RDAP CDN Distribution Experience

tomh@apnic.net
ROW #8
9 May 2019

Implementation history

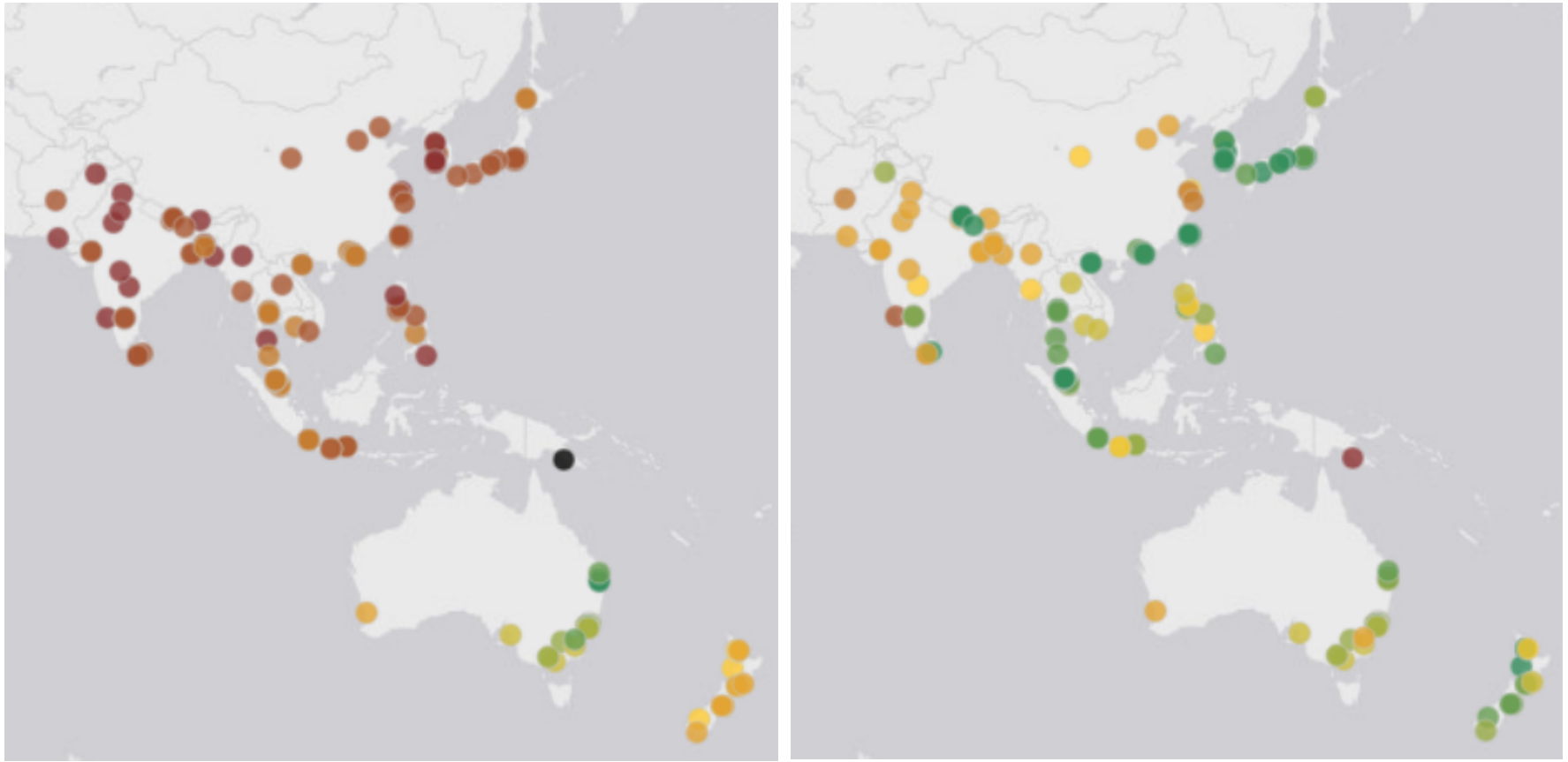
- 2014: Original implementation
  - Generate RDAP responses on the fly based on Whois data
    - Simple, but slow
- 2016: History server (for draft-ellacott-historical-rdap-00)
  - Pregenerate responses at startup, and maintain them over time
    - More memory, but faster and more resilient
- 2018: Adapt History server for standard RDAP
  - Same advantages/disadvantages

Latency issues

- Late 2018: Try to improve latency
  - RDAP still running as a single node in AU
  - Obvious option: replicate nodes, like we do with Whois
  - But RDAP is HTTP: maybe there are better HTTP-specific options
- Requirements
  - Improve latency, particularly for clients in the AP region
  - New objects should propagate ‘quickly’ (within 5 minutes)
  - No particular requirements around technology/platform, etc.

Cloudflare Workers and KV

- A solid candidate based on the requirements:
  - 180 PoPs, and very good AP coverage
  - Workers (effectively JS HTTP handlers) run in each PoP
  - Workers KV (distributed key-value store) is presented to workers as an object with get/set methods
  - KV updates are synchronised within 10s
  - Worker cold start time is better than other serverless options, and fast enough that it could be used to handle queries directly



< 10 ms: 69 | < 20 ms: 34 | < 30 ms: 18 | < 40 ms: 17 | < 50 ms: 15 | < 100 ms: 28
< 200 ms: 6 | < 300 ms: 1 | > 300 ms: 1 | No Data: 4 | Unreachable: 0

The initial plan

- Continue to run current server, but adapt it to push updated objects to the Cloudflare Workers KV
  - KV currently in beta, but in GA will support 1B objects per namespace, up to 2MB each
- Workers handle simple object fetches (IP/entity/domain)
  - The overwhelming majority of our traffic
- Anything more complicated (redirect to other registry, search, history, etc.), let current server handle it

Results

- For our specific situation, the plan didn't work:
  - Not a problem with Workers or KV as such: they were very easy to work with, setup was simple, everything was as advertised
  - But KV data is held centrally: the first fetch from a given PoP takes time (averages 750ms in our unscientific testing)
  - Plus, KV data is evicted from PoPs quickly (<30s), unless it is retrieved regularly by workers from that PoP
  - This would be great for a small set of regularly-fetched objects (KV data that has been recently fetched is returned in 30-100ms), but that's not what we have

Next steps

- Evaluate other options
  - There's not exactly a shortage
  - Lots of recent activity, too
    - Fastly Terrarium, Google CloudRun
- Possibly use a CDN as a simpler 'smart' cache in front of distributed nodes

Questions?