IETF Hackathon: Measurement & Analysis for Protocols Research Group (MAPRG)

IETF 105

20-21 July 2019

Montreal

Hackathon Plan

- **Problem:**
  - **Related draft:** kIP: a Measured Approach to IPv6 Address Anonymization
    https://arxiv.org/abs/1707.03900
    https://www.ietf.org/proceedings/99/slides/slides-99-maprg-kip-a-measured-approach-to-ipv6-address-anonymization-03.pdf
    https://www.youtube.com/watch?v=qYtaKuzXaiM#t=59m55s

  - **Specific problem to solve:** Implement a modified PATRICIA trie or base-2 radix tree for "longest prefix match" making feasible for the tens to hundreds of billions of active IPv6 addresses used on the web today.

- **To solve it:** enhance aguri_tree to make portions of the tree *immutable.*
  https://www.iijlab.net/~kjc/software/#aguri

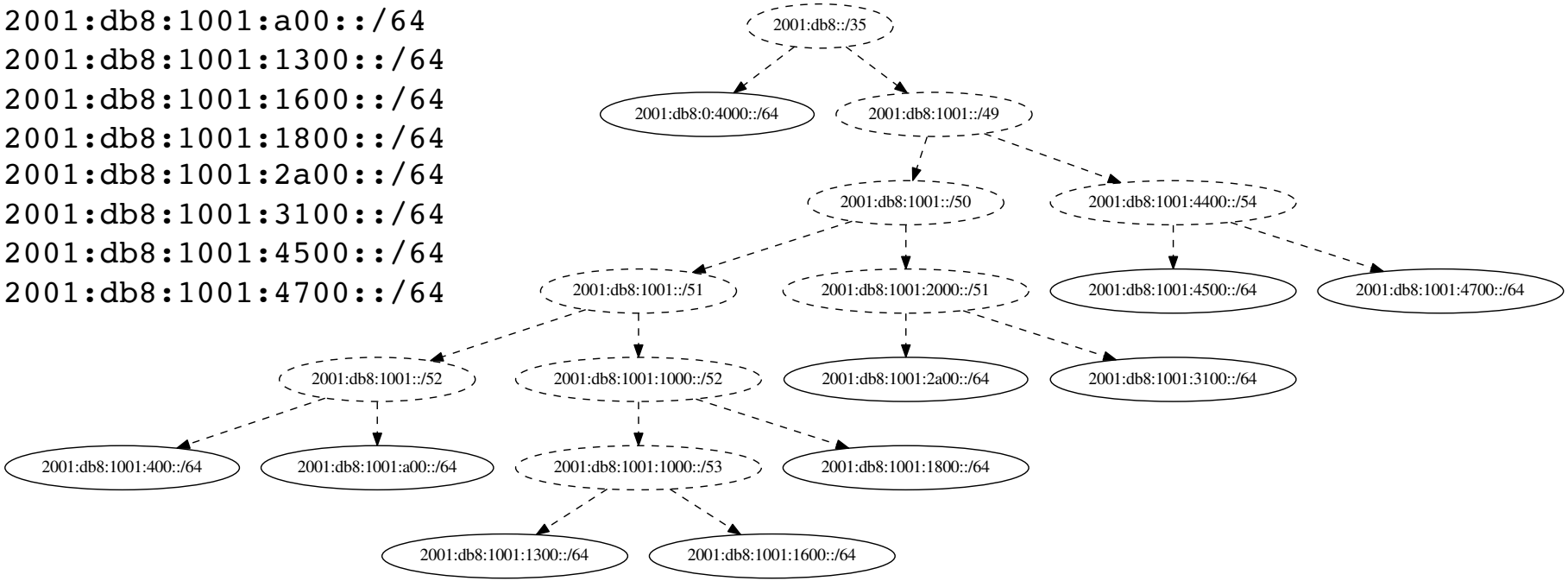
  This allows partitioning of the problem for map-reduce/cluster operation, splitting the active IP addresses into manageable subsets (files) to produce intermediate results that are subsequently combined using prior mode (non-immutable nodes).

What got done

- <What you achieved? (key results)>
  - <New ideas - what team agreed on>
  - github pending
  - What was novel?
  - <Demos - links to videos>

What got done

2001:db8:0:4000::/64
2001:db8:1001:400::/64
2001:db8:1001:a00::/64
2001:db8:1001:1300::/64
2001:db8:1001:1600::/64
2001:db8:1001:1800::/64
2001:db8:1001:2a00::/64
2001:db8:1001:3100::/64
2001:db8:1001:4500::/64
2001:db8:1001:4700::/64



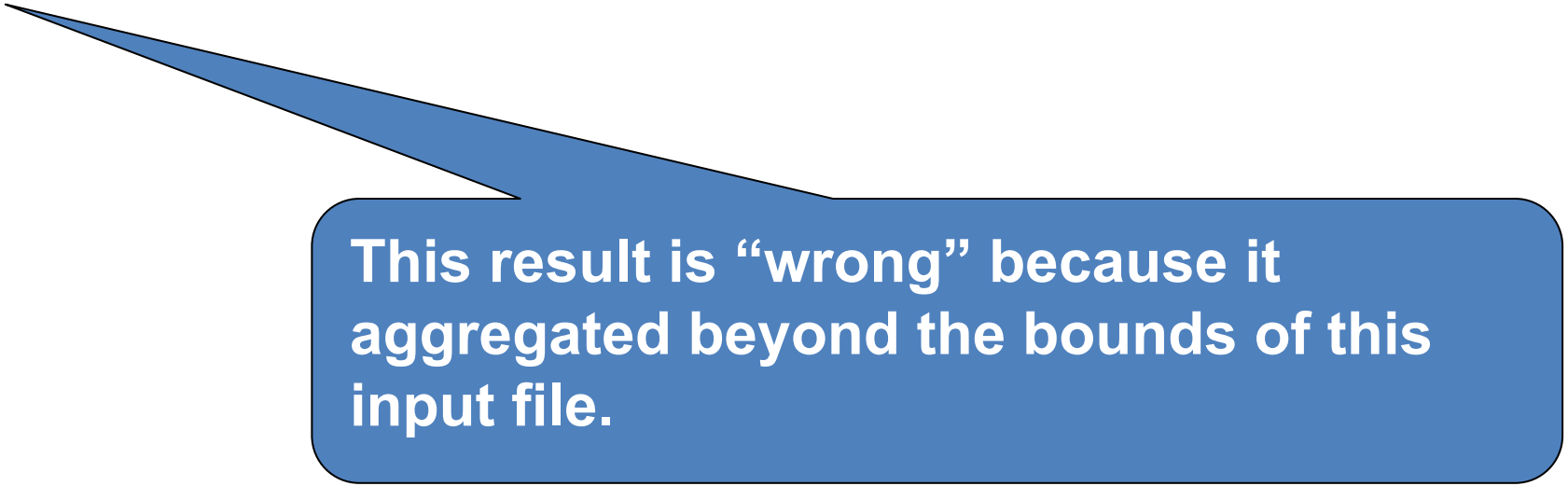
What got done

```
$ ./agurify -v -p50 -k32 input_fragment_v6.txt
::/0
```

Aggregate to prefixes having at a minimum of k=32 simultaneously active addresses

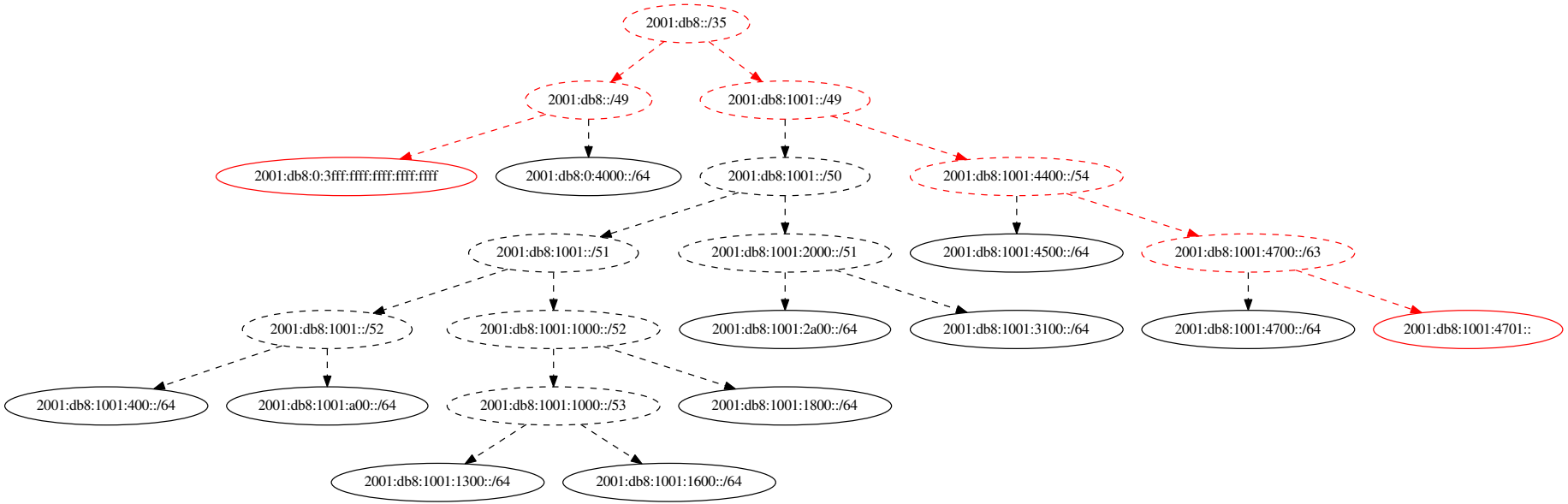
What got done

```
$ ./agurify -v -p50 -k32 input_fragment_v6.txt
::/0
```



What got done

2001:db8:0:3fff:ffff:ffff:ffff:ffff added immutable...
2001:db8:1001:4701:: added immutable...



What got done

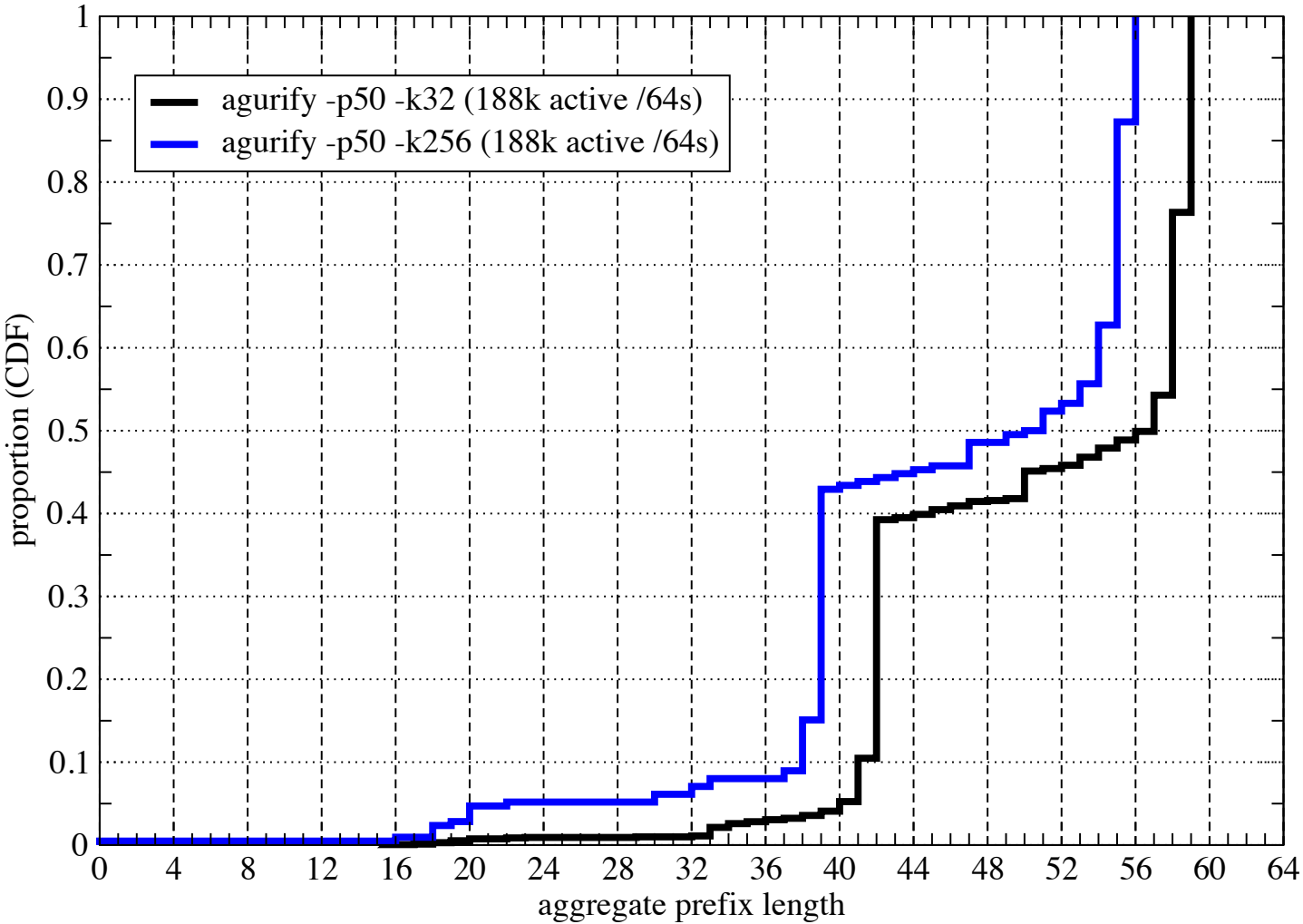
```
$ ./agurify -v -i -p50 -k32 input_fragment_v6.txt
20010db800003fff:[illegible]ffffffffffffffff added immutable...
20010db81001470[illegible]00000000000 added immutable...
```

Produce an intermediate result, i.e., don't aggregate to any prefix which could cover any address not appearing in this input file.

What got done

```
$ ./agurify -v -i -p50 -k32 input_fragment_v6.txt
20010db800003fffffffffffffffffff added immutable...
20010db810014701000000000000000000 added immutable...
        2001:db8:0:4000::/64 335
       2001:db8:1001::/50 335
        2001:db8:1001:4500::/64 335
        2001:db8:1001:4700::/64 335
```

This is a useful intermediate result that can be processed, iteratively, toward the final result.


agurify -p50 -k32 (188k active /64s)
agurify -p50 -k256 (188k active /64s)
proportion (CDF)
aggregate prefix length
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64

What we learned

- Lessons learned from this hackathon
  - Implemented IPv6 address “bignum” arithmetic rather than using GMP (GNU Multiple Precision Arithmetic Library)
  - New implementation/operation guidance?
    - Candidate best practice for aggregation-based IP address anonymization for privacy, *e.g.,* GDPR compliance
    - Aids investigation of address assignment practice, *e.g.,* to produce homogenous end user aggregates for matching with content in delivery networks
  - New work to take to WG?

    A reference implementation of kIP

Wrap Up

Team members:
Dave Plonka

{based on code by
Kenjiro Cho,
Ryo Kaizaki}

MAPRG meets Friday morning:
https://datatracker.ietf.org/rg/maprg/about/
https://trac.ietf.org/trac/irtf/wiki/map

maprg-chairs@ietf.org