

microclouds

*Sevki
s@sevki.org*

ABSTRACT

This article is briefly describes how to do cutesy stuff with python over go.

Background

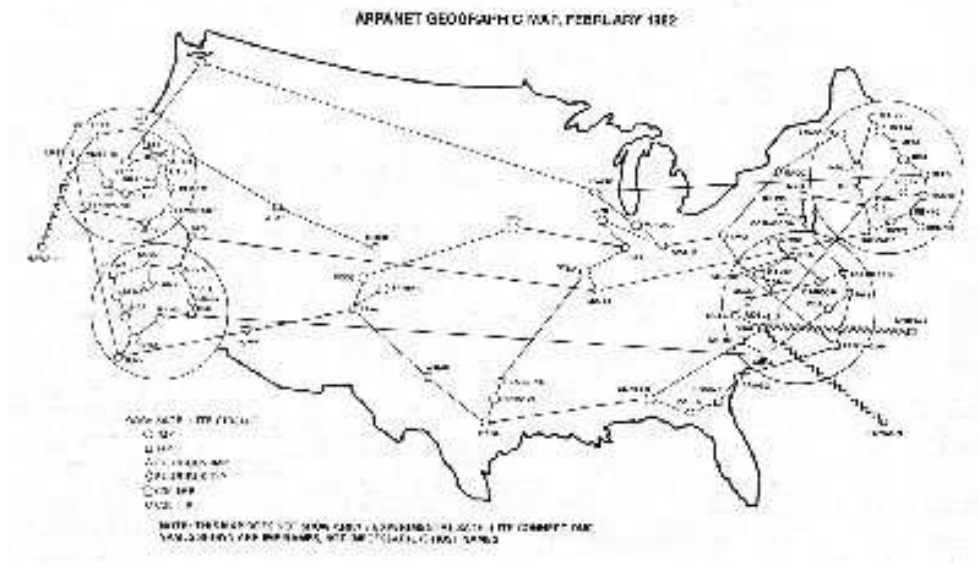
In the 1964, before the internet was the internet we could draw all the nodes on the ARPANET on a geographic map



The ARPANET in December 1969

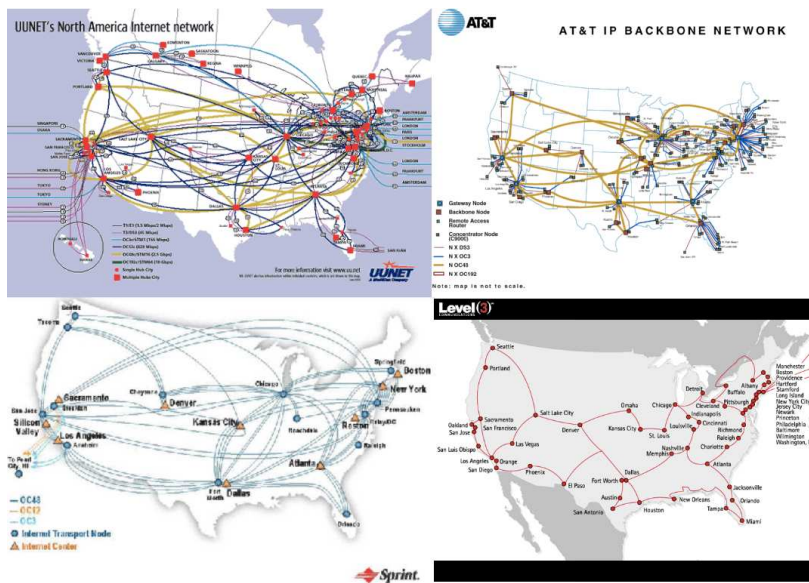
source

by 1982 the network is so crowded the map starts becoming less legible so we started clustering things in to sets



source

After the privatization of the network the internet maps start getting lines with curves.

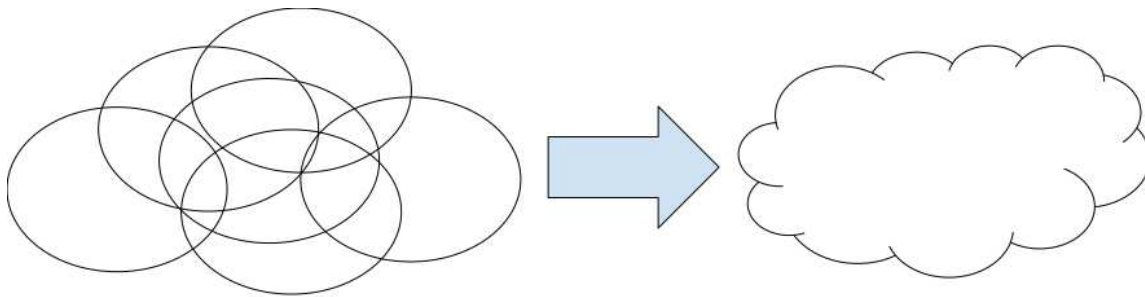


source

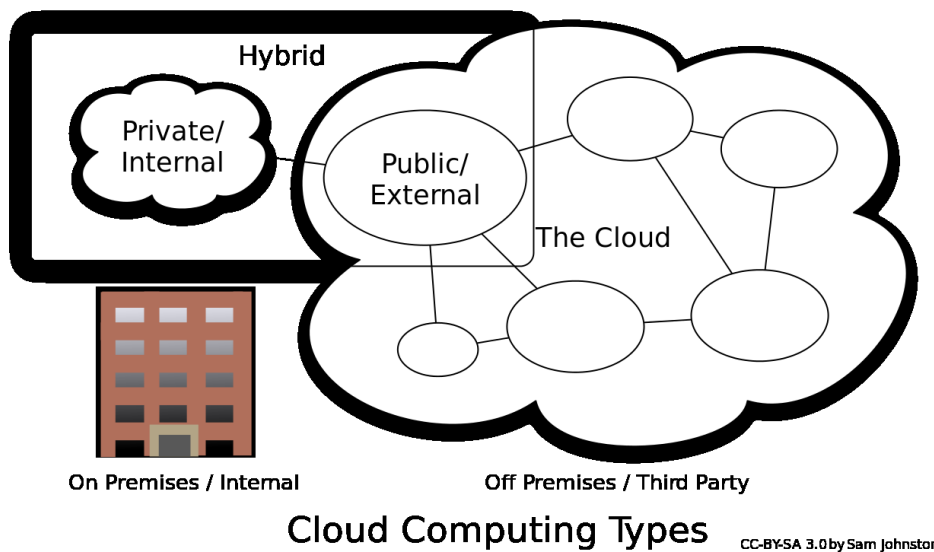
Before the privatization there was a single network there may have been multiple links from one node to the other but since they were all operated by the same entity there really wasn't much point in distinguishing them. After the privatization we started to delineate links with curves so that we can overlay different ISPs on the same map without them bleeding into each other too much.

And that lasted a hot minute before we gave up strated drawing a bunch of eccentric circles to denote the

boundaries of the external system. Which look like clouds, so we gave up all together and started drawing clouds to mean, a system that is too big and too dynamic to be meaningfully mapped.



Today when someone refers to the cloud, they maybe referring to the internet, but more often than not a they are probably talking about cloud computing and/or cloud computing platforms like AWS, GCP or Azure.



Promise of the cloud

The promise of the cloud, or rather cloud computing, is that everything within the boundary of the "cloud" is handled for you.

If a hard drive fails, your cloud provider will replace it. If top of the rack switch fails, your cloud provider will replace it. If the rack is borked, your cloud provider will replace it.

Yes your cloud provider will take care of all these issues for you, with a catch. There will be downtime.

Your cloud provider will migrate you to the working versions of the resources you're paying for, which is great. But they won't be guarantee failover capacity unless you're paying for it.

And that is to be expected, because the cloud is not homogenous.

Your cloud keeps shouting "No Homo"

Microsoft, Google and Amazon are the biggest public cloud providers. In 2016 Gartner estimated Google alone has 2.5 million servers. So it should come as no surprise that a typical Google datacenter employs ~150 people. Those 150~ employees are replacing HDDs, SSDs, Memories, CPUs, switches, racks and stuff as quickly as they are failing.

While non of the big cloud providers speak of these numbers publicly, backblaze does. According to them the drive failure rate is around 0.89%. So let's assume Google, has 2.5 million servers, let's say each server has 24 HDD slots, let's assume all of them are filled with HDDs and not SSDs.

$2,500,000 \times 24 = 60,000,000$ drives in total $60,000,000 \times 0.89 = 53,400,000$ drive failures per year
 $53,400,000 \div 365 = 146,301.369863$

If you haven't worked for a Hyperscaler, you might be thinking no way Google destroys 146,301 Hard Disk Drives per day, and you're right, 2,5million servers were from a 2016 estimate, I suspect it'll be more today.

Say Hi to Denise, her full time job is destroying HDDs.



Also here is a bit more information about Google's scale, https://www.youtube.com/watch?v=P_D2LLgI5uI

According to research is only %9 percent of the market. <https://www.srgresearch.com/articles/quarterly-cloud-spending-blows-past-30b-incremental-growth-continues-riseso> the numbers for AWS and Azure are going to be even higher.

Which is why Google has invested in a robot to destroy hard drives that fail <https://www.datacenterknowledge.com/google-alphabet/robots-now-annihilate-hard-drives-google-data-centers>

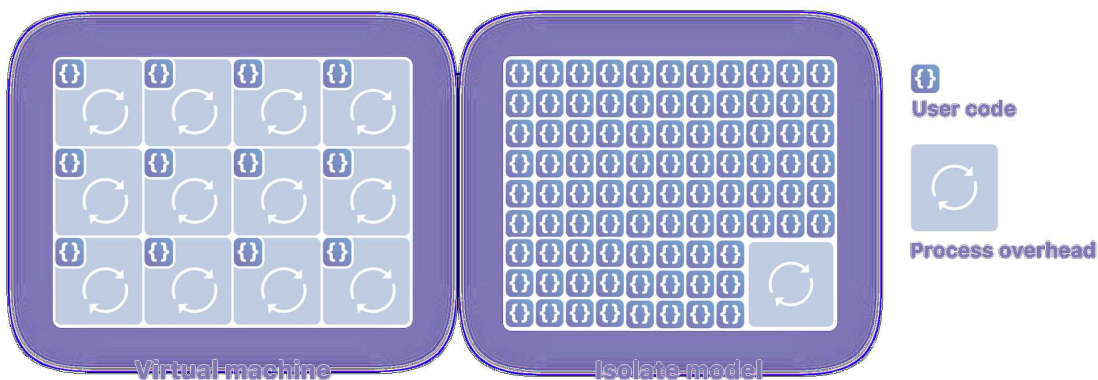
All this to say; at Microsoft, Google and Amazon's scale, procuring the same hardware for millions if not billions of servers at different continents from different vendors is simply impossible.

So the way these cloud platforms standerdiser their platforms is by virtualizing your hardware. If software defines the hardware characteristics it's easier to make every machine have the same ammount of sockets, cores, memory by converging them on the same common denominator.

Multitenancy

Which brings us to multitenancy. Multitenancy is how your cloud provider makes money. You tell them the size of the machine you want, and they allocate it to you.

Almost every talk I gave at Cloudflare with Zach, we used a version of this image, to describe the overhead of virtual machines.



source: Cloud Computing without Containers

While you can see there is a lot of process overhead with packing virtual machines on the the same machine, since all the VMs all have their own network stack, operating system and so on, that is not your cloud providers problem and all of them do the same exact thing so they are all pretty much on the same footing there.

So while your cloud provider will be able to migrate you on the same machine (if your HDD or the CPU that your process is running on fails), things get iffier when you move to another machine.

A bunch oh years ago I saw Mark Russinovich demo live migrating a VM with a running application to another physical machine and the proof of the demo working was monotonic clock would jump as much as the duration it takes to migrate the VM.

Cloud is not magic

Cause of course they can transfer your VM to another DC but it would take a (relatively) long time during which they'd have to copy over the VMD and the entire VM state, which could very easily be TiBs large.