Mastodon Network Analysis

Ved Kothavade
University of Maryland, College Park
College Park, MD
vedk@terpmail.umd.edu

Phan Ngyuen
University of Maryland, College Park
College Park, MD
phan2003@terpmail.umd.edu

***Abstract*—The Mastodon network is a large area of research for sociologists and computer scientists alike, due to its distributed nature and unique audience. In this report, we crawl Mastodon network and measure statistics about the nature of the instances that make it up, and the connections between them. There exists research done on the content of the posts on Mastodon instances, but there is none to be found regarding the nature of the servers that run instances themselves, nor their connectivity with each other.**

I. Introduction

We want to explore characteristics of the Mastodon network related to the geographical distribution of nodes, cloud providers nodes are hosted on, and more [1]. This paper is largely inspired by "Design and evaluation of IPFS: a storage layer for the decentralized web", which demonstrates the kinds of measurments that could be done on a decentralized network [2].

II. Methodology

While writing a crawler is simple in principle, to mitigate time constraints we began with a seed list of alive nodes created by an external, open-source Fediverse crawler [3]. This crawler is updated every 6 hours. From the list created by this crawler, we filter down to Fediverse nodes hosting Mastodon or Mastodon-compatible software (as opposed to, for example, WordPress), and then to nodes which supported the `peers` API we leverage to determine node connectivity.

From this filtered list of nodes, we then collected data about each node leveraging the MaxMind GeoIP databases and by resolving the IPs for the domains, and looking up their ASN, AS organization names, country codes [4]. Furthermore, we use an `instance` API to collect user and post counts for all instances.

Upon collecting this data about nodes in a database, we use the neo4j graph database to insert all nodes, the properties of these nodes, and finally create `PEERS_WITH` relationships between peers as defined by the aforementioned `peers` API [5], [6].

We believe that this is a representative dataset of the Mastodon network, and thus feel that the analyses we conduct from this graph dataset represent the true network well. This is supported by the fact that statistics published by the Mastodon project state that there are 8739 instances and 9651558 users, which we are close to–especially considering that we count non-Mastodon software which supports the protocol [7].

In the collection and analysis of this data, we created a Go program which connects to peers to ensure they are alive, determines whether their software is Mastodon API compatible, and then uses the `instance` API to collect user and post counts for all instances. This program further creates a number of SQLite databases and CSV files which are used for the analysis in this paper, directly or via neo4j. In addition, we created a number of Python scripts to process the data and create some of the visualizations, using Pandas, Mathplotlib, and GeoPandas. These are all available on GitHub [8].

III. Analysis

A. Location of Instances

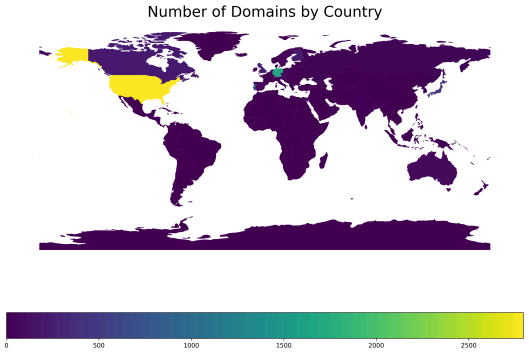

Fig. 1: Choropleth of Instance Locations

TABLE I: Instance Counts in Top 10 Countries

Rank	Country	Count	Share
1	US	2795	30.72%
2	FR	1735	19.07%
3	DE	1700	18.69%
4	PT	652	7.17%
5	JP	415	4.56%
6	GB	249	2.74%
7	FI	234	2.57%
8	CA	229	2.52%
9	NL	181	1.99%
10	SG	138	1.52%
Total		9097	

As Table I shows, the majority of instances are in the United States, followed by France, Germany, Portugal, and Japan. In fact, the top 5 countries alone account for 80.21% of all instances.

TABLE II: Countries with Highest Average Posts per User

Country	Total Posts	Total Users	Number of Instances	Avg Posts per User
NL	105269893	102337	181	1028.66
JP	70037027	73338	415	954.99
KR	24614210	50073	41	491.57
CR	191040	461	4	414.4
DK	76087	219	19	347.43
SG	5174483	17407	138	297.26
RO	1038226	3802	10	273.07
BR	204996	763	9	268.67
ZA	100546	421	8	238.83
CA	5990997	32453	229	184.61
PT	5346764	34050	652	157.03
LT	224167	1725	10	129.95
AU	4081321	32007	78	127.51
SE	248521	2079	48	119.54
GB	5264831	45125	249	116.67

When looking at the countries with the highest average posts per user, we see that the Netherlands and Japan are far above the rest, with their average posts per user being approximately twice as high as the next highest country, South Korea. Furthermore, both of these countries have a meaningful number of instances, incidiating that this is not due to a few outliers.

B. Cloud Providers

TABLE III: Instance Counts by Cloud Provider

Rank	Cloud Provider	Count	Share
1	None	6200	59.33%
2	OVH	2245	21.48%
3	Hetzner	1207	11.55%
4	DigitalOcean	493	4.72%
5	AWS	200	1.91%
6	GCP	68	0.65%
7	Azure	37	0.35%
Total		10450	

Most instances are not hosted on cloud providers, and among those that are, surprisingly the most common are none of the big 3, but instead OVH, Hetzner, and DigitalOcean. In total, we see that 40.67% of instances are hosted on cloud providers, indicating that the majority of instances are run on personal servers. This suggests that the Fediverse's decentralized nature extends beyond just its social structure to its infrastructure, with many operators choosing to maintain their own hardware rather than relying on major cloud providers.

C. Autonomous Systems

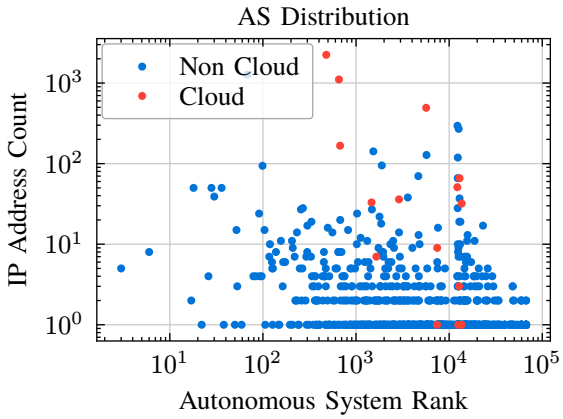

Fig. 2: Distribution of IPs across ASes according to their size (measured by their AS rank).

We mapped IP addresses to Autonomous Systems using GeoLite2, found the CAIDA AS Rank for each AS. According to CAIDA, the AS Rank is a measure of the influence of an AS to the global routing system, as calculated by their sizes, peering agreements, and more [9].

We then plotted the number of IP addresses per AS, colored by whether the AS is a cloud provider or not. We see that the cloud providers are disproportionately large, and that there are many more IP addresses in the cloud providers than in the non-cloud providers.

The graph reveals that most Mastodon instances are hosted on high-ranking AS systems (higher numbers indicate lower influence in the global routing system), suggesting that the Fediverse infrastructure largely relies on smaller, less central network operators rather than the internet backbone providers. This aligns with the decentralized ethos of the Mastodon network, with many instances avoiding both major cloud providers and core internet infrastructure

TABLE IV: Autonomous Systems covering $> 50\%$ of all found IP Addresses

ASN	AS Rank	AS Name	Count	Share
479	OVH	16276	2242	21.45%
68	CLOUDFLARENET	13335	1273	12.18%
655	HETZNER-AS	24940	1108	10.6%
5664	DIGITALOCEAN-ASN	14061	493	4.72%
12281	AKAMAI-LINODE-AP	63949	295	2.82%
Total			10450	51.78%

D. Most Peered Instances

TABLE V: Top 10 Most Peered Instances

Instance	Peers	Users	Posts
mastodon.social	16821	2734739	131726225
mastodon.online	16464	189377	10608587
mstdn.social	16439	261299	20088443
mas.to	16421	183971	10976052
fosstodon.org	16300	62442	4240866
chaos.social	16274	13005	8360180
hachyderm.io	16274	56449	4074546
infosec.exchange	16215	75993	4361671
mastodon.gamedev.place	16188	33514	1721158
social.tchncs.de	16178	22857	3573231
mastodon.world	16171	191803	7250207

From Table V, we see that the most peered instance is `mastodon.social`, with 16464 peers. There appears to be a trend that the more peered an instance is, the more users and posts it has, but this is not strict.

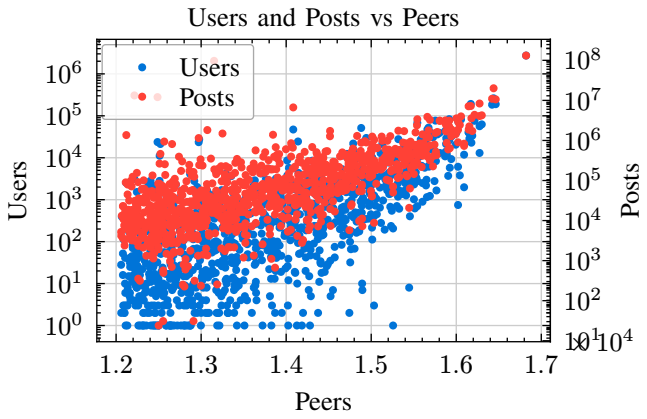

Fig. 3: Users and Posts vs Peers for 1000 Most Peered Instances

We see that there is a positive correlation between the number of peers and the number of users and posts in Fig. 3.

E. Average Path Length

TABLE VI: Average path length between any 2 points

Statistic	Value
Est. Avg Path Length	1.58
Min Length	1.0
Max Length	3.0
Standard Deviation	0.49
Pairs Considered	4537592

To estimate the Mastodon network's average shortest-path length, we randomly sample 1000 nodes of 9000 as source nodes. For each source, we run Dijkstra's and find the path length to every other node. Finally, we aggregate the sampled distances with `avg()`, `min()`, `max()`, and `stdev()` functions to yield estimates of the global mean, minimum, maximum, and standard deviation of path lengths. The result is an average path lenngth of 1.58 with standard deviation of 0.49.That's exceptionally low, and most pairs are either directly peering or share exactly one intermediary. This indicates a very well connected network. With the same number of nodes and average number of degree, the Erdős–Rényi null model tells predicts the average path length between any 2 nodes to be around 1.13, effectively making this an almost complete graph. Due to how tightly connected this network is, we can also make the assumption peers aren't selected based on having similar content, since everyone is connected to one another.

F. Community Detection Limitations

We attempted to apply several community detection algorithms from Neo4j's Graph Data Science (GDS) library to identify potential clusters within the Mastodon network [10]. However, all these attempts proved unsuccessful due to the exceptionally high connectivity of the network as evidenced by the average path length statistics in Table VI.

The Weakly Connected Components (WCC) algorithm, which identifies groups of nodes that are connected to each other through any path regardless of direction, failed to produce meaningful results. With an average path length of 1.58 and a maximum of only 3.0, nearly all nodes were placed into a single giant component, providing no useful partitioning of the network.

Similarly, the Louvain Modularity algorithm, which detects communities by optimizing modularity (the density of connections within communities versus connections between communities), could not identify distinct communities. The algorithm relies on finding areas with higher internal than external connectivity, but with most instances being only one or two hops away from any other instance, there were no clear boundaries for community formation.

The Label Propagation algorithm, which works by propagating labels through the network and forming communities of nodes that share the same label, also failed to converge to meaningful communities. The extremely high interconnectivity meant that labels propagated too quickly across the entire network, again resulting in essentially a single community.

These failures highlight the unique structure of the Mastodon network: despite being decentralized in terms of governance and operation, its instance connectivity patterns resemble those of a nearly complete graph. This suggests that while Mastodon instances may vary widely in terms of content focus, user base, and operation, they maintain a remarkably high level of technical interconnection, prioritizing comprehensive network access over community-based peering.

IV. Conclusion

In this paper, analyzed the Mastodon network, examining its geographical distribution, infrastructure characteristics, and connectivity patterns. Our findings reveal several important insights about this decentralized social media platform.

First, we found that Mastodon instances are predominantly concentrated in a handful of countries, with the United States, France, Germany, Portugal, and Japan hosting the ma-

jority. This geographical concentration suggests that despite Mastodon's decentralized architecture, its adoption remains uneven globally. Given the Fediverse's reputation as a network largely consisting of technical, niche communities, this meets our expectations.

Second, our infrastructure analysis revealed that most Mastodon instances are not hosted on major cloud providers but are instead operated on personal servers or smaller hosting services (40.67% are on the cloud). This aligns with the Fediverse's ethos of decentralization and independence from large corporate infrastructure. This is further supported by the fact that when the AS ranks of instances are plotted, the majority have high ranks, indicating that they are not on core internet infrastructure.

Third, and perhaps most surprising, was our discovery of the Mastodon network's remarkably high connectivity. With an average path length of just 1.58 between any two instances, the network resembles an almost complete graph where most nodes are either directly connected or separated by a single intermediary. This high level of connectivity rendered traditional community detection algorithms ineffective, as the network lacks the distinct clusters or communities typically found in other social networks.

These findings suggest that while Mastodon achieves decentralization in terms of governance and instance ownership, its technical connectivity pattern is highly integrated. Future work could explore the evolution of this connectivity over time, investigate the social and content-based relationships between instances beyond technical connections, and examine how this high connectivity affects content moderation and information diffusion across the network. One thing we'd have loved to explore, but did not have the time for, was the nature of instances that are blocked from the rest of the network–do they form their own communities?

References

[1] Mastodon, "Mastodon." 2025.

[2] D. Trautwein *et al.*, "Design and evaluation of IPFS: a storage layer for the decentralized web," in *Proceedings of the ACM SIGCOMM 2022 Conference*, in SIGCOMM '22. Amsterdam, Netherlands: Association for Computing Machinery, 2022, pp. 739–752. doi: 10.1145/3544216.3544232.

[3] A. Batischev and S. Silviu, *Minoru/minoru-fediverse-crawler*. 2025. [Online]. Available: https://github.com/Minoru/minoru-fediverse-crawler

[4] MaxMind, Inc., "GeoLite2 Free Geolocation Data." 2025.

[5] Neo4j, "Neo4j." 2025.

[6] Mastodon, "Mastodon." 2025.

[7] Mastodon, "Mastodon." 2025.

[8] V. Kothavade and P. Ngyuen, *kothavade/mastodon-paper*. 2025. [Online]. Available: https://github.com/kothavade/mastodon-paper

[9] "AS Rank." doi: https://doi.org/10.21986/CAIDA.DATA.AS-RANK.

[10] Neo4j, "Neo4j Graph Data Science Library." 2025.