

May 09, 2023

BUILDING THE MICRO MIRROR FREE SOFTWARE CDN



As should surprise no one, based on my past projects of [running my own autonomous system](#), building [my own Internet Exchange Point](#), and [building a global anycast DNS service](#), I kind of enjoy building Internet infrastructure to make other people's experience online better. So that happened again, and like usual, this project got well out of hand.

Linux updates

You run `apt update` or `dnf upgrade` and your Linux system goes off and download updates from your distro and installs them. Most people think nothing of it, but serving all of those files to every Linux install in the world is a challenging problem, and it's made even harder because most Linux distributions are free and thus don't have a project budget to spin up a global CDN (Content Distribution Network) to have dozens or hundreds of servers dedicated to putting bits on the wire for clients over the Internet.

How Linux distros get around this budget issue is that they host a single "golden" copy of all of their project files (Install media ISOs, packages, repository index files, etc) and volunteers around the world who are blessed with surplus bandwidth download a copy of the whole project directory, make it available from their own web server that they build and maintain themselves, and then register their mirror of the content back with the project. Each free software project then has a load balancer that directs clients to nearby mirrors of the content they're requesting while making sure that the volunteer mirrors are still online and up to date.

At the beginning of 2022, one of my friends (John Hawley) and I were discussing the fact that the network who used to be operating a Linux mirror in the same datacenter as us had moved out of the building, and maybe it would be fun to build our own mirror to replace it.

John: "Yeah... it would probably be fun to get back into mirroring since I used to run the mirrors.kernel.org mirrors" (world's largest and most prominent Linux mirror)

Me: "Wait... WHAT?!"



So long story short, the two of us pooled our money together, and went and dropped \$4500 on a SuperMicro chassis, stuffed it full of RAM (384GB) and hard drives (6x 16TB) and racked it below the Google Global Cache I'm hosting in my rack in Fremont.

Like usual, I was posting about this as it was happening on Twitter (RIP) and several people on Twitter expressed interest in contributing to the project, so I posted a paypal link, and we came up with the offer that if you donated \$320

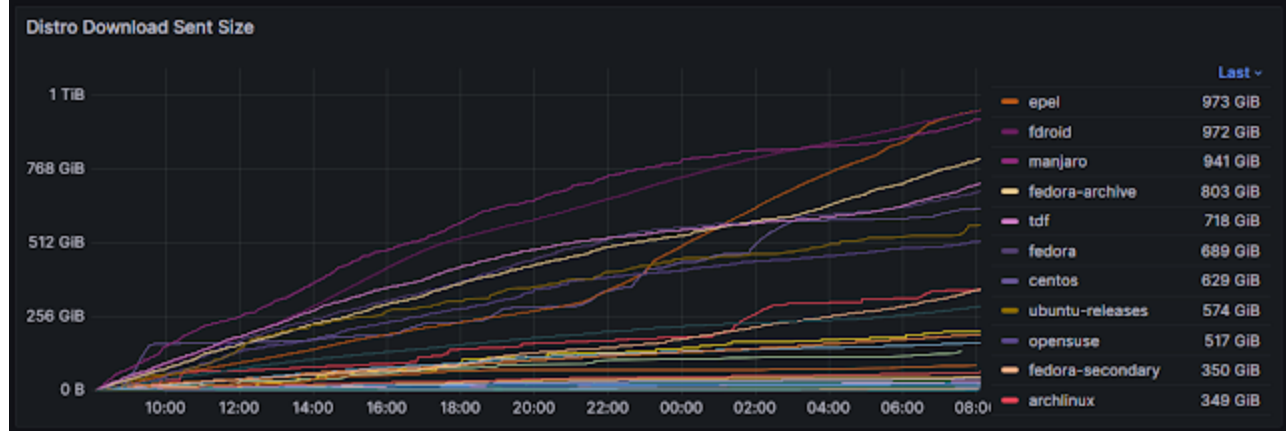
to the project, you'd get your name on one of the hard drives inside the chassis in Fremont, since that's how much we were paying for each of the 16TB drives.



This "hard drive sponsor" tier also spawned what I think was one of the most hilarious conversations of this whole project, where one of my friends was trying to grasp why people were donating money to get their name on a piece of label tape, stuck to a hard drive, inside a server, inside a locked rack, inside of a data center, where there most certainly was no possibility of anyone ever actually seeing their name on the hard drive. A rather avant-garde concept, I will admit.

The wild part was that we "sold out" on "Hard Drive Sponsor" tier donors, and enough people contributed to the project that we covered almost all of the hardware cost of the original mirror.fcix.net server!

So long story short, we decided to spin up a Linux mirror, fifty of my friends on Twitter chipped in on the project, and we were off to the races trying to load 50TB of Linux distro and free software artifacts on the server to get it up and running. All well and good, and a very typical Linux mirror story.



Where things started to get out of hand is when John started building a Grafana dashboard to parse all of the Nginx logs coming out of our shiny new Linux mirror and analyzing the data as to how much of what projects we were actually serving. Pivoting the data by various metrics like project and release and file type, we came to the realization that while we were hosting 50TB worth of files for various projects, more than two thirds of our network traffic was coming from a very limited number of projects and only about 3TB of files on disk! And this is where the idea of the Micro Mirror began to take shape.

The Micro Mirror Thesis

If the majority of the network traffic on a Linux mirror is coming from a small slice of the assets hosted on the mirror, then it should be possible to build a very small and focused mirror that only hosts projects from that "hot working set" subset and while less effective than our full sized mirror, could be only half as effective as our full size mirror at 10% of the cost.



So we set ourselves the challenge of trying to design a tiny Linux mirror which could pump out a few TB of traffic a day (as opposed to the 12-15TB/day of traffic served from mirror.fcix.net) with a hardware cost less than the \$320 that we spent on one of the hard drives in the main storage array. Thanks to eBay and my love for last gen enterprise thin clients, we settled on a design consisting of the following:

- HP T620 thin client
- 2x4GB DIMMs
- 2TB M.2 SSD

This could all be had for less than \$250 on eBay used, and conveniently fits nicely in a medium flat rate USPS box, so once we build it and find a random network in the US willing to plug this thing in for us, we can just drop it in the mail.

We built the prototype and one of my other friends in Fremont offered to host it for us, since we're only using the 1G-baseT NIC on-board the thin client, and we were off to the races. Setting the tiny mirror up only hosting Ubuntu ISOs, Extra Packages for Enterprise Linux, and the CentOS repo for servers easily exceeded our design objective of >1TB/day of network traffic. Not a replacement for traditional "heavy iron" mirrors that can host a longer tail of projects, but this is 1TB of network traffic which we were able to peel off of those bigger mirrors so they could spend their resources serving the less popular content, which we wouldn't be able to fit on the single 2TB SSD inside this box.



Now it just became a question of "well, if one Micro Mirror was pretty successful, exactly how many MORE of these little guys could we stamp out and find homes for???"

These Micro Mirrors have several very attractive features to them for the hosting network:

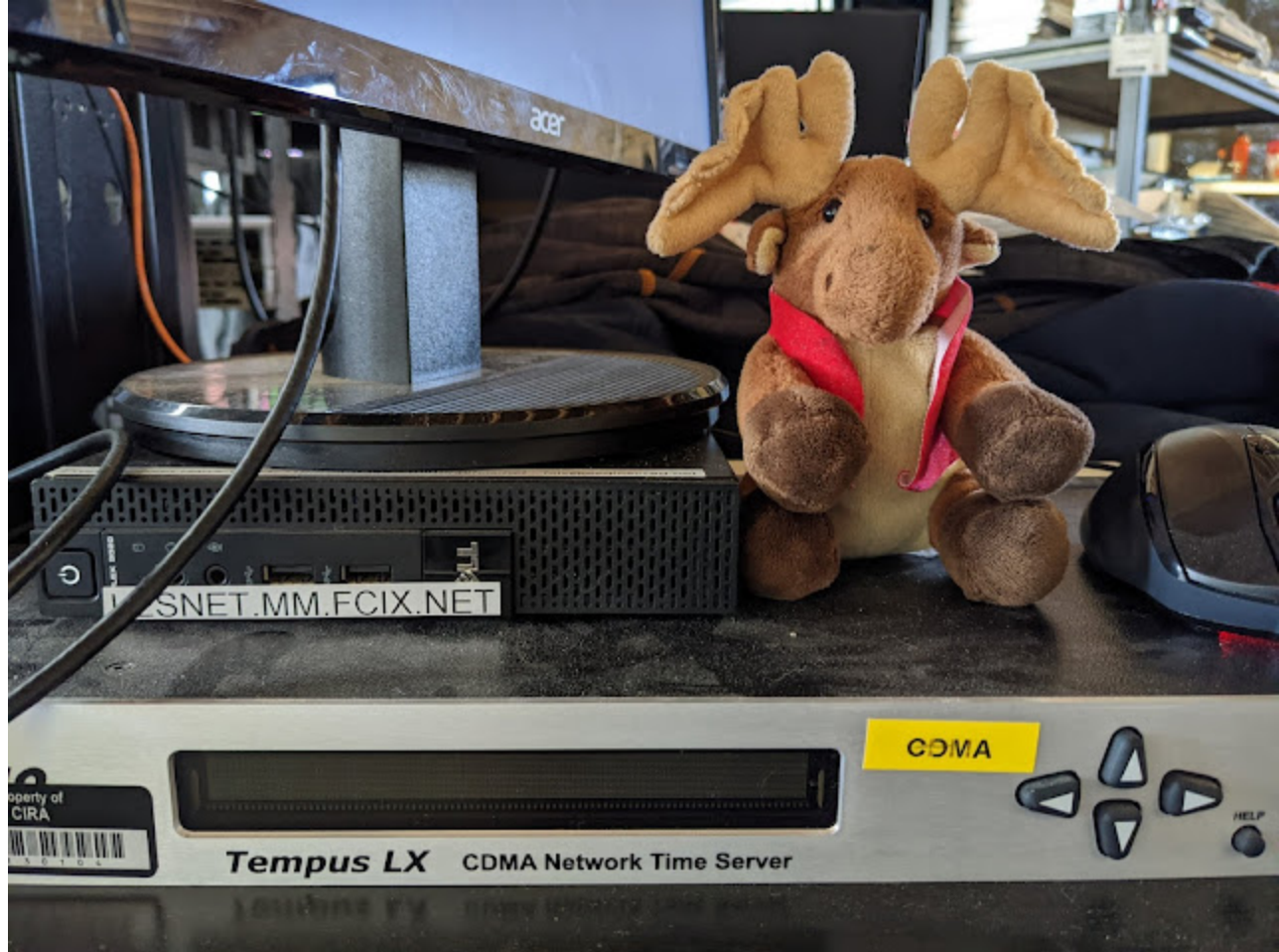
- They are fully managed by us, so while many networks / service providers *want* to contribute back to the free software community, they don't have the spare engineering resources required to build and manage their own mirror server. So this fully managed appliance makes it possible for them to contribute their network bandwidth at no manpower cost.
- They're very small and can fit just about anywhere inside a hosting network's rack.
- They're low power (15W)
- They're fault tolerant, since each project's load balancer performs health checks on the mirrors and if this mirror or the hosting network has an outage the load balancers will simply not send clients to our mirror until we get around to fixing the issue.

Then it was just a question of scaling the idea up. Kickstart file so I can take the raw hardware and perform a completely hands-off provisioning of the server. [Ansible playbook](#) to take a short config file per node and fully provision the HTML header, project update scripts, and rsync config per server, and suddenly I can fully stamp out a new Micro Mirror server with less than 30 minutes worth of total work.



Finding networks willing to host nodes turned out to be extremely easy. Between Twitter, Mastodon, and a few select Slack channels I hang out on, I was able to easily build a waiting list of hosts that surpassed the inventory of thin clients I had laying around. Then we just needed to figure out how to fund more hardware beyond what we were personally willing to buy. Enter [LiberaPay](#), an open source service similar to Patreon where people can pledge donations to us to keep funding this long term.

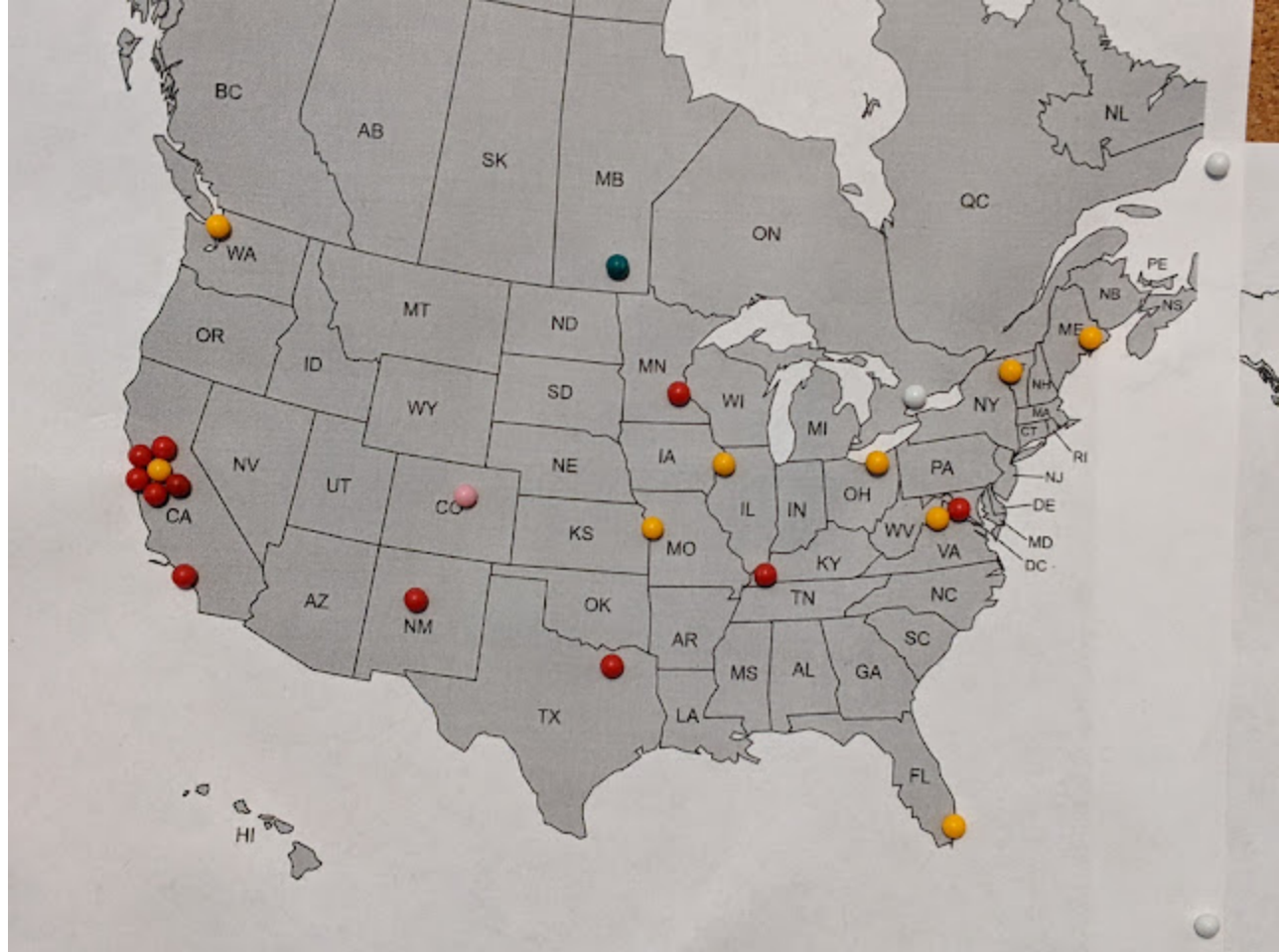
So now we have a continual (albeit very small) funding source, and a list of networks waiting for hardware, and it's mainly been a matter of waiting for enough donations to come in to fund another node, ordering the parts, provisioning the server, dropping it in the mail, and waiting for the hosting network to plug it in so we can run our Ansible playbook against it and get it registered with all the relevant projects.



So now we had a solid pipeline set up, and we could start playing around with other hardware designs than the HP T 620 thin client. The RTL8168 NIC on the T 620s are far from ideal for pumping out a lot of traffic, and we actually got feedback from several hosting networks that they just don't have the ability to plug in baseT servers anymore, and they'd much prefer a 10Gbps SFP+ NIC handoff to the appliance.



The desire for 10G handoffs has been a bit of a challenge while still trying to stay within the \$320 hardware budget goal we set for ourselves, but we have been doing some experiments with the HP T 620 Plus thin client, which happens to have a PCIe slot that fits a Mellanox ConnectX3 NIC, and we also received a very generous donation of a pile of Dell R220 servers with 10G NICs from Arista Networks (Thanks Arista!)



So now the project has very easily gotten out of hand. We have more than 25 Micro Mirror nodes of various permutations live in production, spanning not only North America but several of the nodes have been deployed internationally. Daily we serve roughly 60-90TB of Linux and other free software updates from these Micro Mirrors, with more than 150Gbps of port capacity. So while not making a profound difference to user experience downloading updates, each Micro Mirror we deploy has helped make a small incremental improvement in how fast users are able to download updates and new software releases.

So if you've started noticing a bunch of *.mm.fcix.net mirrors for your favorite project, this is why. We hit a sweet spot with this managed appliance and have been stamping them out as resources permit.

Interest in Helping?

The two major ways that someone can help us with this project is funding the new hardware and providing us locations to host the Micro Mirrors:

- Cash contributions are best sent via my [LiberaPay account](#).
- Any service providers interested in hosting nodes in their data center network can reach out to mirror@fcix.net to contact us and get on our wait list.

We are not interested in deploying these nodes off of any residential ISP connections, so even if you have a 1Gbps Internet connection from your ISP, we want to limit the deployment of these servers to wholesale transit contexts in

data centers where we can work directly with the ISP's NOC.

Of course, nothing is preventing anyone from going out and setting up your own Linux mirror. Ultimately having more mirror admins out there running their own mirrors is better than growing this Micro Mirror project for the sake of diversity. If you're looking to spin up your own mirror and have any specific questions on the process, feel free to reach out to us for that as well.

I also regularly [post about this project on Mastodon](#), if you want to follow along real time.

[Share](#)

POPULAR POSTS



November 15, 2017

CREATING AN AUTONOMOUS SYSTEM FOR FUN AND PROFIT

[Share](#)

May 05, 2021

UNLOCKING THIRD PARTY TRANSCEIVERS ON OLDER ARISTA SWITCHES

[Share](#)

Powered by [Blogger](#)



Kenneth Finnegan

[VISIT PROFILE](#)

[Archive](#)

