

How to Understand The Rise of Commodity Servers in the Cloud - Brightwork Research & Analysis

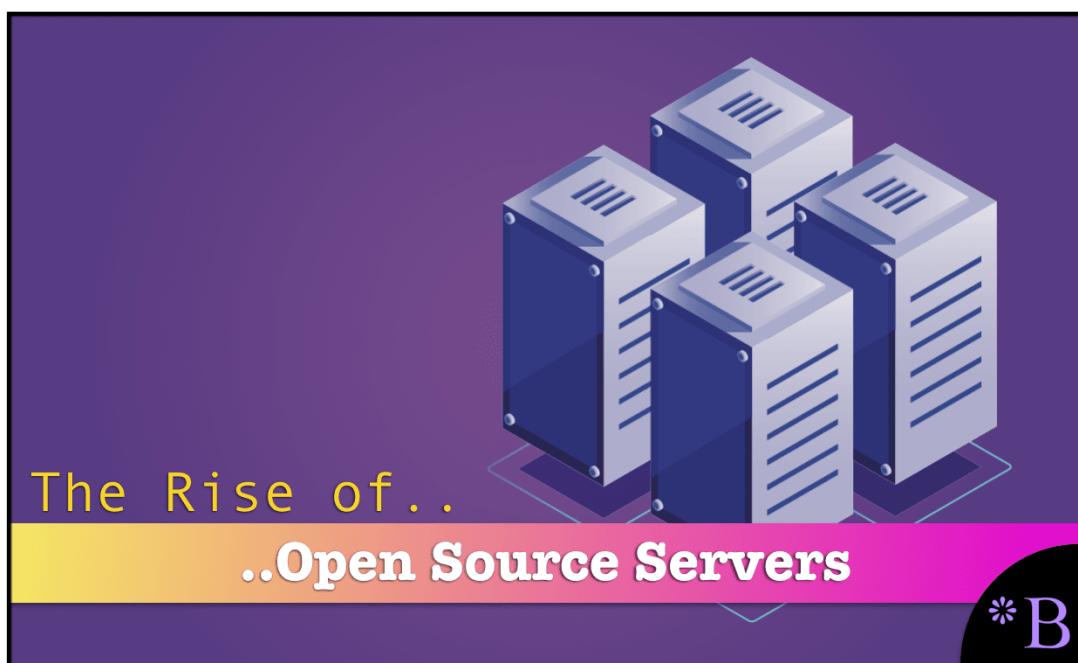
Shaun Snapp

20-25 minutes

Last Updated on April 17, 2021 by

Executive Summary

- Commodity servers are rising in popularity while proprietary servers are declining.
- Cloud service providers use these commodity servers.



Text Introduction (Skip if You Watched the Video)

Commodity servers are rising in popularity while proprietary servers are declining. Cloud service providers like AWS and Google Cloud use these commodity servers. However, for decades, proprietary server manufacturing like HP controlled the information about servers and made it appear that proprietary servers were the only option. A big part of the story around cloud services is the rise of large numbers of commodity servers. You will learn how commodity servers used in cloud services follow these design principles and what it means.

A Big Part of the Story Around Cloud Services

A big part of the story around cloud services is the rise of large numbers of commodity servers. Commodity servers used in cloud services follow these design principles.

“A governing principle of commodity computing is that it is preferable to have more low-performance, low-cost hardware working in parallel (scalar computing) (e.g. AMD x86 CISC[2]) than to have fewer high-performance, high-cost hardware items[3] (e.g. IBM POWER7 or Sun-Oracle’s SPARC[4] RISC). At some point, the number of discrete systems in a cluster will be greater than the mean time between failures (MTBF) for any hardware platform[dubious – discuss], no matter how reliable, so fault tolerance must be built into the controlling software.[5][6] Purchases should be optimized on cost-per-unit-of-performance, not just on absolute performance-per-CPU at any cost.”(1)

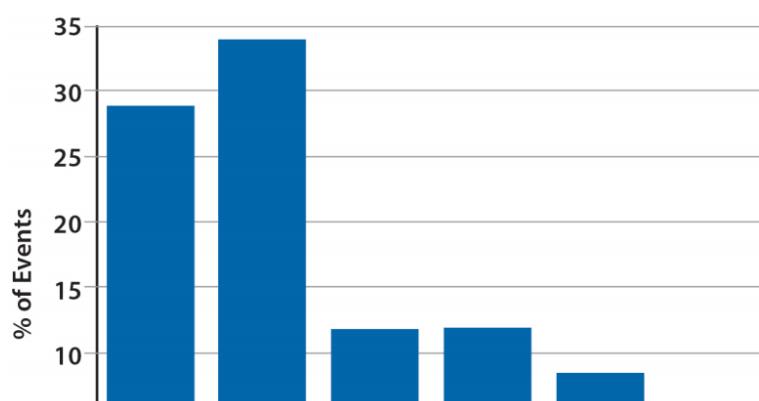




Figure 7.1: Distribution of service disruption events by most likely cause at one of Google's main services, collected over a period of six weeks by Google's Robert Stroud.

The Basis of the Components

These servers are based upon “off the shelf” components, and their lack of proprietary nature means that the hardware vendors do not have the same type of lock as non-commodity vendors (for instance, vendors like HP, IBM, and Dell that a relatively small percentage of their servers to hyperscale cloud providers). It is widely reported that the failure rate of commodity servers is higher. There is considerable redundancy (or extra servers) that allows these servers to be repaired in a “lazy fashion” at the cloud provider’s schedule. Less often observed, hardware failures are only one of the points of failure of data centers.

This is explained in the following quotation.

“In reality, typical servers see an MTBF substantially less than 30 years, and thus the real-life cluster MTBF would be in the range of a few hours between failures. Moreover, large and complex Internet services are often composed of several software modules or layers that are not bug-free and can themselves fail at even higher rates than hardware components. Consequently, WSC applications must work around failed servers in software, either with code in the application itself or via functionality provided via middleware such as a provisioning system for virtual machines that restarts a failed VM on a spare node.”

Notice, even with commodity hardware, the hardware itself is not a primary source of failure.

As explained by the following quotation, commodity servers tended to be purchased in large quantities but by a small number of

customers.

“The only people buying commodity servers these days are cloud providers like Amazon, Microsoft, and Google, and they are more likely to build their own than buy.”

This is a bit misstated. The quotation should have said that these providers **tend to provide their specifications to contract manufacturers**. These cloud service providers do make some components of the servers themselves. Still, most of the server is manufactured in Asia from either their specifications or specifications influenced by them and shared at The Open Compute Project.

A bit more about their general reliability is explained in the following quotation.

“Because Internet service applications run on clusters of thousands of machines—each of them not dramatically more reliable than PC-class hardware—the multiplicative effect of individual failure rates means that some type of fault is expected every few hours or less (more details are provided in Chapter 6). As a result, although it may be reasonable for desktop-class software to assume a fault-free hardware operation for months or years, this is not true for datacenter-level services—Internet services need to work in an environment where faults are part of daily life. Ideally, the cluster-level system software should provide a layer that hides most of that complexity from application-level software, although that goal may be difficult to accomplish for all types of applications.”

If we look at the history of computer hardware, it is clear that there has been a long-term **reduction** in computer hardware’s proprietary nature. Commodity servers can be seen as one side of the spectrum. The server becomes nearly unbranded (aka a white box), with mainframes and appliances, aka “engineered systems,”

being on the other. On-premises proprietary servers would then occupy the middle space.

In the early years of computers, when IBM and the seven dwarfs ruled roost, the computer industry's money was **hardware**, and software was included with hardware purchases. In the 1960s, IBM used to share the open-source code to the operating systems on their mainframes through the early part of the 1960s, but then later stopped as they went to a proprietary model, that is as software became remunerative in its own right. The best way of thinking of this era is how the Japanese firms that work in software operate today and use software to enhance and sell hardware. This position between hardware and software has switched, and the overall legal framework of software has also changed, as explained in the following quotation.

“The basic bargain is that the patent holder publicly discloses the invention and receives in return a temporary exclusive right to make, sell or license others to make or sell the patented good. Is software a patentable invention? Until the early 1980’s, the consensus answer to this question in both the US and Europe was “probably not.” Over the course of the 1980s, however, the U.S. Court of Appeals of the Federal Court gradually removed limitations on the patenting of software in a series of decisions that culminated in the famous 1998 State Street Bank decision, which overturned the prohibition on the patenting of mathematical algorithms and business methods.”

Now hardware is generally low margin (with the notable exception of IBM’s mainframe business (which is highly specialized). And of course, Apple in the consumer market, which is connected to their software, is no broadly Apple clones. However, save for a few exceptions, the software has become the higher-margin item between the two. However, purely commodity computer hardware is the **outlier** in the hardware market, although it is the fastest-

growing computer server market segment.

Cloud service providers like AWS and Google and Azure rely upon commodity hardware. Many things make hyperscale cloud service providers distinct from other companies in the enterprise software space. Given AWS and Google's critical reliance on commodity hardware and Linux, it is a bit surprising that this story is not more widely covered. And understanding this means understanding the history of how we got to where we are now.

The Creation of a New Hardware Modality by Google (and AWS)

Google innovated in data centers' scale by **shunning the use of proprietary hardware and software and building** its infrastructure on open source and commodity or open-source hardware. What Google does with all of this hardware is amazing. (we will focus more on Google here, as Google publishes information about themselves, while AWS prefers to keep things more secretive).

A single Google search, for example, uses a cluster of 10,000 (as per 2009) computers running Linux. However, this is a bit misleading, as the hardware components are a commodity (or should we say designed by the buyer rather than a hardware vendor), but both **AWS and Google custom design their hardware**. Google even produces their server power supplies.

These two companies did not come up with these ideas independently from one another. Google was the initial innovator in this area, and Amazon (before they had expanded into AWS, of course) was so impressed by Google's approach and decided to follow suit.

All of this runs on Linux. Linux, initially the extreme underdog, has worked its way into servers onto mainframes and phones (with

Android, which is currently 84% of the smartphone market). The Apache web server and MySQL run most of the Internet. Linux is itself based upon UNIX, which was created as open-source by Bell Labs, and for a while was essentially open-source mostly because AT&T was prevented from creating commercial products outside of telecommunications by the US Department of Justice. (this position by AT&T was later reversed as conservative forces become more powerful, which meant reducing antitrust enforcement).



The Linux Foundation hosts many of the most important open source projects in the world, including Linux. With more than 1,000 companies backing tens of thousands of active developers, our projects harness the power of open source development to fuel innovation at unmatched speed and scale.



It should be observed that while no one controls Linux, it is managed by several entities that coordinate and communicate for the most important open source project in the world. The Linux Foundation is one of these entities. Linux is the centerpiece, but there are many open-source projects at The Linux Foundation.

A Revolutionary Approach

When Google and AWS were making their decisions, using an open-source operating system was considered innovative and risky. Azure also really only began to take off in growth when it also

opened to using Linux (after many years of distributing fake FUD about Linux).

To fully appreciate how new this approach was, it is necessary to consider the time's environment. This is the description of the status quo of the time is explained thoroughly in the following quotation.

“To those running the commercial Unix cathedrals of Sun, DEC, IBM, and others, the notion of distributing source code to those operating systems, or that enterprise workloads could be handled on commodity hardware, **was unfathomable. (emphasis added)** It simply wasn’t done — until companies like Red Hat and Suse began to flourish. Those upstarts offered the missing ingredient that many customers and vendors required: a commercially supported Linux distribution.

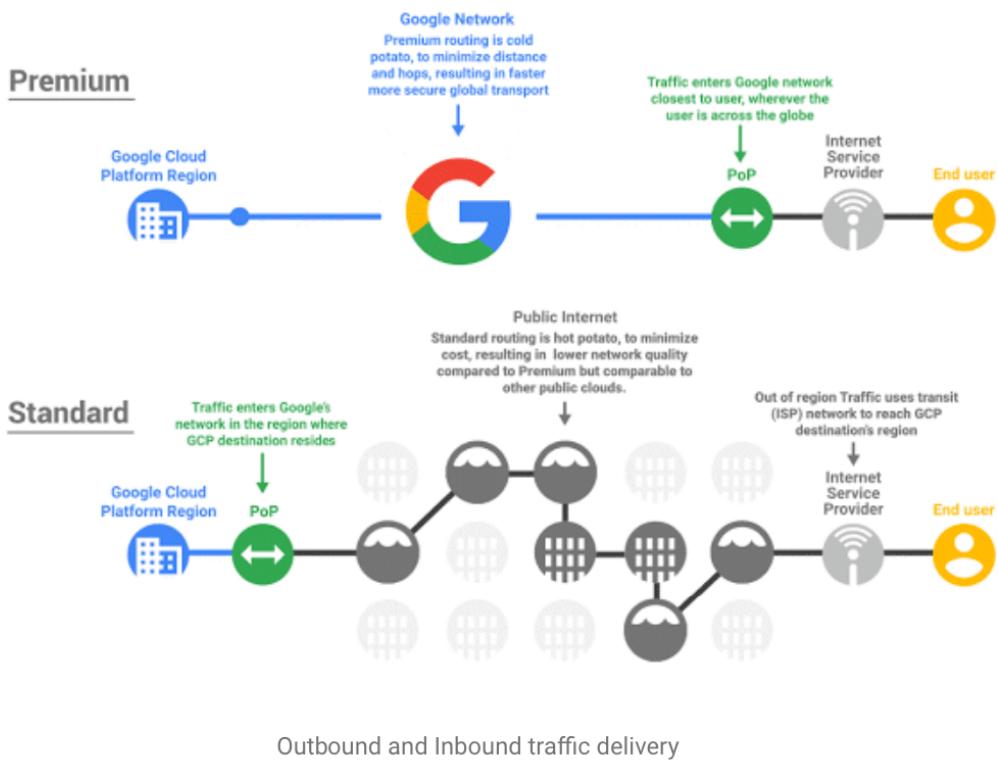
The decision to embrace Linux at the corporate level was made not because it was free, but because it now had a cost and could be purchased for significantly less — and the hardware was significantly cheaper, too. When you tell a large financial institution that it can reduce its server expenses by more than 50 percent while maintaining or exceeding current performance and reliability, you have their full attention.”

Redundant Inexpensive Linux Servers

Google developed the approach of vast numbers of inexpensive redundant Linux-based servers (once estimated in 2016 as around 2.5 million per Google data center, an earlier estimate of 450,000 servers from 2007 shows how the Google data centers have grown.). This decision was made when everyone told Google to purchase hardware from the established proprietary vendors like HP instead.

This is observed in the following quotation from Google.

“Our hardware must be controlled and administered by software that can handle massive scale. Hardware failures are one notable problem that we manage with software. Given the large number of hardware components in a cluster, hardware failures occur quite frequently. In a single cluster in a typical year, thousands of machines fail and thousands of hard disks break; when multiplied by the number of clusters we operate globally, these numbers become somewhat breathtaking. Therefore, we want to abstract such problems away from users, and the teams running our services similarly don’t want to be bothered by hardware failures. Each datacenter campus has teams dedicated to maintaining the hardware and datacenter infrastructure.”



How the data centers are designed is not the only area of innovation on AWS and Google. At a premium tier (top), customers can access Google's network, which is significantly faster than the public Internet. Google has laid its own undersea cables.

Much is made of Google's redundant strategy regarding individual servers. However, Google's data centers are also redundant.

Google has built overcapacity into their network so that other data centers can handle the load if Google were to have another data center taken offline. This also allows Google to “work at their own pace” in replacing servers that fail.

Global Infrastructure



Only a few hyperscale cloud service providers like AWS, GCP, and Azure have this type of build in redundancy.

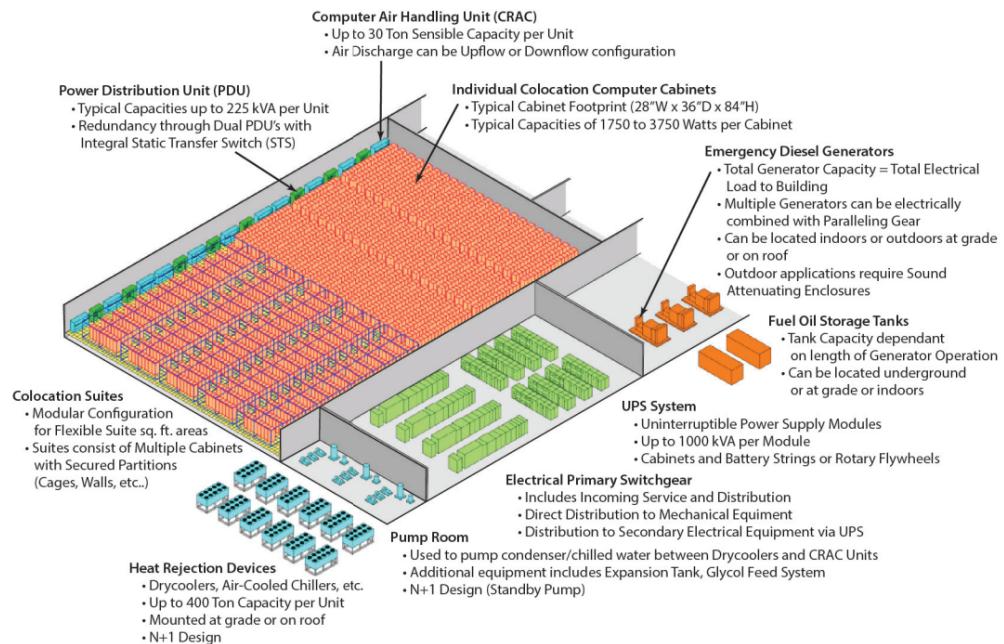


Figure 4.1: The main components of a typical datacenter (image courtesy of DLB Associates [36]).

This is a typical data center design.

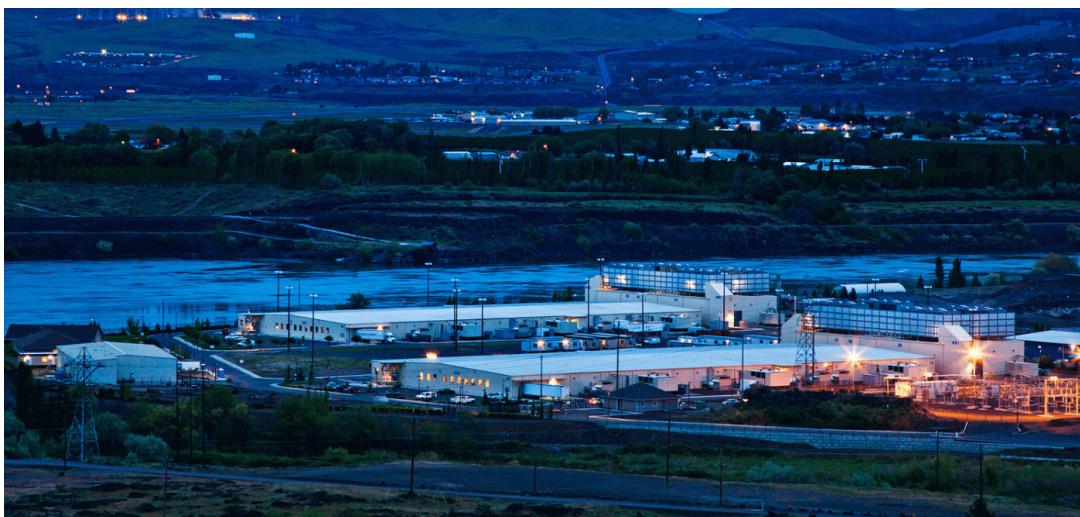
AWS and Google's data centers are not simply large conglomerations of computers. They are now integrated in a way **that they can be thought of like a giant computer**. This is explained in the following quotation.

“Our central point is simple: this computing platform cannot be viewed simply as a miscellaneous collection of co-located machines. Large portions of the hardware and software resources in these datacenters must work in concert to deliver good levels of Internet service performance, something that can only be achieved by a holistic approach to their design and deployment. In other words, we must treat the datacenter itself as one massive computer. The enclosure for this computer bears little resemblance to a pizza box or a refrigerator, the images chosen to describe servers in the past decades. Instead it looks more like a building or warehouse—computer architecture meets traditional (building) architecture. We have therefore named this emerging class of machines warehouse-scale computers (WSCs).”

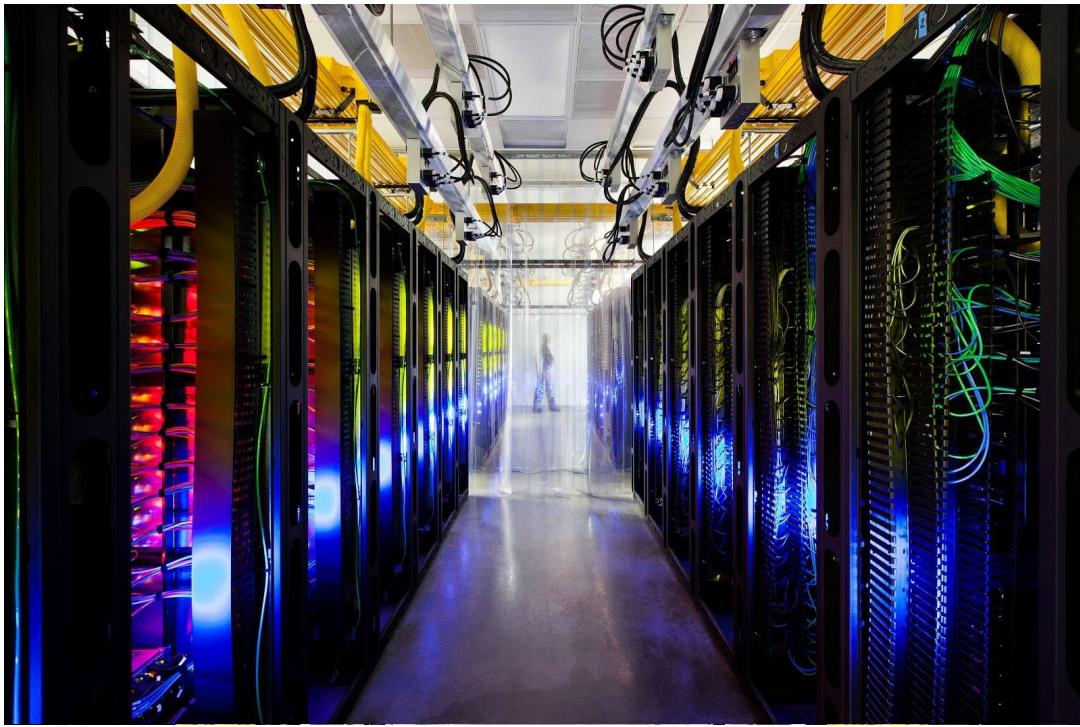
Google's data centers are at the following locations presently (and, of course, may be at many more by the time you read this).

- *“Berkeley County, South Carolina since 2007*
- *Council Bluffs, Iowa announced 2007, the first phase completed 2009, expanded 2013 and 2014, 130 employment positions.*
- *Douglas County, Georgia since 2003, 350 employment positions*
- *Jackson County, Alabama[3]*
- *Lenoir, North Carolina, announced 2007, completed 2009, over 110 employment positions.*
- *Montgomery County, Tennessee announced 2015*
- *Pryor Creek, Oklahoma at MidAmerica announced 2007, expanded 2012, 100 employment positions.*

- *The Dalles, Oregon, since 2006, 80 full-time employment positions*
- *Henderson, Nevada announced in 2018: 1,210 acres of land bought in 2017 in the Tahoe Reno Industrial Center[4]; project approved by the state of Nevada in November 2018[5][3]*
- *Quilicura, Chile announced 2012, online since 2015, up to 20 employment positions expected. A \$140 million investment plan to increase capacity at Quilicura was announced in 2018.[6]*
- *Saint-Ghislain, Belgium announced 2007, completed 2010, 12 employment positions.*
- *Hamina, Finland announced 2009, the first phase completed 2011, expanded 2012, 90 employment positions.*
- *Dublin, Ireland announced 2011, completed 2012, no job information available.*
- *Eemshaven, Netherlands announced 2014, completed 2016, 200 employment positions.*
- *Jurong West, Singapore announced 2011, completed 2013, no job information available.*
- *Changhua County, Taiwan announced 2011, completed 2013, 60 employment positions”[4]*



The Exterior of The Dallas, Oregon Google Data Center location.



The interior of The Dallas location in Oregon.

It is amazing to think of how far things had come since Google's first server in 1998 (when the authors and many others discovered Google).

The specification was the following:

“Sun Microsystems Ultra II with dual 200 MHz processors, and 256 MB of RAM. This was the main machine for the original Backrub system.

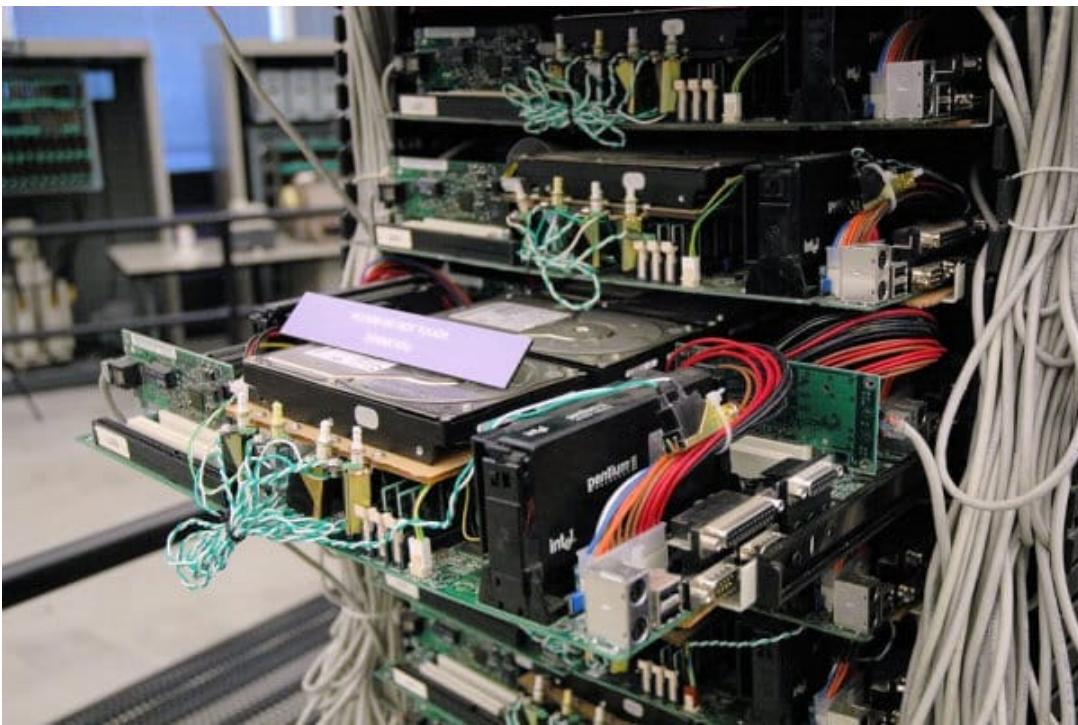
2 × 300 MHz dual Pentium II servers donated by Intel, they included 512 MB of RAM and 10 × 9 GB hard drives between the two. It was on these that the main search ran.

F50 IBM RS/6000 donated by IBM, included 4 processors, 512 MB of memory and 8 × 9 GB hard disk drives.

Two additional boxes included 3 × 9 GB hard drives and 6 × 4 GB hard disk drives respectively (the original storage for Backrub). These were attached to the Sun Ultra II.

SDD disk expansion box with another 8 × 9 GB hard disk drives donated by IBM.

Homemade disk box which contained 10 × 9 GB SCSI hard disk drives.”



One of Google's early “homemade servers.” The phrase “put together with rubber bands and Scotch tape” comes to mind.

The following is an analysis of Google's early homemade server.

“If Google's first production server resembles a hastily cobbled together amalgam of off-the-shelf computer parts circa 1999, well, that's because it is. Just like Google's original servers at Stanford. If you think this rack is scary, you should see what it replaced.

Instead of buying whatever pre-built rack-mount servers Dell, Compaq, and IBM were selling at the time, Google opted to hand-build their server infrastructure themselves. The sagging motherboards and hard drives are literally propped in place on handmade plywood platforms. The power switches are crudely mounted in front, the network cables draped along each side. The poorly routed power connectors snake their way back to generic PC power supplies in the rear.

Some people might look at these early Google servers and see an

amateuristic fire hazard. Not me. I see a prescient understanding of how inexpensive commodity hardware would shape today's internet. I felt right at home when I saw this server; it's exactly what I would have done in the same circumstances. This rack is a perfect example of the commodity x86 market D.I.Y. ethic at work: if you want it done right, and done inexpensively, you build it yourself."

The Number of Servers Used by Google and AWS

This ability is also explained by the number of servers purchased by Google and AWS.

"Once you reach a certain size, says Hamilton, it only makes sense to build your own gear. Buying traditional servers gets too expensive. If you're buying enough hardware to negotiate favorable deals with the likes of Intel – and you've got the means to hire people who can run this kind hardware operation – you can significantly cut costs by going straight to Asia."

"Amazon is running its web operation in much the same way it runs its famous retail business. Ultimately, EC2 is just selling a commodity. So many others can sell the same thing. In order to make it work, you have to operate on very low financial margins. "Amazon, from our retail upbringing, has this background and this comfort and this skill in running high-volume, low-margin businesses," says Andy Jassy, who oversees EC2 and its sister Amazon Web Services. "We have that DNA."

"It will be interesting to see, over the next 10 years or so, how successful the traditional server vendors will be competing against that kind of server capacity," Pinkham says. "Once developers realize they can use this much cheaper, homogenous infrastructure, the power may shift toward the folks who build the cheapest, simplest hardware.""

HPE and Commodity Servers

In 2017, one of the largest hardware providers globally, HP (now HPE), exited selling commodity servers to cloud providers and only will sell higher-end servers. This same strategy of moving upmarket or cutting off the lower market offerings was duplicated by Oracle when they purchased Sun.

The reason for this is simple, HPE cannot make money off of these servers. First, HPE does not itself manufacture the servers. Instead, their value-add is they design them and market them. They are (of course) assembled in China by contract manufacturers like Foxconn, from components that are produced all over the world (though mostly in Asia).

Furthermore, the specifications are open source, which means that neither HPE nor any other hardware vendor owns the IP for the servers' designs.

This is explained in the following quotation.

“Many of those cloud hardware specifications are open source, making it even harder for the likes of HPE and Dell to differentiate from the design manufacturers. Both US vendors and Asian makers — such as Quanta and Wiwynn — are members of the Open Compute Project, the vehicle through which Facebook, Microsoft, and others open source their custom specs and have suppliers compete among each other for large orders of data center hardware built to those specs.”

And this brings us to the Open Compute Project, the open-source specification project we cover in this [article](#).

References

<https://www.morganclaypool.com/doi/pdf/10.2200/S00516ED2V01Y201306CAC024>

“One of the largest Google data centers is located in the town of The Dalles, Oregon, on the Columbia River, approximately 80 miles (129 km) from Portland. Codenamed “Project 02”, the \$600 million[28] complex was built in 2006 and is approximately the size of two American football fields, with cooling towers four stories high.[29] The site was chosen to take advantage of inexpensive hydroelectric power, and to tap into the region’s large surplus of fiber optic cable, a remnant of the dot-com boom. A blueprint of the site appeared in 2008.”

https://en.wikipedia.org/wiki/Google_data_centers

https://en.wikipedia.org/wiki/Google_data_centers