Big data Journey

From a pallet of parts to big data analytics

Sebastian Castro
.nz Registry Services

Introduction

- Apache Hadoop
  - Open-source software framework
    - Storage using HDFS
    - Data processing in batch using MapReduce
    - High level queries using Pig, Hive, Impala
    - Job scheduling using Oozie
    - Machine learning using Mahout
    - Data streaming with Storm
- Hadoop distributions (including commercial support)
  - HortonWorks, Cloudera, MapR, Datameer, Pentaho

In the begining

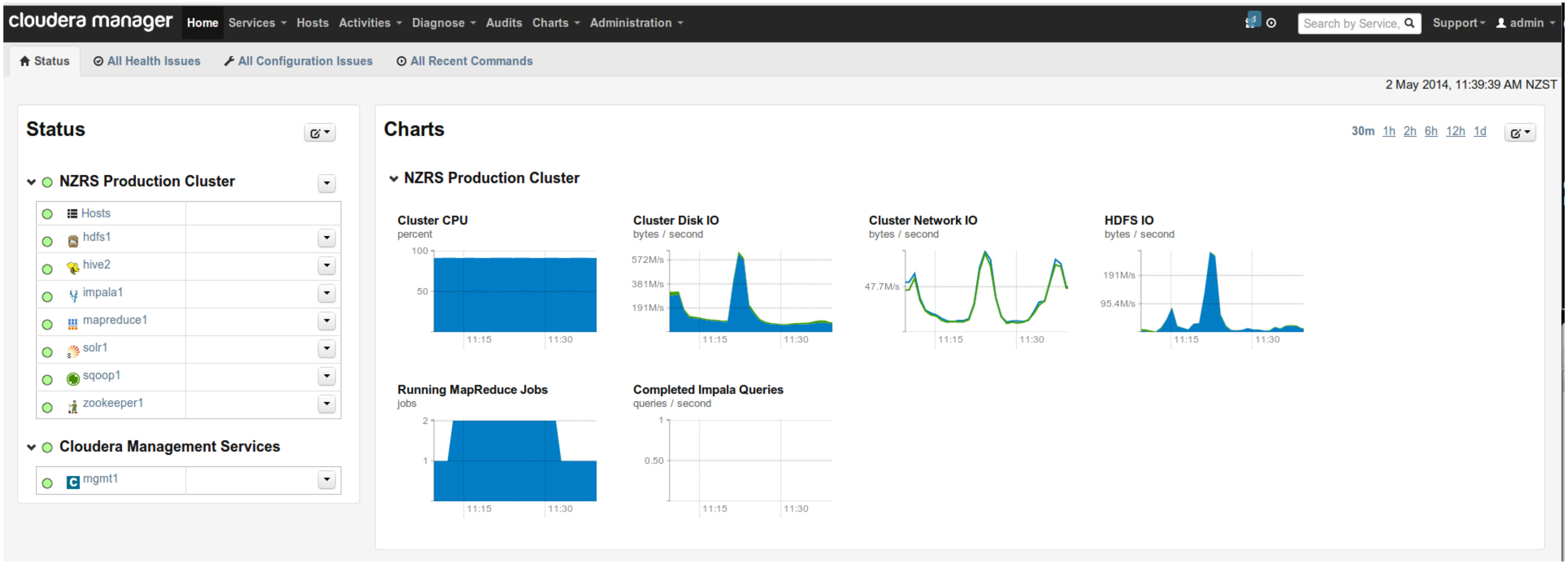
Two pallets of hardrives: 240 x 2TB

Boxes of memory: 240 x 4GB memory modules

Once built

- 1 router
- 2 GigaEthernet switches (48 ports each)
- 2 namenodes
- 20 datanodes
- 1 KVM

Cluster management



Data Catalogue

- Raw Compressed DNS PCAP
  - One file per hour per location
  - 4 nameservers (ns1 to ns4.dns.net.nz), 6 instances
  - 2 primaries
  - One external nameserver
  - Files organized as <month>/<day>/<location>
    - It's called partitioning in HDFS and allows you to run queries on a subset of files
  - Access using Hive through the Hive PCAP SerDe (RIPE)

```
hadoop@nzrs-nn01:~$ hdfs dfs -du -h /dns_data
221.8g  /dns_data/201212
246.6g  /dns_data/201301
286.5g  /dns_data/201302
374.6g  /dns_data/201303
407.2g  /dns_data/201304
420.5g  /dns_data/201305
481.3g  /dns_data/201306
421.1g  /dns_data/201307
404.1g  /dns_data/201308
391.5g  /dns_data/201309
458.2g  /dns_data/201310
445.7g  /dns_data/201311
430.9g  /dns_data/201312
418.0g  /dns_data/201401
454.2g  /dns_data/201402
554.8g  /dns_data/201403
494.9g  /dns_data/201404
14.9g   /dns_data/201405
6926.6g  /dns_data
```

Data Catalogue

PCAP table schema

Field	Type
ts	bigint
ts_usec	decimal
ip_version	int
protocol	string
src	string
src_port	int
dst	string
dst_port	int
len	int
ttl	int
udpsum	int
dns_queryid	int
dns_flags	string
dns_qr	boolean
dns_opcode	string
dns_rcode	string
dns_qname	string
dns_qtype	int
dns_answer	array<string>
dns_authority	array<string>
dns_additional	array<string>
yyyymm	string
yyyymmdd	string
host	string

Partitioning fields

Data Catalogue

- Parsed BGP table dumps
  - Once a day file from routeviews, converted to text using libbgpdump
  - Contains mapping network prefix to ASN
  - Created using a batch process every day

```
hdfs dfs -du -h -s
/bgp_data
3.9g  /bgp_data
```

Data Catalogue

- Queries per domain per hour per host
  - Aggregated data from the raw pcap
  - Partitioned by day
  - Hive batch process run every day

```
hdfs dfs -du -h -s /aggregated_data/queries_domain_hour

202.2g /aggregated_data/queries_domain_hour
```

Data Catalogue

- Zonescan
  - Imported from a MySQL database using Sqoop
  - 8 runs so far
  - Queried using Hive/Impala
  - Being worked as part of an offering to registrars for metrics
  - Aggregates exported back to PostgreSQL

```
hdfs dfs -du -h -s
/user/hive2/warehouse/z
onescan.db

65.8g
```

Data Catalogue

- Other datasets
  - Raw BIND logs from NZ-based resolver
  - Raw web pages collected using Apache Nutch
  - DNSSEC interarrival query-time (more on this later)
  - Number of queries per hour per country per SLD (aggregated)
  - Queries for bitflipped domains

Data Analysis

- The ability to scale allowed us to
  - Compare the full set of strings in the registry (500,000 unique strings) for simmilarity using some simmilarity functions
    - Jaro-Wrinkler, Monge-Elkan, Needleman-Wunch
    - Need to add Damerau-Levenshtein distance
    - Difficulty: Generate the cartesian product of 500,000 x 500,000 strings

```
hdfs dfs -du -s -h /tmp_data/jw_distance
3701.5g  /tmp_data/jw_distance
```

DNSSEC-validating resolver detection

- Hypothesis
  - A validating resolver has to refresh the DNSKEY for a signed domain every TTL or more seconds
    - Assuming it will query signed domains frequently
  - For a given IP address, calculate the time elapsed between consecutive <DNSKEY, *domain*> queries.
  - What kind of distribution that time has?

DNSSEC-validating resolver detection

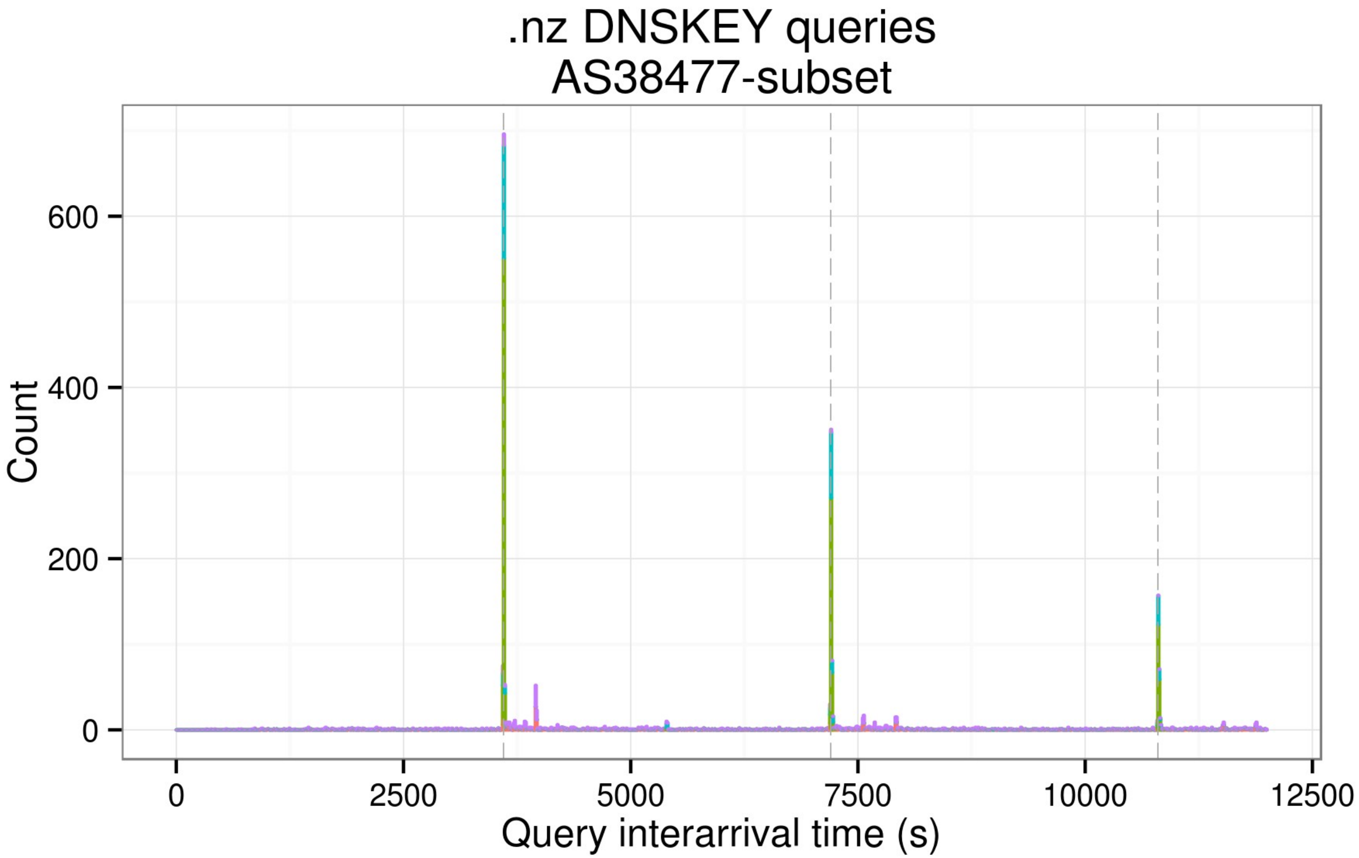
- Queries for .nz DNSKEY during Dec-2013
- Using AS numbers detected by Geoff ("Measuring DNSSEC use", NZNOG14 version)
  - DNSSEC validation per network – Top 1
    - AS22047, VTR Banda Ancha (CL)
  - DNSSEC validation per network – Top NZ
    - AS58600 – Flip
    - AS17705 – Inspire
    - AS38477 – Unleash
    - AS55853 – Megatel
- Control case: AS38477 address sub-set of known validators

DNSSEC-validating resolver detection

- Data preparation
  - Find all addresses sending <.nz, DNSKEY> queries for the selected ASes
  - Select the stream of queries per address
  - Calculate the interarrival time of consecutive queries (ordered by time)

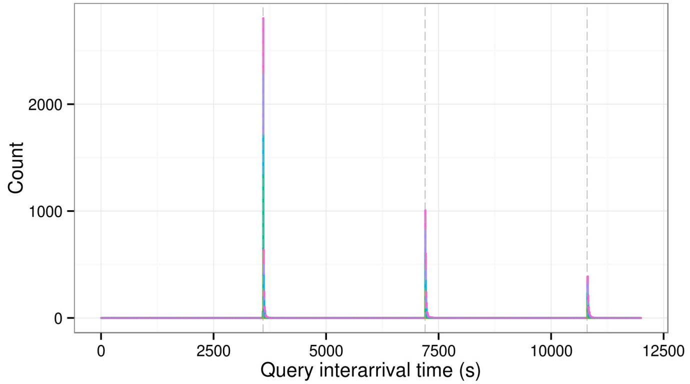
DNSSEC-validating resolver detection

- Base case: 4 addresses

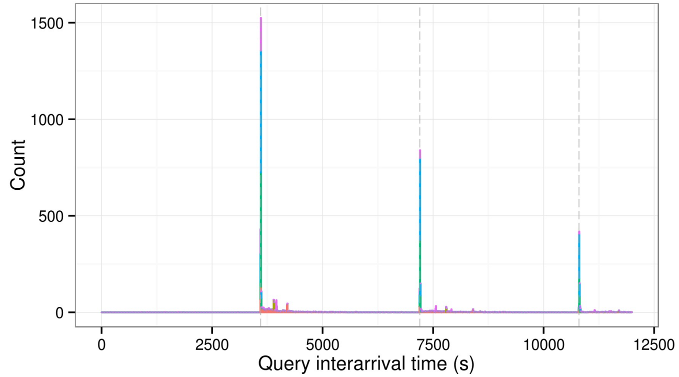


DNSSEC-validating resolver detection

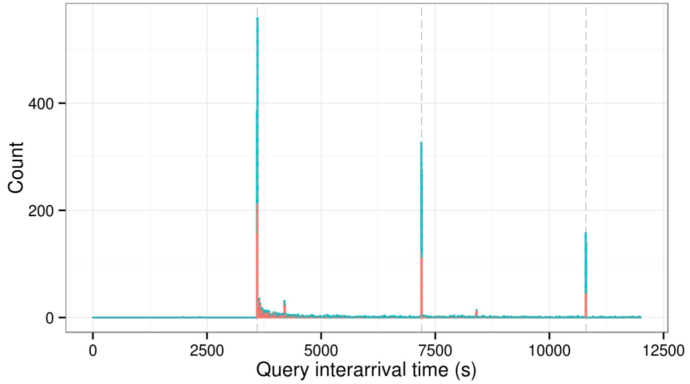
.nz DNSKEY queries
AS22047



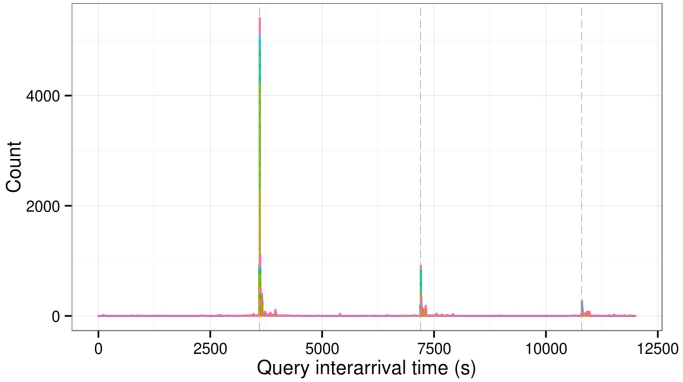
.nz DNSKEY queries
AS17705



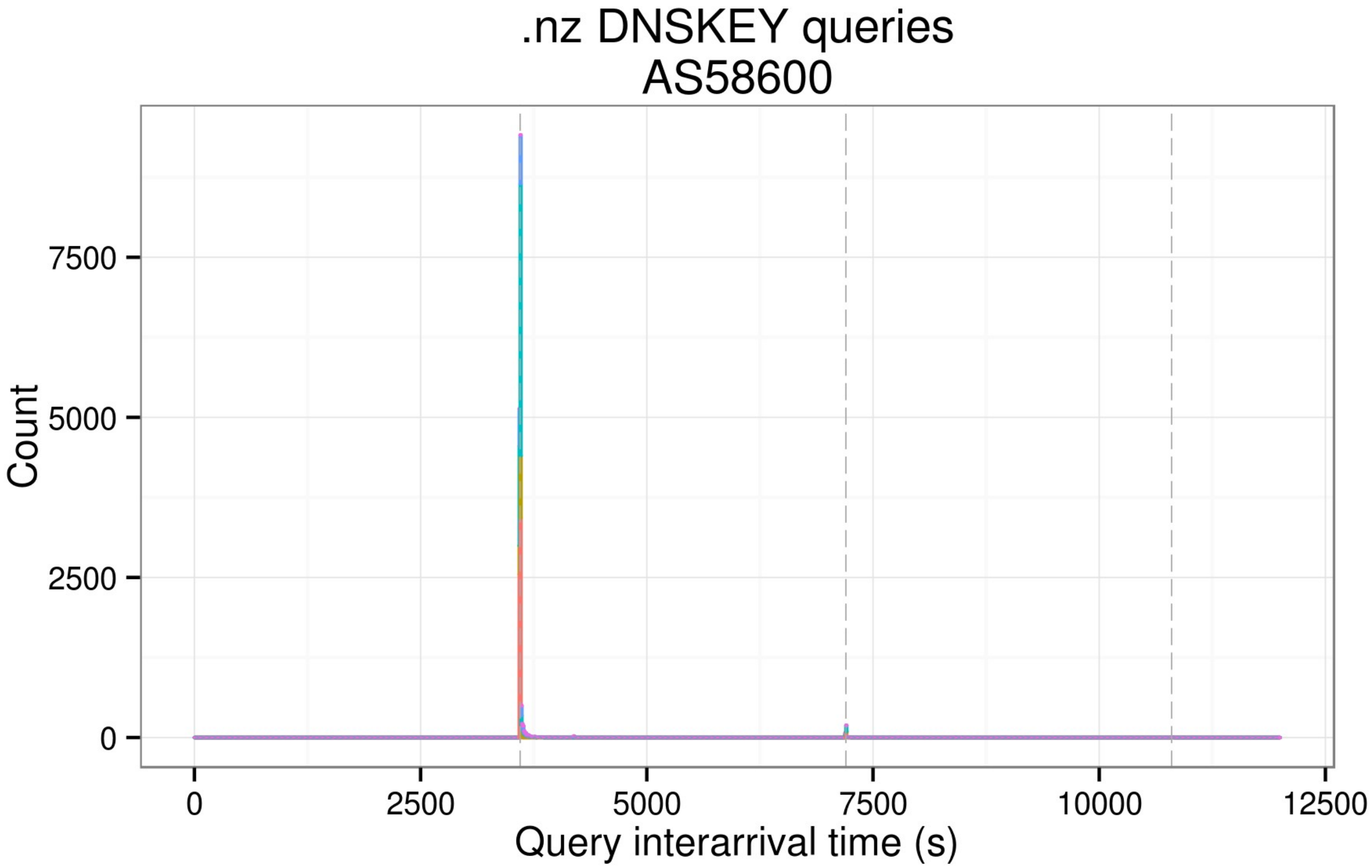
.nz DNSKEY queries
AS55853



.nz DNSKEY queries
AS38477



DNSSEC-validating resolver detection



Code!

- https://github.com/NZRegistryServices/nzrs-hive-udf
- Collection of Hive UDF (User Defined Functions) to do:
  - String simmilarity (based on Simmetrics library)
  - getLabel: Pick labels of a domain name
  - addressFamily: Tell if an string represents an IP address of certain family
  - GenericUDFDelta: Returns the difference between two consecutives values grouped by a key
  - prefixMatch: Is this address part of the given prefix? (v4 or v6)

Lessons learned

- Plan for an extra box for monitoring, management and other side functions
  - Currently running on namenodes, which run more than a few roles for the cluster
- Vendor-specific management interfaces help a lot!
  - Don't underestimate the value of 200+ configuration parameters
- Hadoop environment is changing rapidly
  - New versions, new tools, improvements every month

Lessons learned (2)

- Prefer text or Hadoop-specific binary formats (SequenceFile, Parquet) over PCAP
  - Greater speed and scalability
    - Hive PCAP SerDe will give one file per mapper
      - Native formats support splitting to analyze the same file by more than one mapper at a time
    - Can be read by other Hadoop tools like Pig or Impala

Future Plans

- The cluster will be the backend of a data product
- New datasets will be added
  - Domain classification per economic activity
  - Domain popularity (based on DNS traffic)
  - Other domain classification (web data?)
- More exploration of existing datasets

Questions?