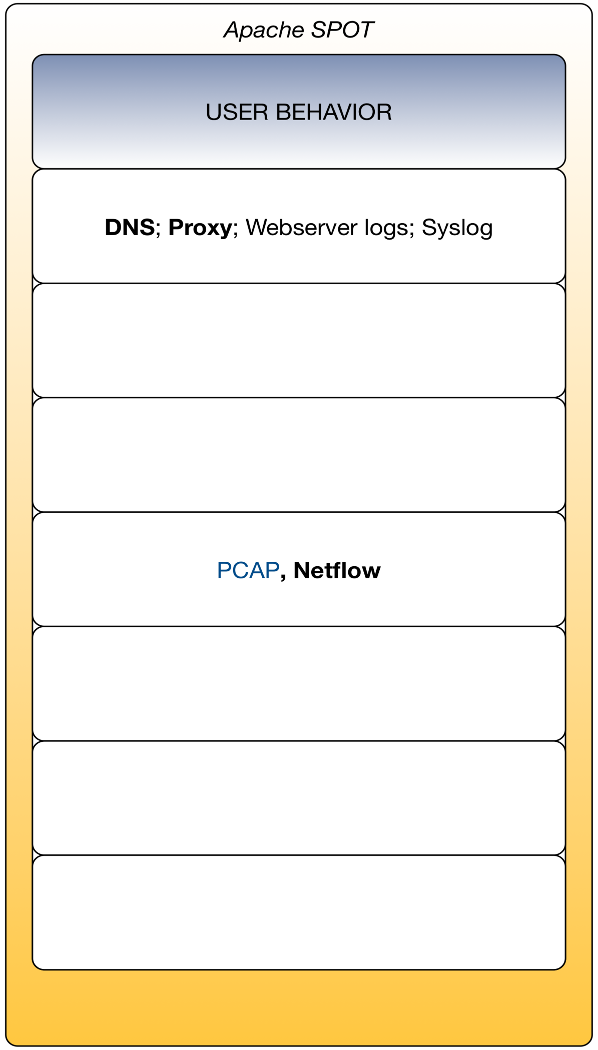
Used Raw Data:

https://en.wikipedia.org/wiki/List_of_network_protocols_(OSI_model)



OSI Model

	Data	*Layer*
Host Layers	**Data**	**Application** Network Process to Application
	Data	**Presentation** Data Representation and Encryption
	Data	**Session** Interhost Communication
	Segments	**Transport** End-to-End Connections and Reliability
Media Layers	**Packets**	**Network** Path Determination and IP (Logical Addressing)
	Frames	**Data Link** MAC and LLC (Physical Addressing)
	Bits	**Physical** Media, Signal, and Binary Transmission



Apache SPOT

USER BEHAVIOR

DNS; **Proxy**; Webserver logs; Syslog

PCAP, **Netflow**

Our Goals (midterm):

- Use *local context* information instead of *single package data* only.

  (A) Temporal communication networks

  (B) Vectorization of measured properties from multiple sources

- Consider additional communication layers:
  - Syslog
  - Webserver logs
  - Cloudera Manager events
  - Cloudera Navigator events

About Event Processing:

- **Kafka** gives an order only within a partition
  - Post-processing in Spark
- **HBase** sorts rows by key
  - Table design is now strictly time related, which is not a very universal approach.
- **Kudu** uses Primary Keys
  Each Kudu table must declare a primary key comprised of one or more columns. Primary key columns must be **non-nullable**, and may not be a boolean or floating-point type. *Every row in a table must have a unique set of values for its primary key columns.* As with a traditional RDBMS, primary key selection is critical to ensuring performant database operations.
  - ***But:*** *Events have timestamps which are* ***not*** *really unique !!!*

Our Activities

- Implement a data pipeline:
  - Kafka => Spark => HDFS => Notebook
  - Kafka => Spark => Kudu
  - Kudu => Spark => HDFS => (Notebook)

- Create reference data sets
  - Scenario A: Terrasort (Big-Batch-Workload)
  - Scenario B: HDFS PUT,GET; HUE (Interactive Workload)
  - Scenario C: Idle cluster (Vacation time)
  - Scenario D: Kafka => Spark => Kudu (Realistic production Workload)
  - Scenario E: Twitter => Spark => Kudu (Realistic production Workload)

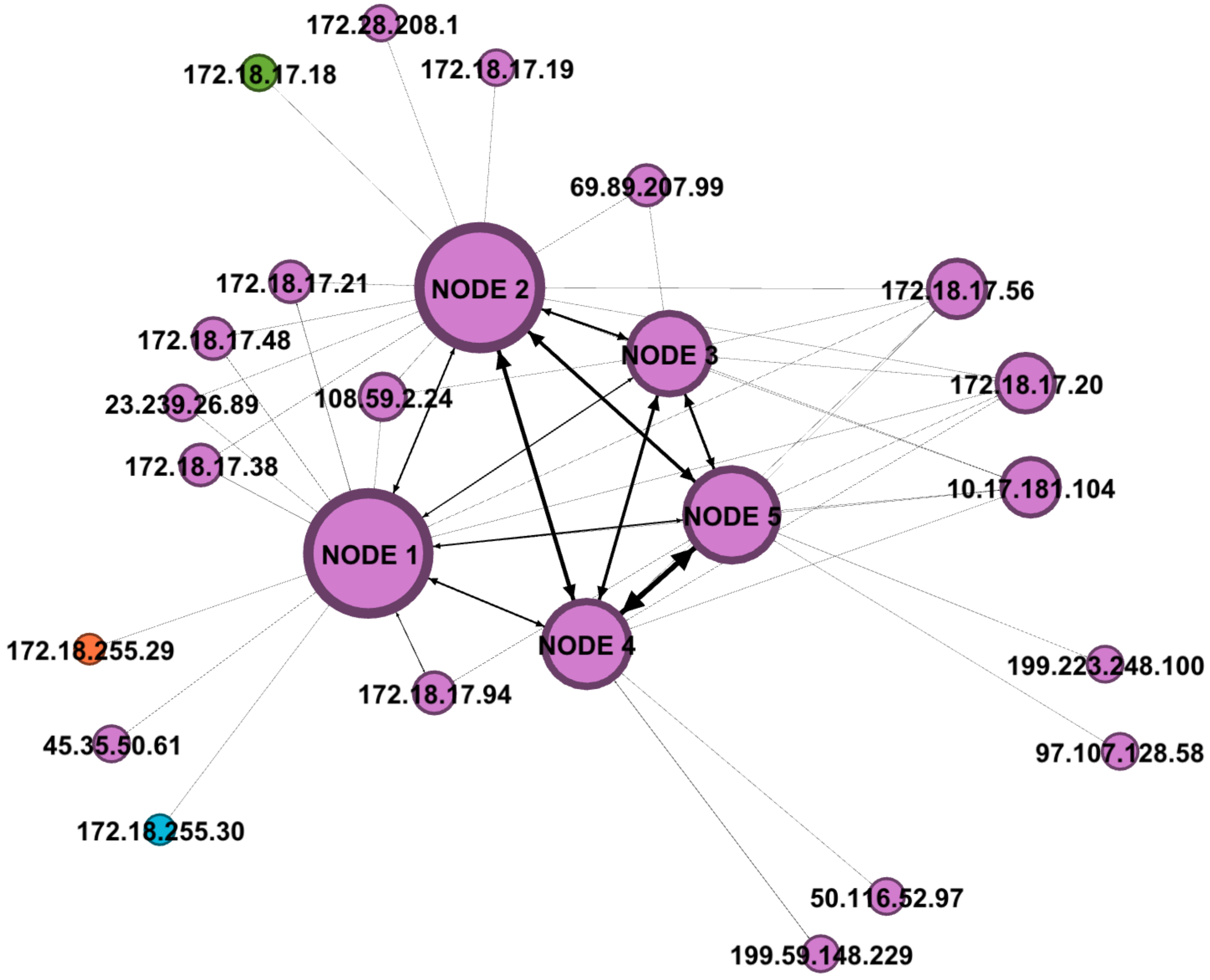
Results

- Scenario A: Batch workload
- Scenario D: External data acquisition
- Scenario E: Idle cluster

Scenario A:

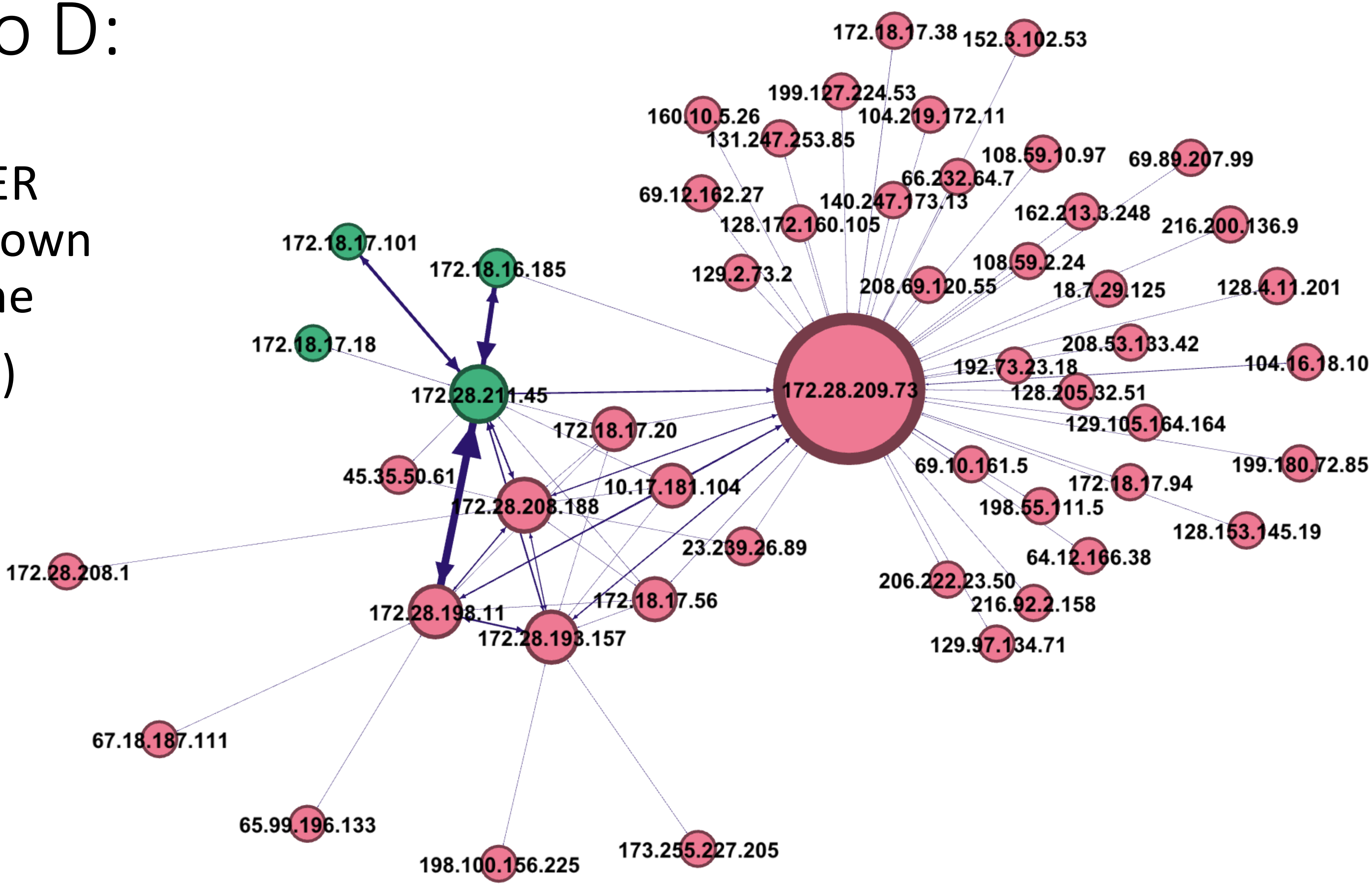
TERRAGEN

TERRASORT



Scenario D:

IDLE CLUSTER (some unknown activity in the background)



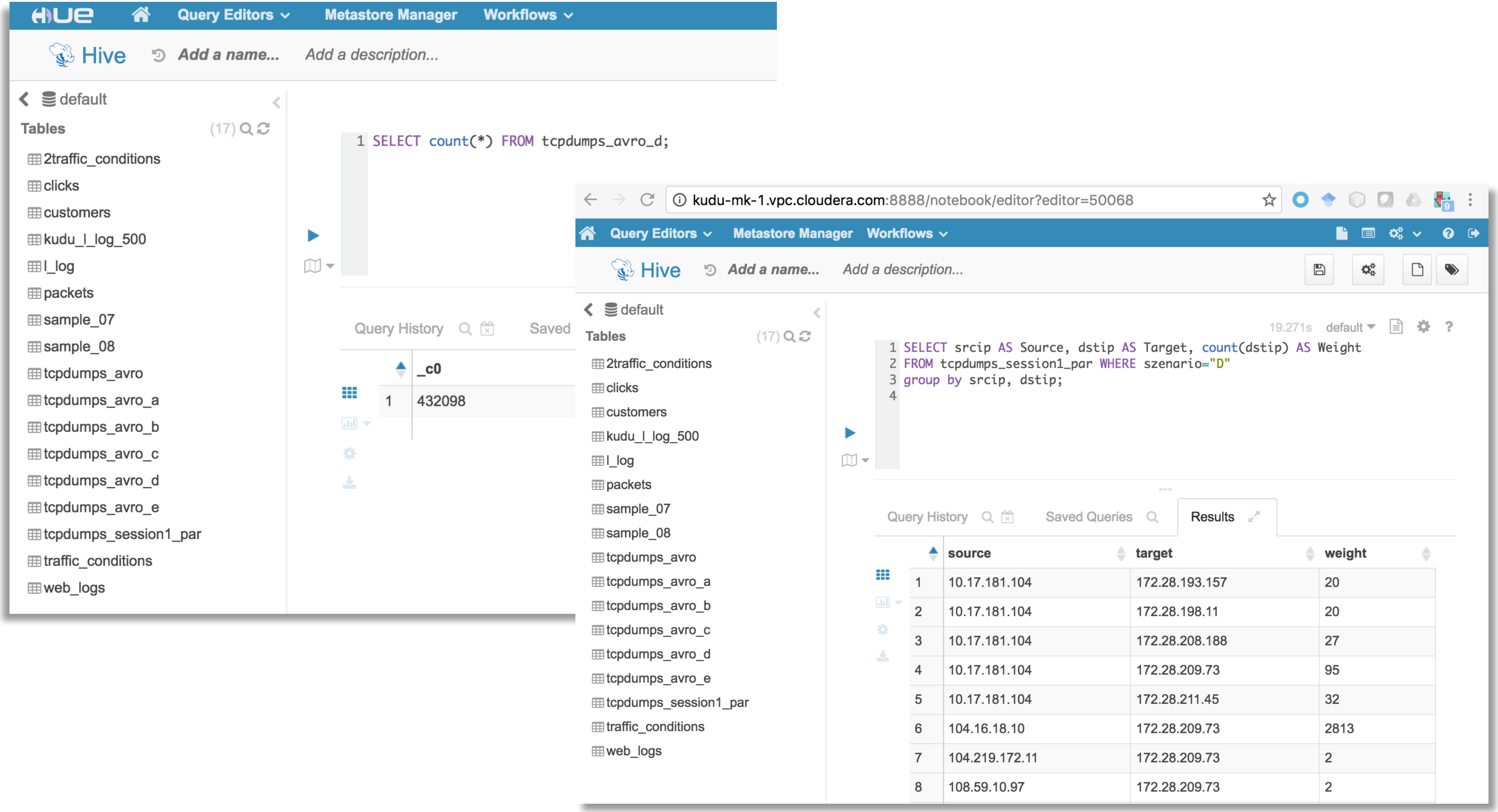
First Iteration:

- We organized our work in 3 phases:
  - Data and domain inspection + solution proposals
  - Environment setup
    - Tool centric: Jupyter, Eclipse, IntlliJ, CloudCat cluster, Git repository
    - Data centric:, Data collector tool, Demo data generation, Data formats
  - Data capturing and data generation
  - Analyzing the data in a well defined environment
- Results are available in Git repos:
  - http://github.mtv.cloudera.com/kamir/Snaffer
  - https://github.com/mbalassi/packet-inspector

- Increase functionality and knowledge by doing small iterations
- Share code and knowledge

How it works …

- We collect raw data in Avro format, using the Snaffer script.
- We transform the events to networks, using Hive.
- We analyze and visualize the networks using Gephi.



2traffic_conditions
clicks
customers
kudu_l_log_500
l_log
packets
sample_07
sample_08
tcpdumps_avro
tcpdumps_avro_a
tcpdumps_avro_b
tcpdumps_avro_c
tcpdumps_avro_d
tcpdumps_avro_e
tcpdumps_session1_par
traffic_conditions
web_logs

```
SELECT count(*) FROM tcpdumps_avro_d;
```

	_c0
1	432098

2traffic_conditions
clicks
customers
kudu_l_log_500
l_log
packets
sample_07
sample_08
tcpdumps_avro
tcpdumps_avro_a
tcpdumps_avro_b
tcpdumps_avro_c
tcpdumps_avro_d
tcpdumps_avro_e
tcpdumps_session1_par
traffic_conditions
web_logs

```
SELECT srcip AS Source, dstip AS Target, count(dstip) AS Weight
FROM tcpdumps_session1_par WHERE szenario="D"
group by srcip, dstip;
```

	source	target	weight
1	10.17.181.104	172.28.193.157	20
2	10.17.181.104	172.28.198.11	20
3	10.17.181.104	172.28.208.188	27
4	10.17.181.104	172.28.209.73	95
5	10.17.181.104	172.28.211.45	32
6	104.16.18.10	172.28.209.73	2813
7	104.219.172.11	172.28.209.73	2
8	108.59.10.97	172.28.209.73	2

Outlook

Entropy of Temporal Network

- Time evolution of the network properties
  - Topology
  - Topological node properties

Milestone One:

- Follow a common DSP model (data science process model)
- Use CDH default tools and gain experience
- Work with Kafka (for input) and Hive tables (for input and output)
- Implement a dataset profiling procedure, using Spark
- Present results, using Jupiter notebook
- Increase functionality and knowledge by doing small iterations
- Share code and knowledge

TODO (1)

- Define data sources according to inspection methods
- Define Avro schema and SOLR schema
- Automatic dataset initalization / validation

- DESCRIBE as WIKI and than instantiate via ANSIBLE

TODO (2)

- SNAProfiler
  - SQL for Network creation
  - Topology per time slice

- Envelop:
  - Allows us to hook in the SNAProfiler component as a JAR.

TODO (3)

- Time Slice Preparation
  - KAFKA => Hbase
  - App—controled time slice management:
    - (K,V) : (EXP_METRIC_TS, NETWORKDATA_as_edgelist)
  - Opposite to TIMESERIES presentation

References

- https://docs.google.com/document/d/12SHvTGJWtewk8CpUClOy22mh7cUow18F_Jg2ZNNE3h8/edit#heading=h.r4wlzr2ctack
- https://docs.google.com/document/d/1sD0_T2fQ7J5k7Ttx1vmAkYkMljMySgKFimm4hNVXxgA/edit#
- http://research.ijcaonline.org/volume74/number17/pxc3890233.pdf
- https://www.cs.princeton.edu/~blei/papers/BleiNgJordan2003.pdf