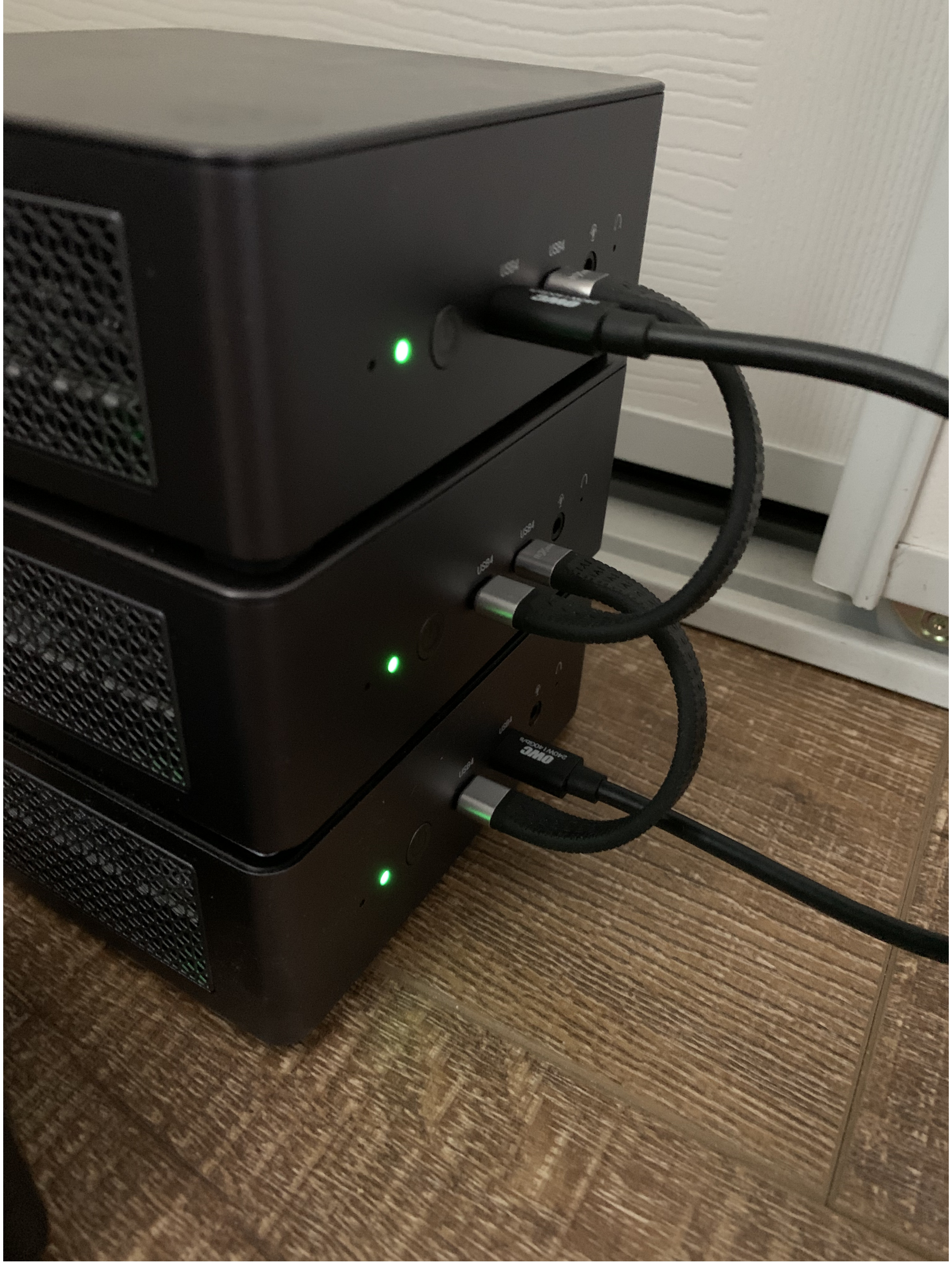
High-speed 10Gbps full-mesh network based on USB4 for just $47.98

Jan 14, 2024

10 minute read

As a software engineer, software is part of your job title; thus, it almost feels like you should only know software. However, in the decades of building software, I realized that gaining knowledge about hardware is equally important to learning code. Although I might never be as good as an expert in hardware, I want to expand myself beyond just software. So, I never shy away from getting my hands dirty with hardware.

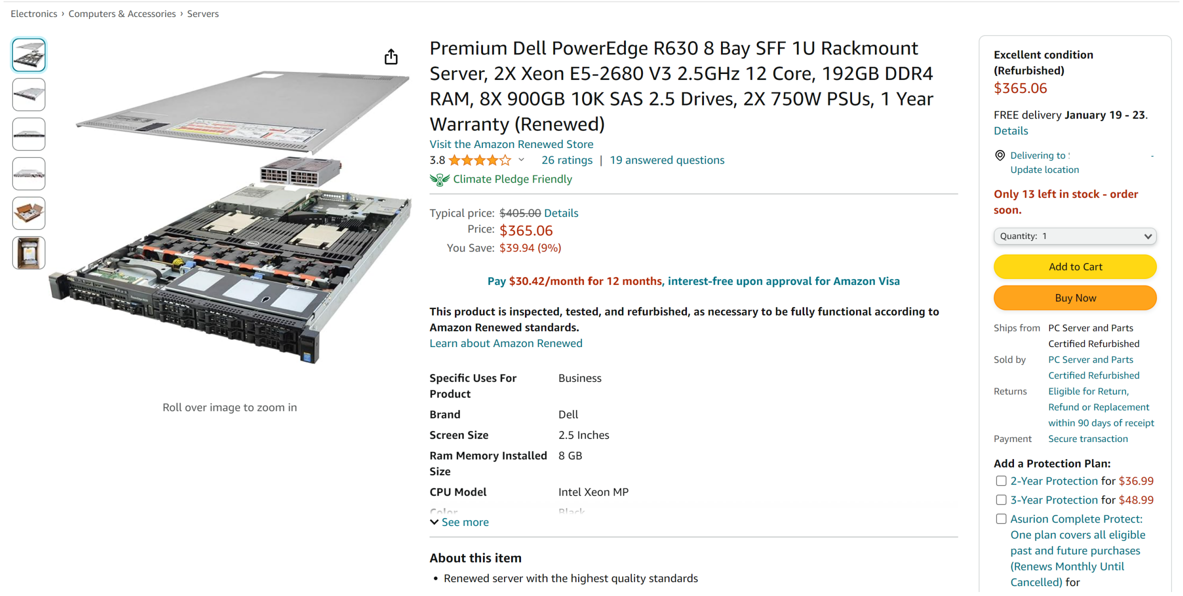
To reduce the cost of my AWS cloud service, I recently decided to move some less mission-critical services into my bare-metal servers. Therefore, I got to learn how to build a bare-metal Kubernetes cluster and set up the network for it. After some research, trial, and error, I finally built and ran a relatively low-cost cluster with a high-speed full-mesh interconnected network. The most interesting part is that the networking is based on a USB4 ethernet bridge instead of a conventional ethernet switch and cables. I tested the network speed, and it can hit 11Gbps. The cost of making the network is only $47.98 USD! Today, I would like to share my experience of building it.



The UM790 Pro bare-metal cluster with full-mesh USB4 cables connections

Standard 1U servers vs mini PCs

When I thought of building a bare-metal cluster, I first asked myself what type of machine to use. The first idea that came to my mind was to purchase some retired used 1U servers. They are unbelievably cheap. Take this refurbished Dell PowerEdge R630 selling on Amazon, for example, and it costs only $380.42


Electronics › Computers & Accessories › Servers

Premium Dell PowerEdge R630 8 Bay SFF 1U Rackmount Server, 2X Xeon E5-2680 V3 2.5GHz 12 Core, 192GB DDR4 RAM, 8X 900GB 10K SAS 2.5 Drives, 2X 750W PSUs, 1 Year Warranty (Renewed)

Visit the Amazon Renewed Store

3.8 ★★★★☆ 26 ratings | 19 answered questions

Climate Pledge Friendly

Typical price: ~~$405.00~~ Details

Price: $365.06

You Save: $39.94 (9%)

Pay $30.42/month for 12 months, interest-free upon approval for Amazon Visa

This product is inspected, tested, and refurbished, as necessary to be fully functional according to Amazon Renewed standards.

Learn about Amazon Renewed

Specific Uses For Product	Business
Brand	Dell
Screen Size	2.5 Inches
Ram Memory Installed Size	8 GB
CPU Model	Intel Xeon MP
Color	Black

See more

About this item

- Renewed server with the highest quality standards

Roll over image to zoom in

Excellent condition (Refurbished)

$365.06

FREE delivery **January 19 - 23**.

Details

Delivering to [redacted]

Update location

Only 13 left in stock - order soon.

Quantity: 1

Add to Cart

Buy Now

Ships from	PC Server and Parts Certified Refurbished
Sold by	PC Server and Parts Certified Refurbished
Returns	Eligible for Return, Refund or Replacement within 90 days of receipt
Payment	Secure transaction

Add a Protection Plan:

- ☐ 2-Year Protection for $36.99
- ☐ 3-Year Protection for $48.99
- ☐ Asurion Complete Protect: One plan covers all eligible past and future purchases (Renews Monthly Until Cancelled) for

refurbished Dell PowerEdge R630 selling on Amazon

The CPU is a bit outdated, but it has two of them, and it comes with 192GB DDR4 RAM, so it's still a fairly powerful machine. While the machine itself isn't expensive, it is not cheap if you consider the cost to operate. A machine like this is very power-hungry. Suppose the power consumption is 1000W per hour. Given California's average residential electricity rate is 15.34 cents per kWh, it would be

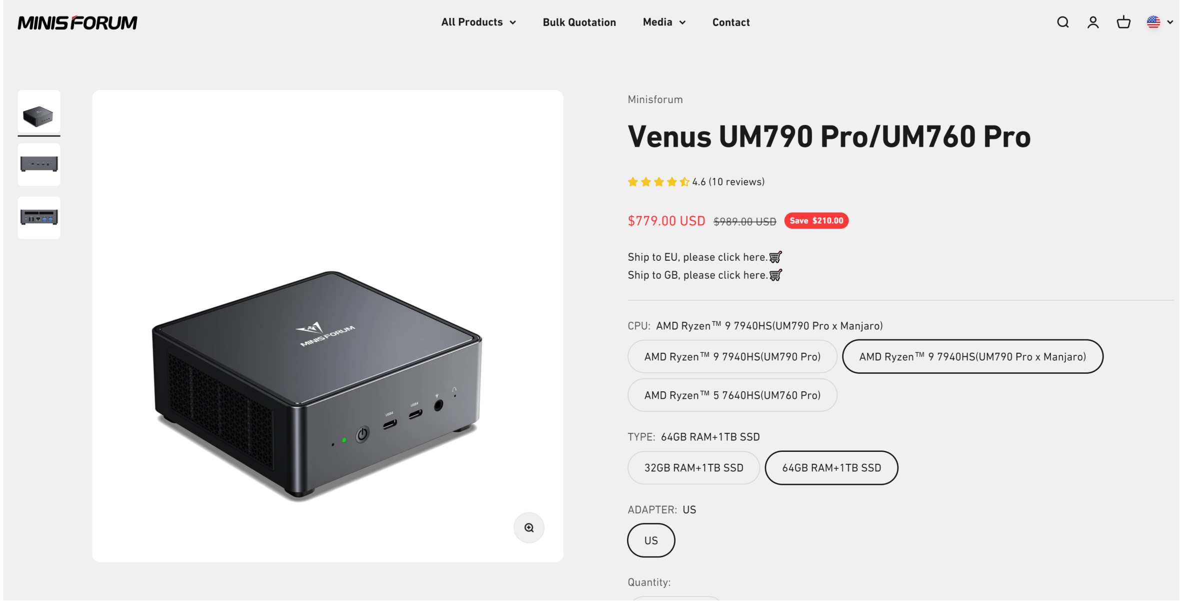
```
1000W * 24 hour * 30 days = 720 kWh.
```

In other words, the costs of running just one of these machines could be

```
720 kWh * 15.34¢/kWh = $110.45 USD per month
```

And that would be $1,325.4 USD per year. You can buy more than three of these servers with just the money you pay for the electric bill. Not to mention they will make huge noise and it's not ideal to keep them in your living area. There are also cooling issues you may need to consider; there will be an extra cost if you need an active cooling solution if you are stacking them up and the generated heat needs to go somewhere. Considering these, I ruled out buying one of those 1U servers pretty soon.

The pace of modern hardware improvement is insanely fast. I've been paying attention to the trend of how powerful tiny devices can run on extremely low power nowadays. Apple was leading the way by introducing their M1 chip, bringing mighty computing power with super low power consumption. Thanks to the competition, Mini PCs are becoming more and more powerful these days. They are also fairly cheap, quiet, and consume little power compared to a full-size PC or a server. I looked at different mini PCs and found this one, Venus UM790 Pro from Miniforums:



Minisforum

Venus UM790 Pro/UM760 Pro

4.6 (10 reviews)

$779.00 USD ~~$989.00 USD~~ Save $210.00

Ship to EU, please click here.
Ship to GB, please click here.

CPU: AMD Ryzen™ 9 7940HS(UM790 Pro x Manjaro)

AMD Ryzen™ 9 7940HS(UM790 Pro) [✓ AMD Ryzen™ 9 7940HS(UM790 Pro x Manjaro)] AMD Ryzen™ 5 7640HS(UM760 Pro)

TYPE: 64GB RAM+1TB SSD

32GB RAM+1TB SSD [✓ 64GB RAM+1TB SSD]

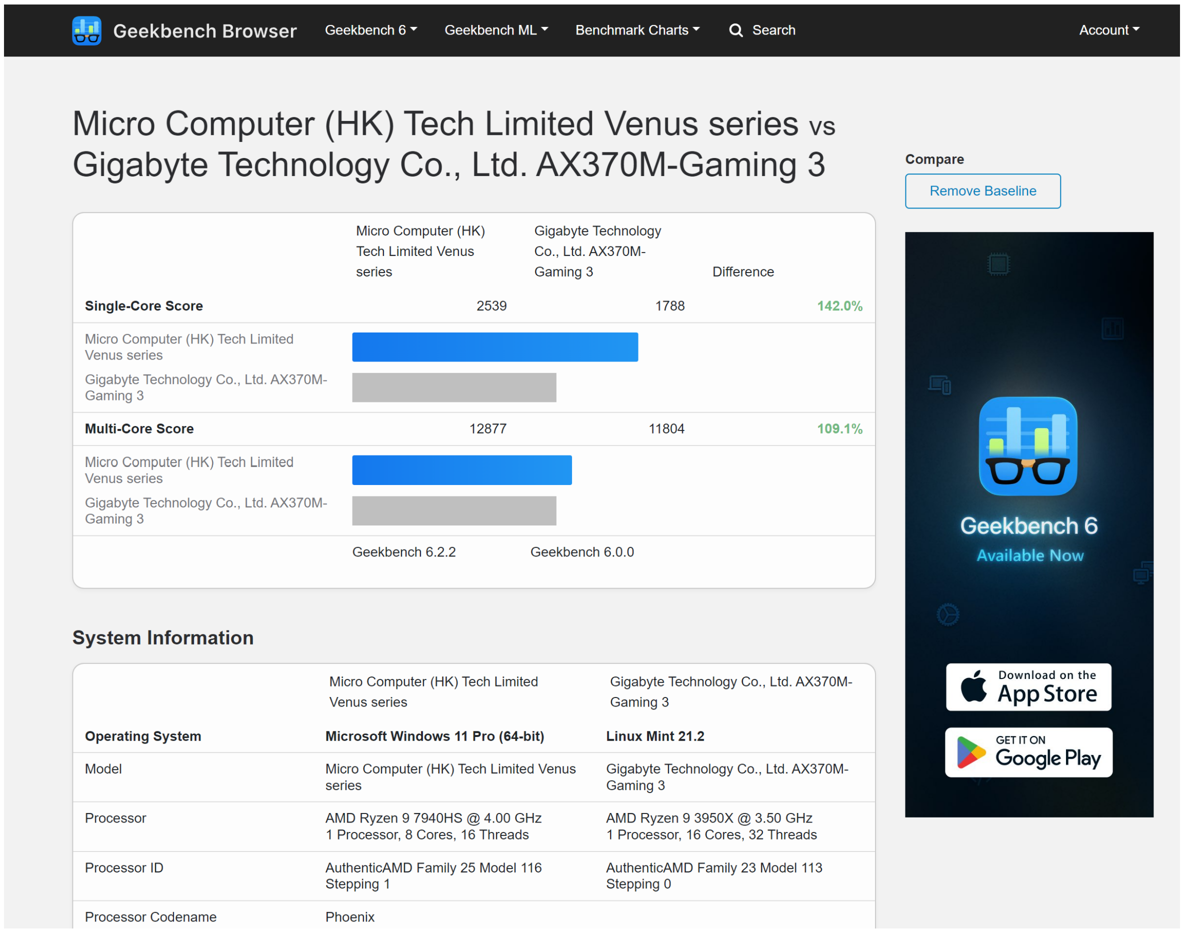
ADAPTER: US

[✓ US]

Quantity:

UM790 Pro product page screenshot

The machine itself isn't expensive. A top spec with 64GB memory and 1T storage only set you back $800 USD. It comes with an AMD Ryzen 9 7940HS CPU. It equips a CPU built for a laptop, so the power consumption is also pretty low. According to this YouTube video reviewing that machine, its idle power is only around 6W, and when running full load, it only consumes 80W. I ran a benchmark on this machine, and it blew my mind 🤯



Micro Computer (HK) Tech Limited Venus series vs Gigabyte Technology Co., Ltd. AX370M-Gaming 3

	Micro Computer (HK) Tech Limited Venus series	Gigabyte Technology Co., Ltd. AX370M-Gaming 3	Difference
Single-Core Score	2539	1788	142.0%
Multi-Core Score	12877	11804	109.1%
	Geekbench 6.2.2	Geekbench 6.0.0	

System Information

	Micro Computer (HK) Tech Limited Venus series	Gigabyte Technology Co., Ltd. AX370M-Gaming 3
Operating System	**Microsoft Windows 11 Pro (64-bit)**	**Linux Mint 21.2**
Model	Micro Computer (HK) Tech Limited Venus series	Gigabyte Technology Co., Ltd. AX370M-Gaming 3
Processor	AMD Ryzen 9 7940HS @ 4.00 GHz 1 Processor, 8 Cores, 16 Threads	AMD Ryzen 9 3950X @ 3.50 GHz 1 Processor, 16 Cores, 32 Threads
Processor ID	AuthenticAMD Family 25 Model 116 Stepping 1	AuthenticAMD Family 23 Model 113 Stepping 0
Processor Codename	Phoenix	

Geekbench score compare between UM790 Pro and a PC with AMD Ryzen 9 3950x CPU

This tiny machine's benchmark score is even better than the top-spec PC I built three years ago with an AMD Ryzen 9 3950x at a fraction of the price and power consumption. If the numbers are correct, this machine only costs $9 USD monthly at full load!

Networking

It didn't take too much longer for me after I purchased the first UM790 Pro and tried it out to decide to extend it to a three-node cluster. But soon after bootstrapping the Kubernetes cluster on these three tiny beasts and installing Ceph as the storage system, I realized I needed a better interconnection between these machines. When Ceph moves big files between the nodes, it takes a long time with just 1G ethernet. The UM790 Pro machine has a 2.5G ethernet port, but my router's LAN ports are only 1G speed, so I was considering buying a 2.5G ethernet switch. It's not the end of the world to run a cluster at 1G speed, but limited bandwidth between nodes limits what you can do with the cluster, so ideally, I still want a higher-speed network between the nodes.

Interestingly, it appears that nowadays, you can get a 2.5G ethernet switch at a reasonably cheap price, something like $100. But those ones are usually from China, with different brand names sharing the same underlying machine, like the ones reviewed in this video.

Why there are so many brand names for the same machine? I guess it's a strategy the manufacturer adopts to have many brands for the same product so that they have more entries appear in the search results on Amazon. Therefore, you get more exposure and, thus, a better chance of a conversion. In the long run, I am speculative about the quality of those cheap 2.5G ethernet switches and the service they will provide. The products were sold under a seemingly throwable brand name, after all. Usually, I would prefer a more established brand if I had to buy one.

While I was debating which switch to buy, doing my research regarding the brand, price, and my requirements, I realized

> *wait a minute* 🤔

There are two USB4 ports on the machine. In theory, it could provide up to 40Gbps speed. Who cares about 2.5G? That's 40Gbps we are talking about here! Considering the money spent on a 2.5G ethernet router plus some Cat6 ethernet cable, why not just make a full-mesh network with USB4 cables? With that in mind, I soon purchased two of these and this one USB4 cable. That's only $47.98 USD in total, and yet it can hit 11Gbps! It would be way more expensive and slower if I went the ethernet route.

Configure the mesh network with NixOS and Systemd

As you've seen in the first picture in this article, the three nodes were fully connected with high-speed USB4 cables. Connecting cables is easy, but the question is, how do you configure the network in Linux? In the process of bootstrapping my Kubernetes cluster, I learned how to use NixOS to configure a reproducible Linux OS environment. It saved me a tremendous amount of trouble for configuring my nodes. NixOS is a package system that comes with a build system. It allows you to build reproducible packages all the way from Linux kernel to all the tiny utility command line tools. So, if you encounter any issues in the package, you can patch them quickly without waiting for the bug to be fixed in the upstream. NixOS and the whole Nix ecosystem deserve their own articles. I may write a “NixOS Explained” like my previous Elliptic Curve Cryptography Explained article. I found the package system beautifully designed but hard to understand at first glance. Hopefully, I can find the time 😅

Anyway, here’s the sample NixOS configuration I wrote for configuring the USB4 full-mesh network:

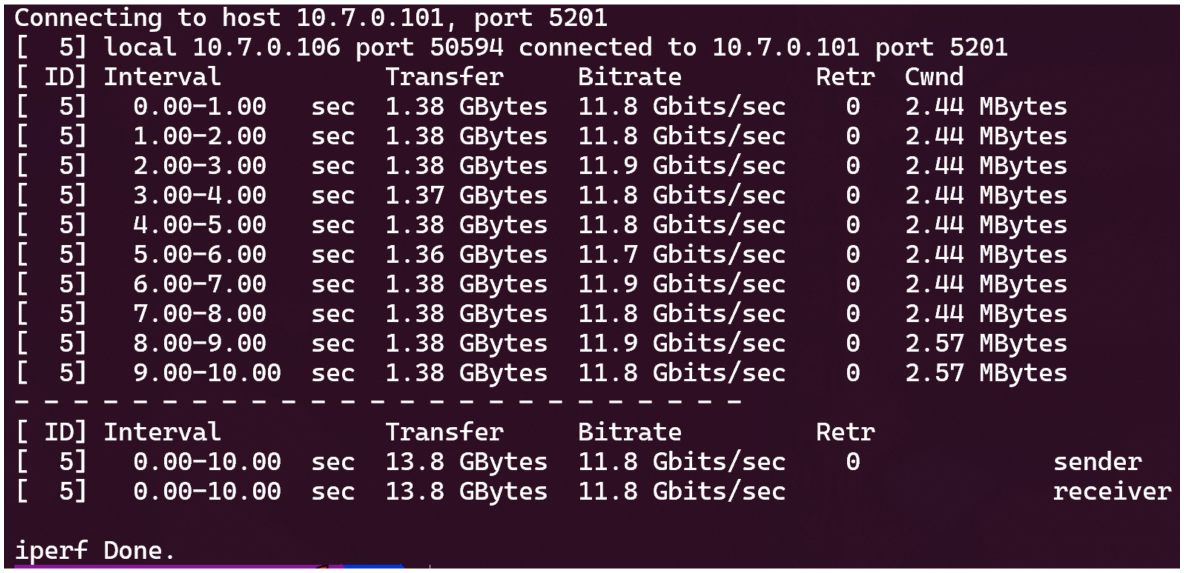
```
{...}: {
  systemd.network.enable = true;
  # To 02
  systemd.network.links."50-tbt-02" = {
    matchConfig = {
      Path = "pci-0000:c7:00.5";
      Driver = "thunderbolt-net";
    };
    linkConfig = {
      MACAddressPolicy = "none";
      Name = "tbt-02";
    };
  };
  systemd.network.networks.tbt-02 = {
    matchConfig = {
      Path = "pci-0000:c7:00.5";
      Driver = "thunderbolt-net";
    };
    addresses = [
      {
        addressConfig = {
          Address = "10.7.0.101/32";
          Peer = "10.7.0.106/32";
        };
      }
    ];
  };
  # To 01
  systemd.network.links."50-tbt-01" = {
    matchConfig = {
      Path = "pci-0000:c7:00.6";
      Driver = "thunderbolt-net";
    };
    linkConfig = {
      MACAddressPolicy = "none";
      Name = "tbt-01";
    };
  };
  systemd.network.networks.tbt-01 = {
    matchConfig = {
      Path = "pci-0000:c7:00.6";
      Driver = "thunderbolt-net";
    };
    addresses = [
      {
        addressConfig = {
          Address = "10.7.0.102/32";
          Peer = "10.7.0.103/32";
        };
```

```
          }
        ];
      };
    }
```

Basically, I use systemd-udevd to configure the Thunderbolt bridge network device and then have another system-network configuration to set an IP and peer IP on the interface.

Benchmark result

Enough of talking. Let's see some benchmark results with iperf3:



```
Connecting to host 10.7.0.101, port 5201
[  5] local 10.7.0.106 port 50594 connected to 10.7.0.101 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  1.38 GBytes  11.8 Gbits/sec    0   2.44 MBytes
[  5]   1.00-2.00   sec  1.38 GBytes  11.8 Gbits/sec    0   2.44 MBytes
[  5]   2.00-3.00   sec  1.38 GBytes  11.9 Gbits/sec    0   2.44 MBytes
[  5]   3.00-4.00   sec  1.37 GBytes  11.8 Gbits/sec    0   2.44 MBytes
[  5]   4.00-5.00   sec  1.38 GBytes  11.8 Gbits/sec    0   2.44 MBytes
[  5]   5.00-6.00   sec  1.36 GBytes  11.7 Gbits/sec    0   2.44 MBytes
[  5]   6.00-7.00   sec  1.38 GBytes  11.9 Gbits/sec    0   2.44 MBytes
[  5]   7.00-8.00   sec  1.38 GBytes  11.8 Gbits/sec    0   2.44 MBytes
[  5]   8.00-9.00   sec  1.38 GBytes  11.9 Gbits/sec    0   2.57 MBytes
[  5]   9.00-10.00  sec  1.38 GBytes  11.8 Gbits/sec    0   2.57 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  13.8 GBytes  11.8 Gbits/sec    0             sender
[  5]   0.00-10.00  sec  13.8 GBytes  11.8 Gbits/sec                  receiver

iperf Done.
```

iperf3 benchmark result shows 11Gbps network speed

As you can see, the network speed reaches 11Gbps!

Afterthoughts

It's fantastic that I can build a network running at 11Gbps at such a low cost. But still, I don't understand why it can only hit 11Gbps at this moment. I saw other people building similar networks were able to hit 20Gbps. As far as I know, USB4 is almost an open-source version of Thunderbolt 3. And it doesn't guarantee the speed to be 40Gbps even if the manufacturer claims it's USB4. So it could simply be that the machine only supports up to this speed.

I also heard other people say that because Intel sells high-speed network controllers, and if USB4 or Thunderbolt 4-based networks can achieve the same level of speed, it might compete with their network controllers, so they capped the speed. If you know why I can only hit 11Gbps instead of 20Gbps or even 40Gbps, please let me know 🙏

Another interesting uncharted area of the idea of a high-speed USB-based network would be how many nodes we can connect and how. With two ports on each machine, I can make a full-mesh network, but what if there are more than three nodes? I recall reading some networking books that mentioned interesting ancient network structures a long time ago, such as ring topology networks or daisy chain networks. There are many drawbacks to those network structures, and network equipment is pretty cheap, so those are rare nowadays. With a limited number of USB4/Thunderbolt ports and relatively expensive cable, maybe it makes sense to construct a network like the ancient ones. What if we can make a high-speed switch built with many USB4 ports and controllers? How much will it cost compared to the ethernet equivalents?

In the near future, when high-speed USB/Thunderbolt controllers and cables become standard and widely available on modern computers, we can see more close-range high-speed networking applications at a very low cost. I may spend some time exploring the idea of building a USB/Thunderbolt-based high-speed network. In the meantime, I would like to know if you have ever built anything cool with these technologies. Please feel free to leave a comment below!
Share this post!

Recent articles:

Why I built a self-serving advertisement solution for myself

Why and how I build and maintain multiple products simultaneously

A repairable future