On network aware clustering and its applications

Balachander Krishnamurthy

http://www.research.att.com/~bala/papers

Background

I have examined Web related issues for a few years focusing on

- Making the Web experience palatable, improving the HTTP protocol, contributing to standards
- Gathering a variety of data across different protocol layers, clean/process it quickly for analysis
- Resulting in a book:

Web Protocols and Practice:
HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement
(672pp, Addison-Wesley, May 2001).

This talk focuses on using network information for Web client clustering

How did I get here?

- Began with Web caching, er, sorry, content distribution
- Current CDN focus is on content hosting companies
- But what if we want to examine things from a client's POV?
- Need to locate interesting sets of clients from outside
- No current registry!

Network aware clustering

Can we

- detect *clusters* of clients based on server logs?
- do so accurately and in an automatated fashion?
- locate special clients (spiders, proxies) within the clusters?
- *use* this information in real time?

Example applications: relocating content, placing proxies, targeted ads...

What is a cluster?

- Non-overlapping collection of IP addresses
- Topologically close
- Under common administrative control

Identifying clusters requires knowledge that is not available to anyone outside the administrative entities.

Quick attempts at clustering

- Use traditional Class A, Class B, and Class C networks
- Assume prefix length is 24 bits
- Simple indeed, but also wrong
- Accuracy: 50%

Really?

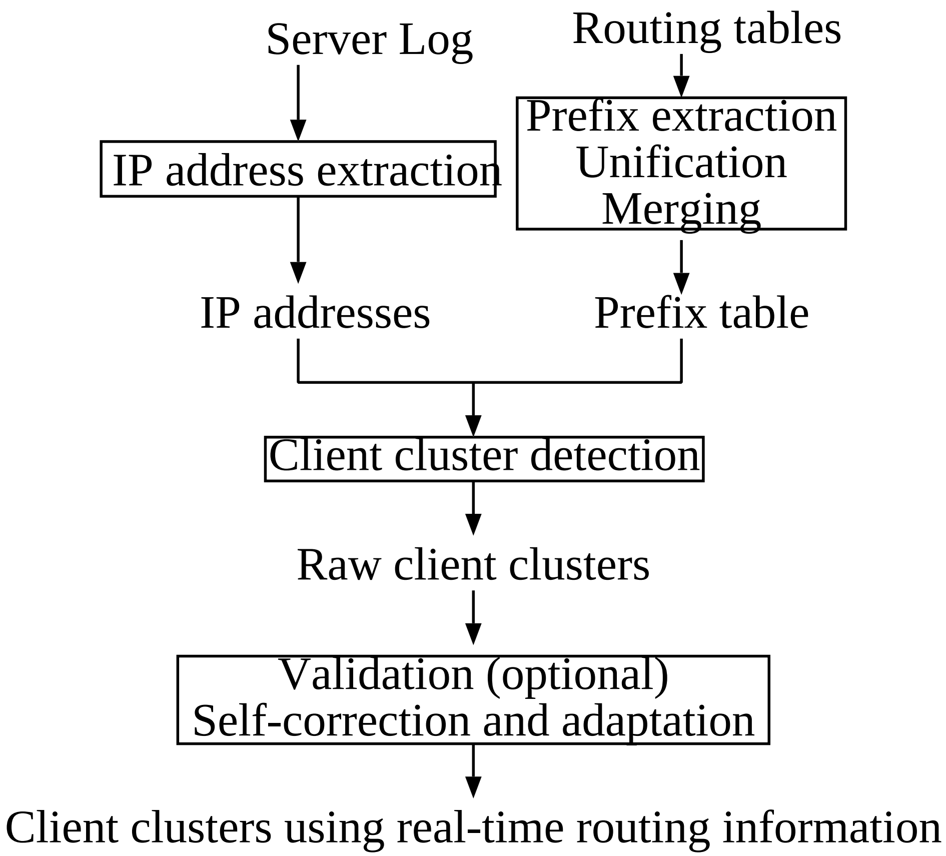
- 151.198.194.17, 151.198.194.34, 151.198.194.50
- Same first three bytes of IP addresses.
- Group into one Class C network, prefix 151.198.194, netmask length 24
- Reality: three different networks with prefixes 151.198.194.16, 151.198.194.32, and 151.198.194.48, netmask length 28 and names:

```
client-151-198-194-17.bellatlantic.net
mailsrv1.wakefern.com
firewall.commonhealthusa.com
```

Clustering clients: in a nutshell

- Currently we know only IP address in server logs
- Want to group clients down to administrative entity controlling sets of addresses
- Use routing table information - obtained from many places via a script, unified into one big table of prefix/netmask entries
- Perform longest prefix match on each client IP address using table
- Classify all IP addresses with same longest matched prefix into one client cluster (identified by the shared prefix).

Network aware clustering: automated process



Where to get routing table snapshots?

Name	Date	Size	Comments
AADS	5/27/2001	32K	BGP RTS updated every 2 hours
ARIN	10/1999	300K	IP network dump
AT&T	4/28/1999	65K	BGP forwarding table snapshots
CANET	12/1/1999	1.7K	Real-time BGP RTS
CERFNET	9/29/1999	50K	Real-time BGP RTS
MAE-EAST	5/27/2001	16K	BGP RTS taken every 2 hours
MAE-WEST	5/27/2001	71K	BGP RTS taken every 2 hours
NLANR	11/1997	200K	IP network dump
OREGON	12/7/1999	70K	Real-time BGP RTS
PACBELL	5/27/2001	45K	BGP RTS updated every 2 hours
PAIX	5/27/2001	18K	BGP RTS updated every 2 hours
SINGAREN	5/27/2001	68K	Real-time BGP RTS
TELSTRA	5/27/2001	110K	
VBNS	12/7/1999	1.8K	BGP RTS updated every 30 minutes
WORLDNET-BGP	12/15/1999	74K	BGP RTS

But am I in the right cluster?

- 441K unique prefix entries by (5/27/01)
- Cluster may be mis-identified by being too large or too small
- Validation: *nslookup* followed by suffix match on string
- Alas, only 50% of hosts are *nslookup*-able
- Optimized *traceroute* - can't get name? check path
- Very conservative notion of correctness: 1 IP address wrong, entire cluster marked incorrect

Experimental validation

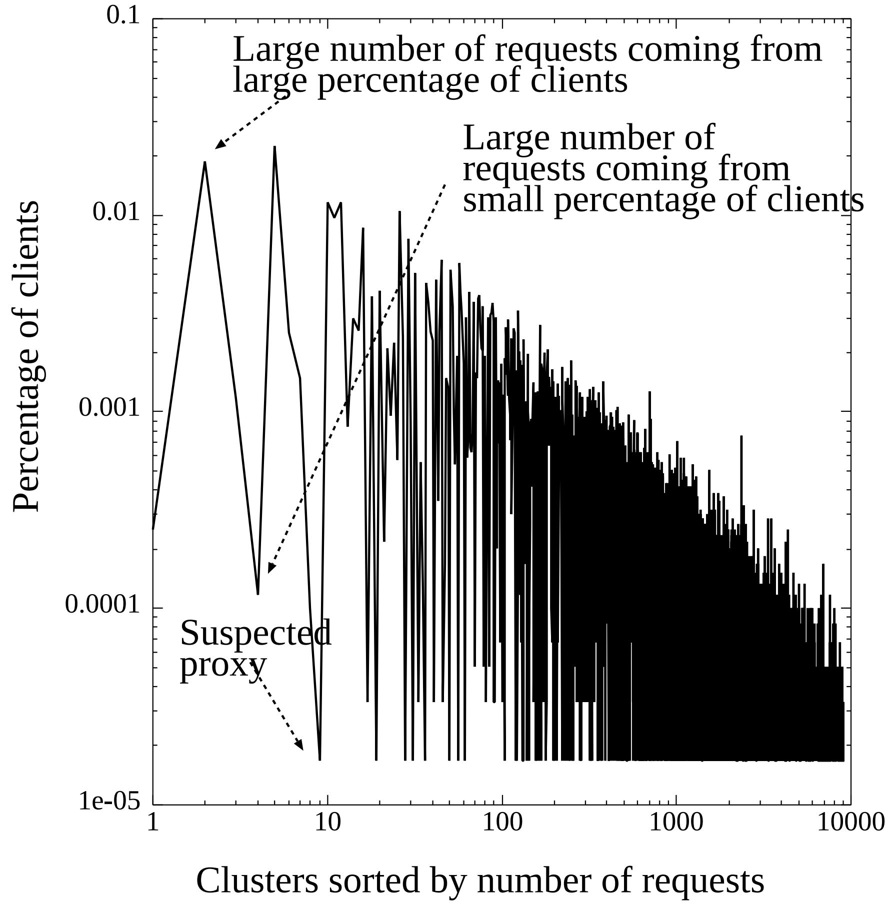
- Wide variety of logs (Apache, EW3 (11), Nagano, Sun, Chile-election)

Number of requests	148K to 104 Million
Number of clients	40K to 2.5 Million
Number of unique URLS	340 to 116K
Duration	24 hours to 94 days
Location	Across North and South America
Organization	Governmental, non-profit, commercial
Nature	transient event logs and popular site logs

- Over 99% of clients clustered in all logs
- Over 90% accuracy (with conservative correctness notion)

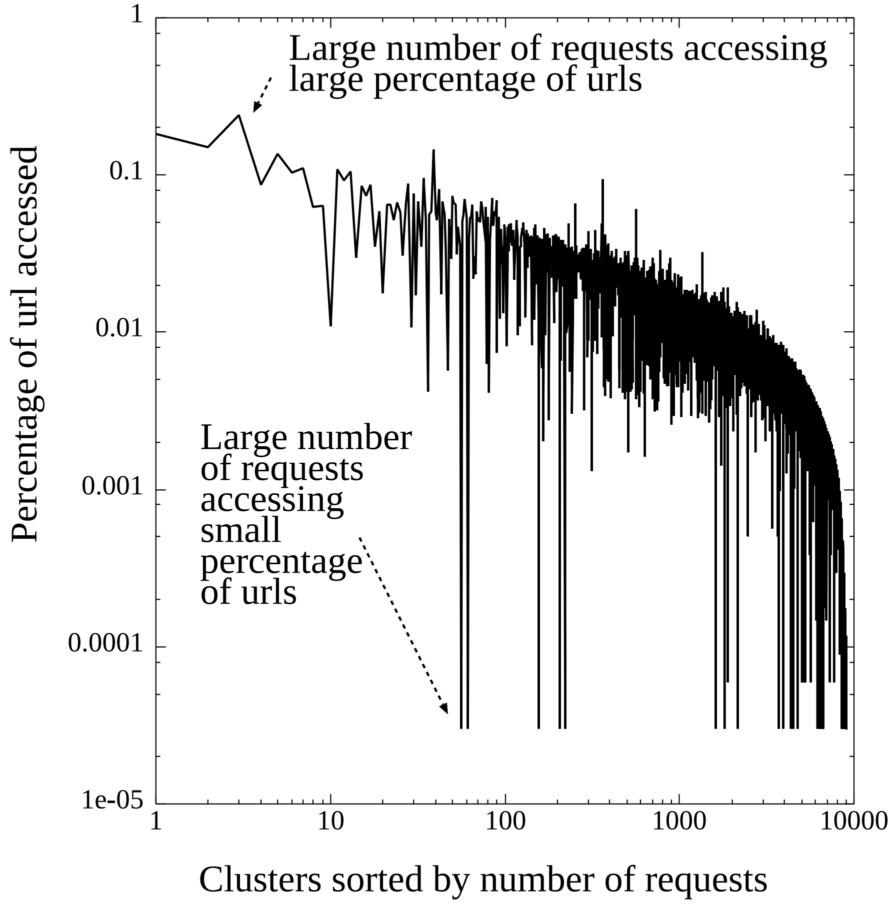
Details in our SIGCOMM 2000 paper.

Results – Nagano log client cluster distribution



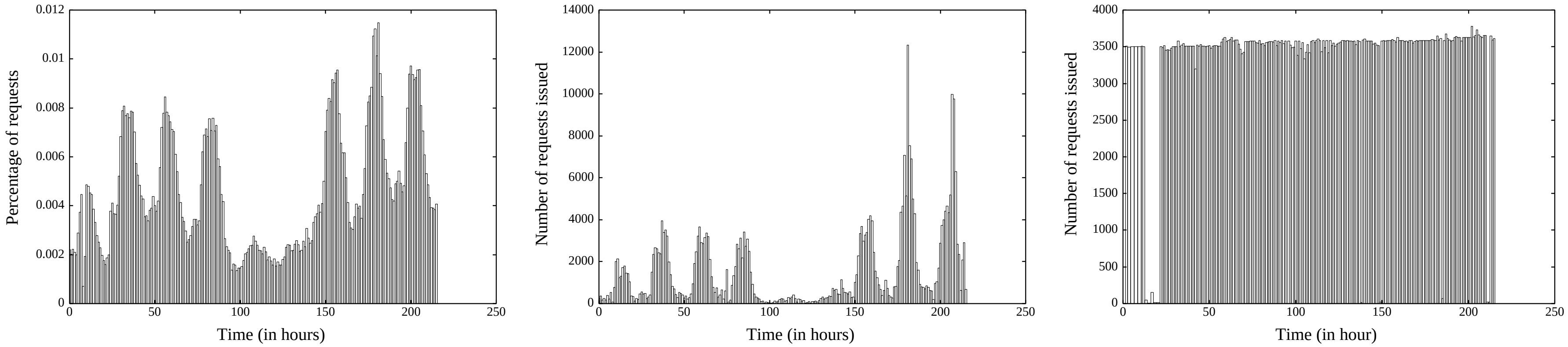
# of clients in client clusters, reverse order of # of requests (x/y in log scale)

Results 2: Nagano log client cluster distribution



# of URLs accessed from within clusters, rev. order of # of reqs (x/y log scale)

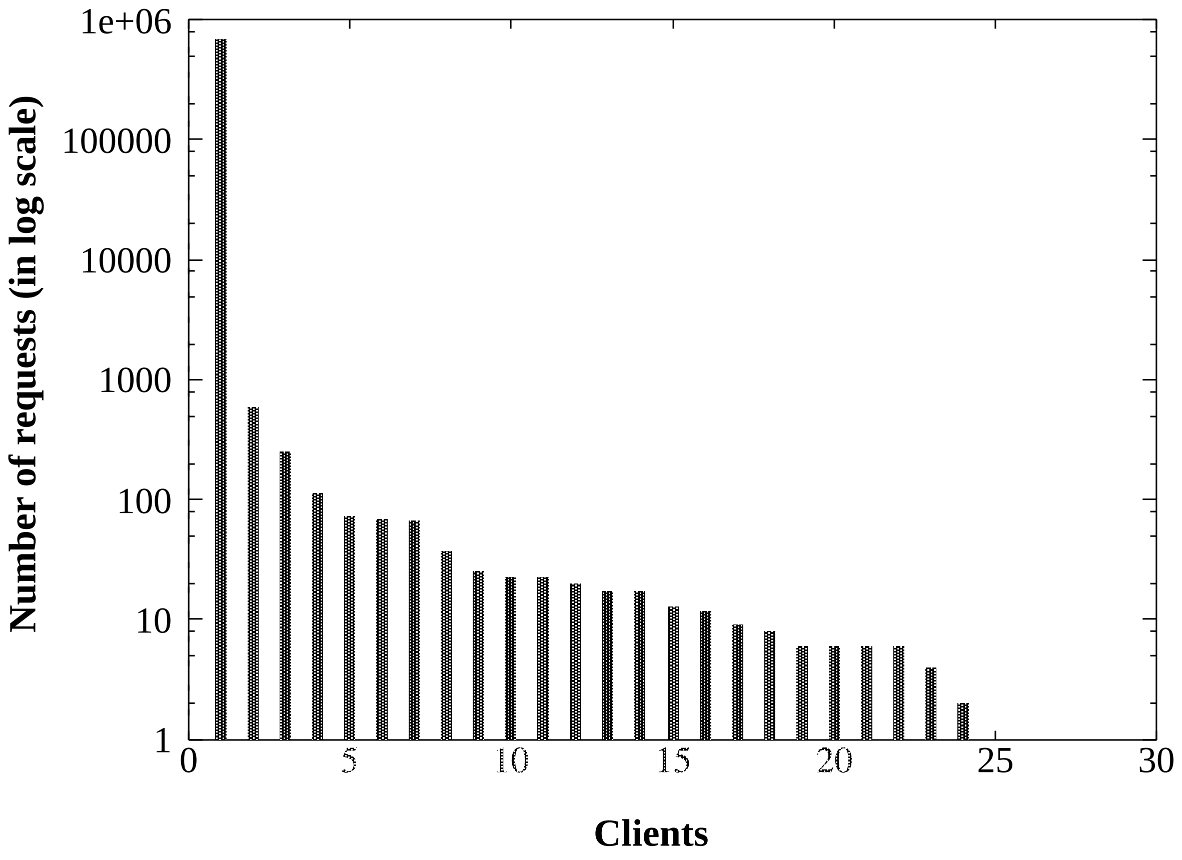
Spiders and proxies



(a) (c) (b)

Histogram of the requests in the Sun server log
(a) entire server log, (b) cluster with a proxy, (c) cluster with a spider.

Spiders and proxies contd.



Requests distribution of a cluster containing a spider (Sun log)
Spider client issued 692,453 requests (99.79% of all requests in cluster)

Recent portal site data:AltaVista logs

- 60M accesses (Spring 2000)
- 2.5M unique IP addresses from 182 countries classified into 100K clusters
- Reduced to 242 clusters (38 countries) by focusing on busy clusters (responsible for 70% of all accesses) with 91% accuracy
- 104 Million accesses (Spring 2001)
- 7.6M unique IP addresses classified into 114,928 clusters
- Reduced to 2874 clusters

Note: No private information about any AV user was made available

New applications of clustering

Clustering is becoming recognized as a clear alternative to other forms of grouping IP addresses:

- Potential of geo-location of IP addresses (SIGCOMM 2001)
- Mirror/replica placement (Infocom 2001)
- Estimating Internet distances
- Power laws of clusters

Our cluster applications

- Internet topology modeling
- Identifying DNS servers in busy clusters
- Reducing traffic in P2P applications
- Helping differentiate between DDOS attacks and flash events

Papers on these to appear: ACM SIGCOMM Internet Measurement Workshop (IMW 2001) Nov, 2001

Internet topology modeling

- AS-level topology too coarse-grained. AS-path does not reflect policy routing paths of data packets
- Router-level topology reflects policy paths but prohibitively expensive to construct
- Synthetic AS graph: generated based on observed charateristics such as power law properties. But not yet shown to be realistic.
- Instead, model topology at an intermediate level of granularity between router-level and inter-domain (AS) level topology

Cluster graphs

- Group routers/hosts into clusters and treach each cluster as a node and inter-cluster connection as edge between the nodes
- Can build cluster graphs via BGP tables and traceroute results, or based on power law properties.
- Instead of running traceroute to all IP addresses it is enough to run it to a few IP addresses in a cluster
- Additionally, cluster graphs are more stable.

Example: Cluster path towards 130.75.2.24

traceroute path	cluster path	*traceroute* path	cluster path
0 *** 1 *** 2 *** 3 ***	cluster 1	9 157.130.254.242	157.130.0.0/16
4 ***	cluster 2	10 212.1.200.46	212.1.200.0/24
5 157.130.0.177	157.130.0.0/16	11 188.1.18.61 12 188.1.88.2	188.1.0.0/16
6 152.63.25.190 7 152.63.17.246 8 152.63.24.29	152.63.0.0/16	13 130.75.9.250 14 130.75.9.253 15 130.75.2.24	130.75.0.0/16

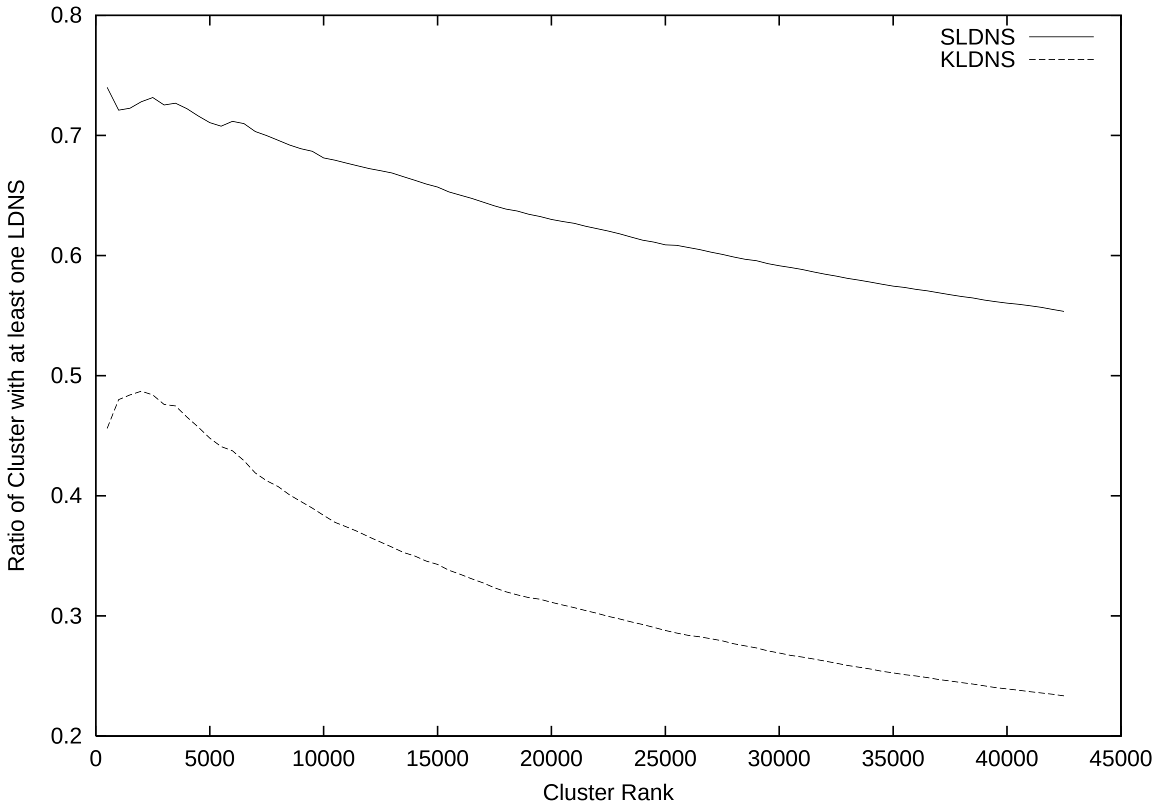
DNS servers in busy clusters

- Extract DNS traffic from *netflow*
- Identify suspected local DNS servers
- Cluster them and look for match in busy clusters from portal logs
- Establish lower bounds of DNS servers' presence in busy clusters

Web/DNS data examined for clustering application

	Server log	Suspected LDNS	Known DNS
IP addresses	7,652,670	564,077	58,527
# of Clusters	114,928	69,923	20,736
# of busy clusters	43,164	–	–

Looking for DNS servers in busy clusters



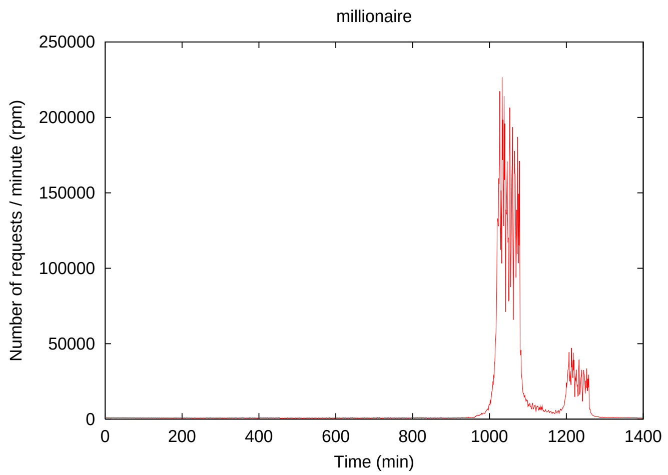
Lower bound on DNS servers in busy clusters

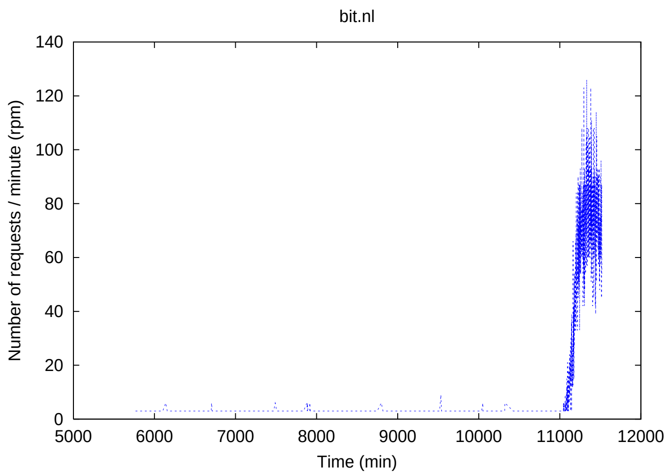
Peer to Peer applications

- New model of content distribution: end-hosts self-organize and share content with each other (e.g. Napster, Gnutella, Freenet)
- Files are replicated among the overlay network on demand
- Need a scalable content location and routing scheme!
- Use clustering to group clients

  One more hierarchy
  Less dynamism
  More scalability

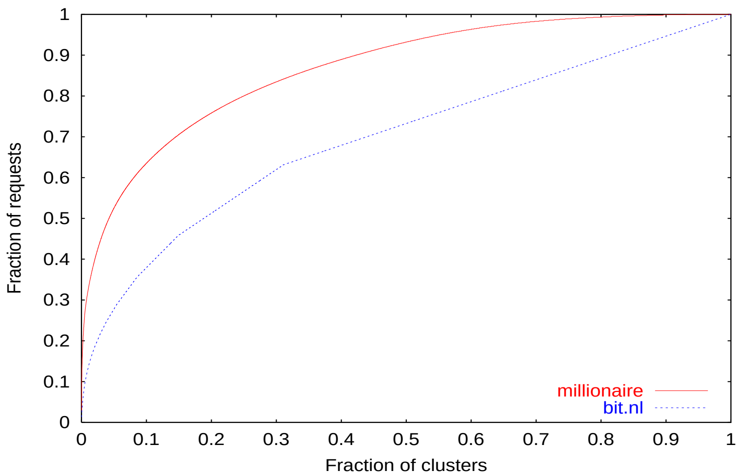
Flash Crowds vs. DDoS attacks





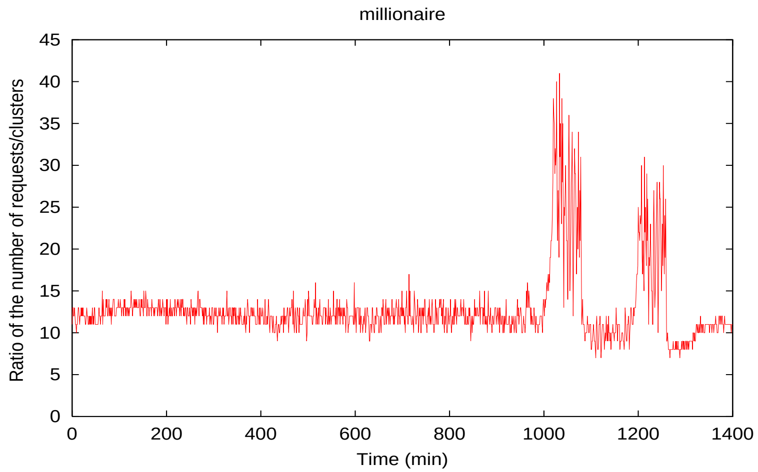
- A sudden growth in the number of requests
  - millionaire: 9M requests in 100 mins
  - bit.nl: 35K requests in 430 mins

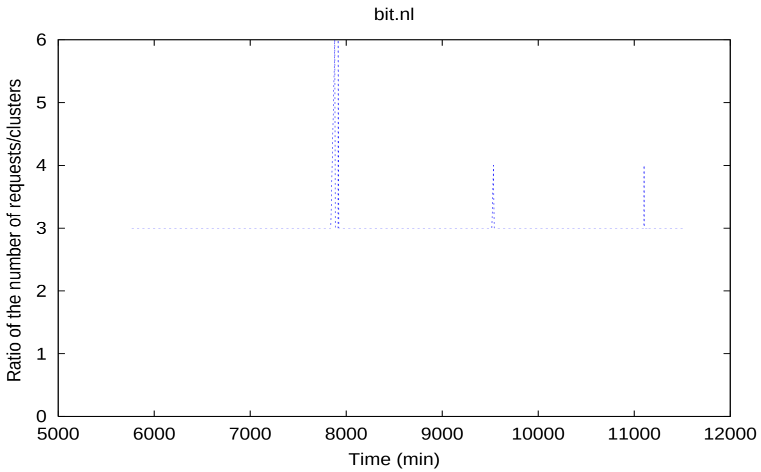
Distribution of Web Clients



- millionaire: source of requests can be better aggregated into clusters during flash events
- bit.nl: the distribution of DDoS attackers is broad

The number of requests per client





- millionaire: A small number of busy clusters account for most of requests
- bit.nl: Each client sends 3 or 6 requests

Ongoing work: Validation and refinement of clusters

- Do sample validation of a cluster. Chernoff bounds: no more than a certain fraction of clients are misclustered. Cheaper but probabilistic validation.
- Need to refine clusters that fail validation
- Subclustering based on last 2 hops of all clients is too expensive
- Run traceroute on a small random sample to capture 2 or more clients from each sub-cluster; pick up largest common prefix of clients selected from each subcluster - represents that particular subcluster. Assign remaining clients to these subclusters based on LPM.

Summary

- Possible to learn about network from outide in automated fashion
- Ongoing work includes an online algorithm (ultrafast LPM)
- Can use BGP routing table snapshots to cluster $> 99\%$ of clients in the server logs and identify proxies/spiders
- Very conservative error rate is $< 10\%$, immune to BGP dynamics

Acknowledgments/collaborators

Data providers: AltaVista, Apache, Chile, CodeRed ...

Application collaborators;

DNS : Chuck Cranor/Emden Gansner/Oliver Spatscheck (ATT)

Flash crowds: Jayeon Jung (MIT), Misha Rabinovich (ATT)

LPM: Glenn Fowler, Adam Buchsbaum, Phong Vo (ATT)

P2P: Yinglian Xie (CMU), Jia Wang (ATT)

Power laws of clusters: Christos Faloutsos (CMU), Jia Wang (ATT)

Sampling: Gagan Aggarwal (Stanford), Suresh Venkat (ATT)

Clustering software platform: Deepay Chkaravarthy (CMU), Jia Wang (ATT)