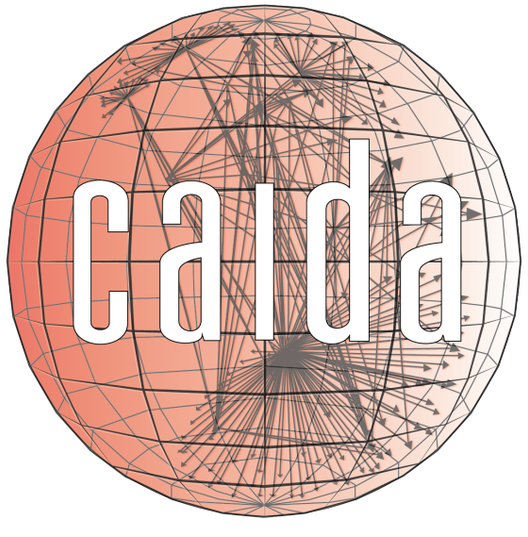
FANTAIL:

Facilitating Advances in Network Topology Analysis

Young Hyun

CAIDA

AIMS-KISMET Workshop
Feb 28, 2020



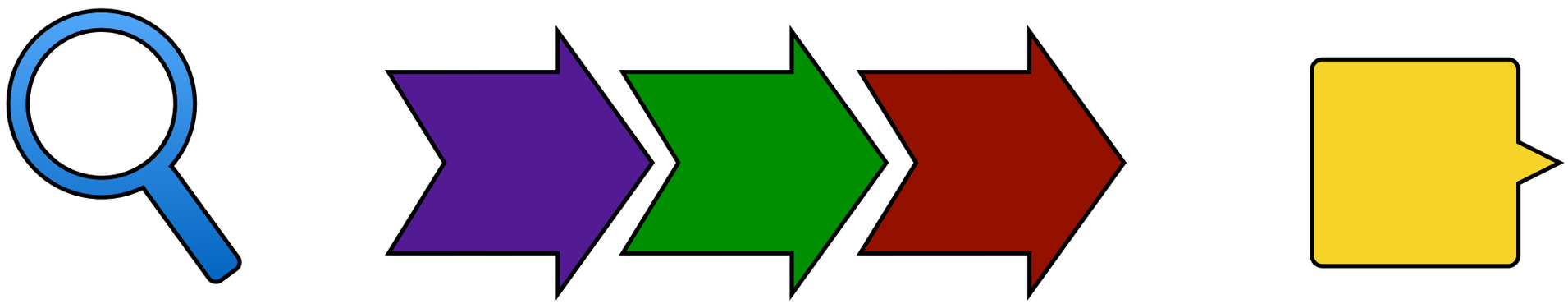
Background

- network community has **vast amounts of traceroute data**
- CAIDA has 90+ billion IPv4 traceroutes in 39+ TB of files
  - growing by 16 billion traces and 7 TB annually
- RIPE Atlas collects ~700 million IPv4/IPv6 traceroutes/month
- Measurement Lab (M-Lab) collects millions of traceroutes/day
- and more ...

Goals

- allow researchers to **more easily use** community topology data
- ... to **search** traceroute data
  - using high-level queries
- ... to perform **data processing and analysis** tasks on matching traces
  - without owning/operating a cluster
  - without learning big data programming

Workflow



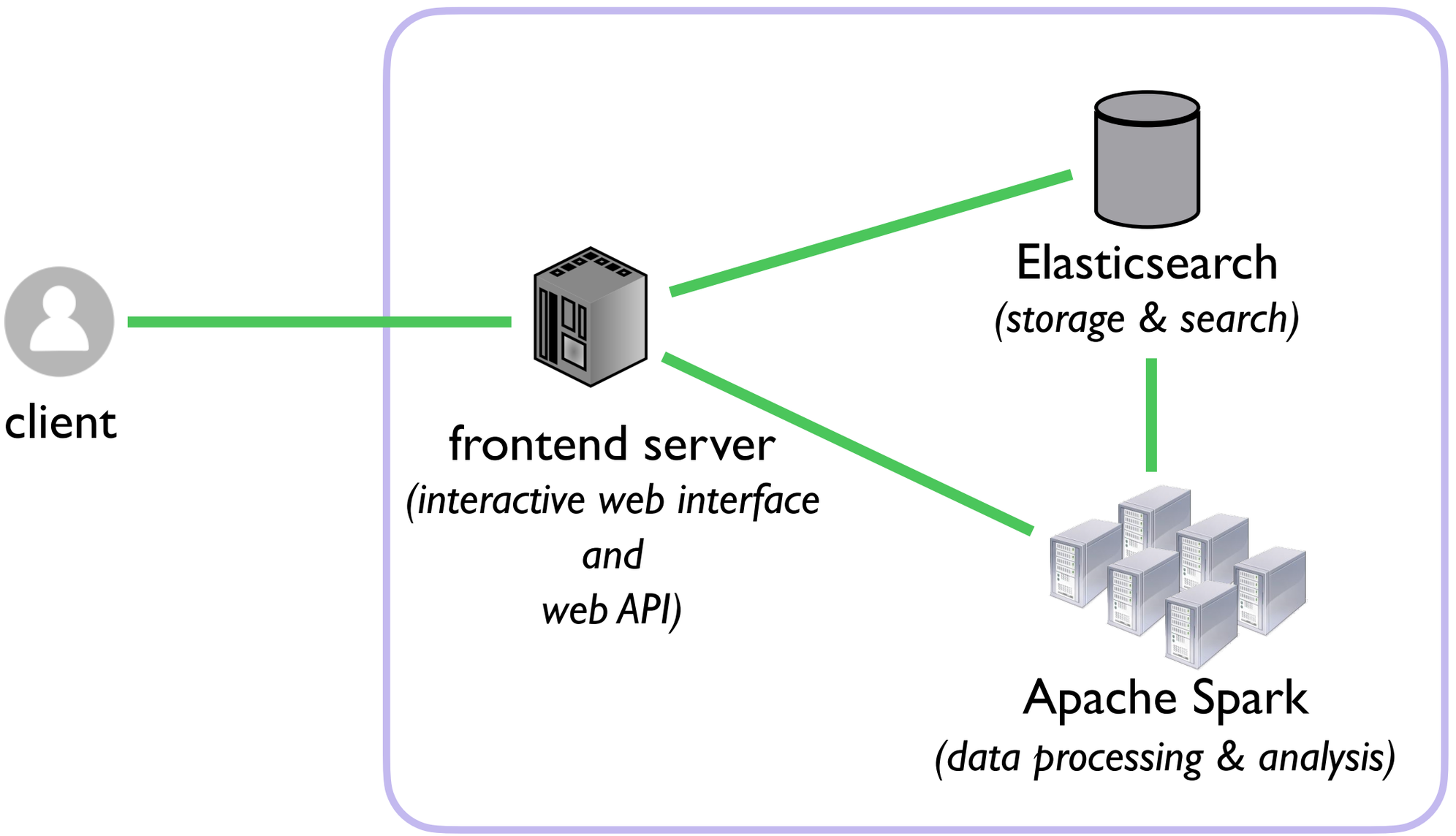
query

data processing/analysis pipeline

(optional)

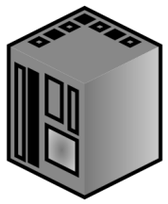
results

Architecture

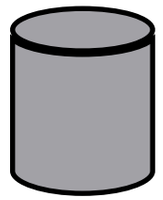


Querying

client



FANTAIL



Elasticsearch

Spark

- User specifies high-level search criteria
- FANTAIL performs low-level Elasticsearch queries against relevant indexes and traceroute fields

Queries (1/2)

Query	*Selection Criteria*
vp V	vantage point is V
vp_as N	vantage point is located in autonomous system (AS) N
vp_country C	vantage point is located in country C
vp_type T	vantage point is hosted by an organization of type T
status N	traceroute has success/failure code N
timestamp *op* N	traceroute has timestamp < = > N
dest_rtt *op* N	RTT of traceroute destination is < = > N ms
pathlen *op* N	length of traceroute path is < = > N
has_mpls *T/F*	whether there is *(T)* or is not *(F)* MPLS in the traceroute path

op is < or = or >

Queries (2/2)

Query	*Selection Criteria*
`dest` G	traceroute destination is any address $x \in G$
`hop` G	traceroutes with any address $x \in G$ appearing at any hop
`neigh` $G_1 \ldots G_n$	traceroutes with n distinct neighboring hop addresses $x_i \in G_i$

T = target address/prefix/AS/country

G = target group T_1 | … | T_m

Queries (2/2)

Query	*Selection Criteria*
`dest` G	traceroute destination is any address $x \in G$
`hop` G	traceroutes with any address $x \in G$ appearing at any hop
`neigh` $G_1 \ldots G_n$	traceroutes with n distinct neighboring hop addresses $x_i \in G_i$

T = target address/prefix/AS/country

G = target group T_1 | **`...`** | T_m

query: **`neigh 10.0.0.0/8|192.168.0.0/16 AS1|AS2|AS3`**

matches any trace with hop addresses x and y such that

($x \in$ **`10.0.0.0/8`** or $x \in$ **`192.168.0.0/16`**)
and ($y \in$ **`AS1`** or $y \in$ **`AS2`** or $y \in$ **`AS3`**)

Queries

- can form complex queries by combining elements with logical *AND*:

```
  vp_country .fr
& timestamp > 2018-01-01
& timestamp < 2019-01-01
& dest 1.0.0.0/8
```

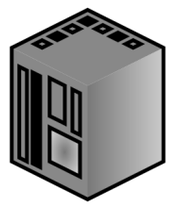
- can use logical *OR* **within** certain elements:

```
  vp_country .fr|.de|.uk
& vp_type business|residential
& dest 1.0.0.0/8|2.0.0.0/8
```

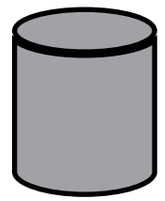
- must have overall form X & (Y_1|...|Y_n) & ... & Z
  - only two levels of logical operators: & at top, | within
  - no *NOT*
  - need restrictions for efficient execution of queries

Data Processing

client

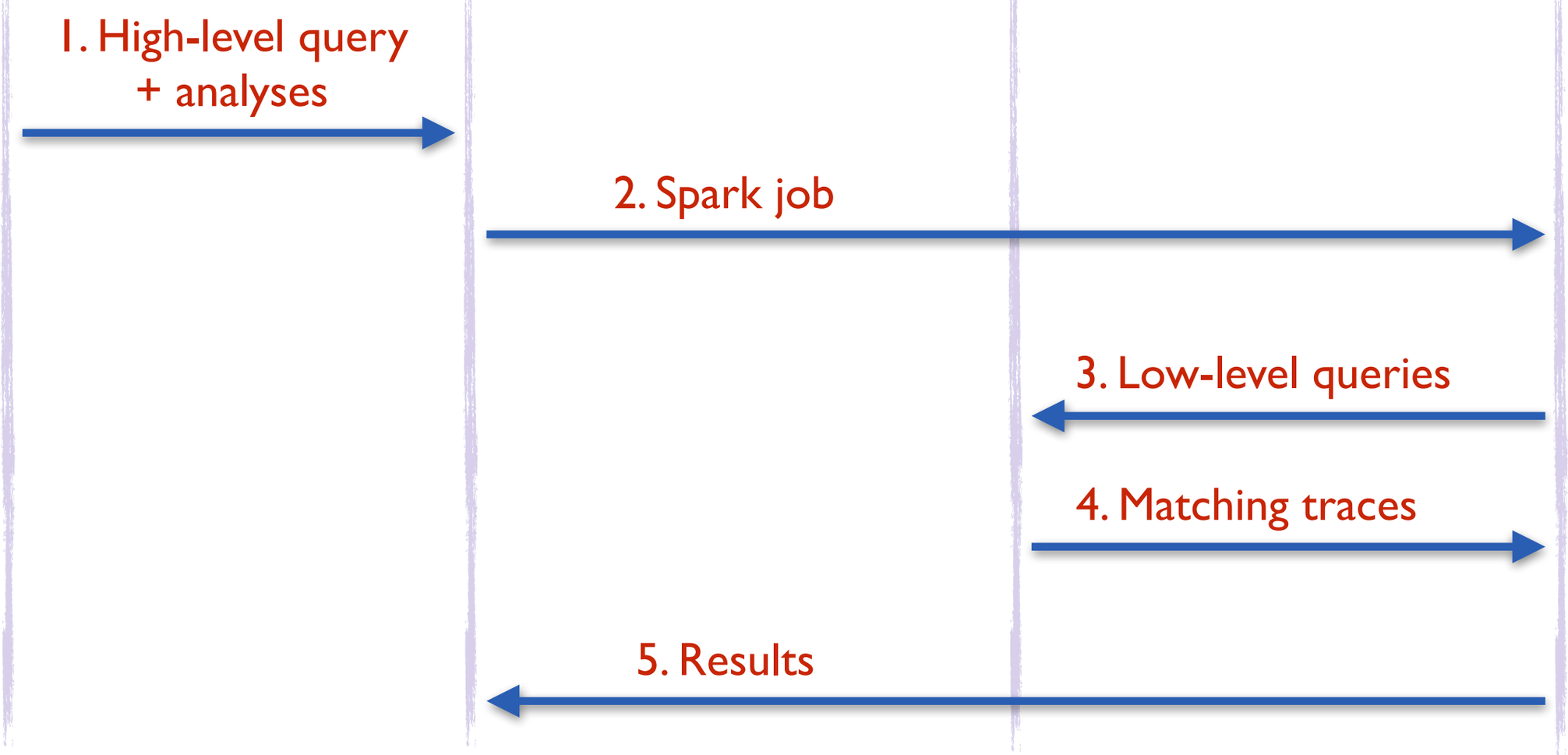


FANTAIL



Elasticsearch

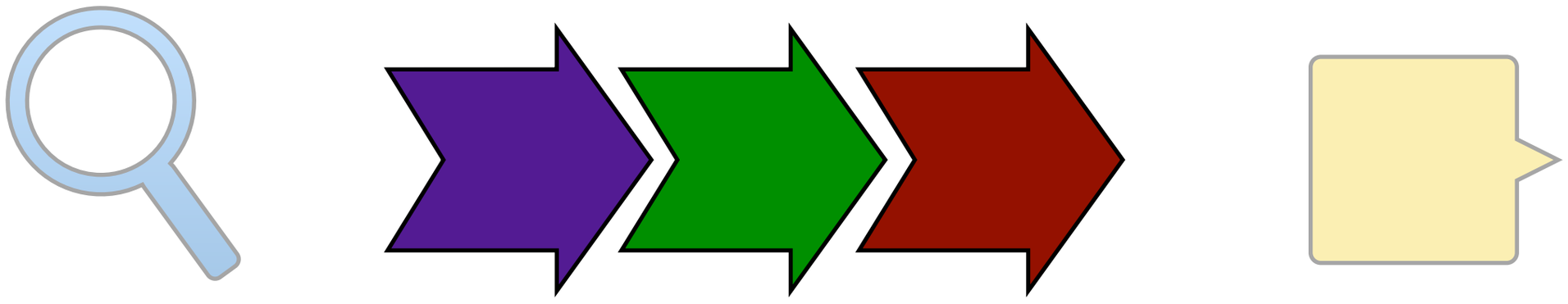
Spark



- User specifies high-level search criteria + desired data processing/analyses to apply to matching traces

Data Processing

user specifies a **chain of analysis modules** to define the data processing/analysis pipeline



query

data processing/analysis pipeline

results

Data Processing

- **analysis module:** a reusable data processing or analysis step that transforms traceroute paths
  - for performing **data reduction** (to minimize the amount of data users have to download and further process)
  - for enhancing raw traceroute data with **annotations**
  - for **offloading** implementation and execution of commonly-needed analysis/data processing tasks from users

Data Processing

- **analysis modules:**
  - extract unique hop addresses, IP links, or IP paths
  - annotate IP addresses with aliases, hostnames, IXPs, ASes, interconnection points (bdrmapIT), presence of MPLS tunnels
  - calculate per-destination RTT distributions

Data Processing

- **analysis *recipes***
  - prepackaged queries and data processing/analysis pipelines provided as a unit
  - customize via parameters, such as the target AS/prefix/country
- **samples:**
  - Find all interconnection links of a given AS in a given time period.
    - Implementation: (1) find all traceroutes that have IP addresses in the target ASes, (2) identify interconnection links in the selected traceroutes with bdrmapIT data, and (3) produce the unique set of interconnection links from these traceroutes.
  - Extract the IP/router-level graph of a target AS (that is, all routers and links that map to the AS) in a given time period.

Thanks! Questions?

- much development work ahead (at 6 months of 3 year funding)
- email `fantail-info@caida.org` if interested in *very* early access (pre-alpha)

FANTAIL is funded by NSF CNS-1925729