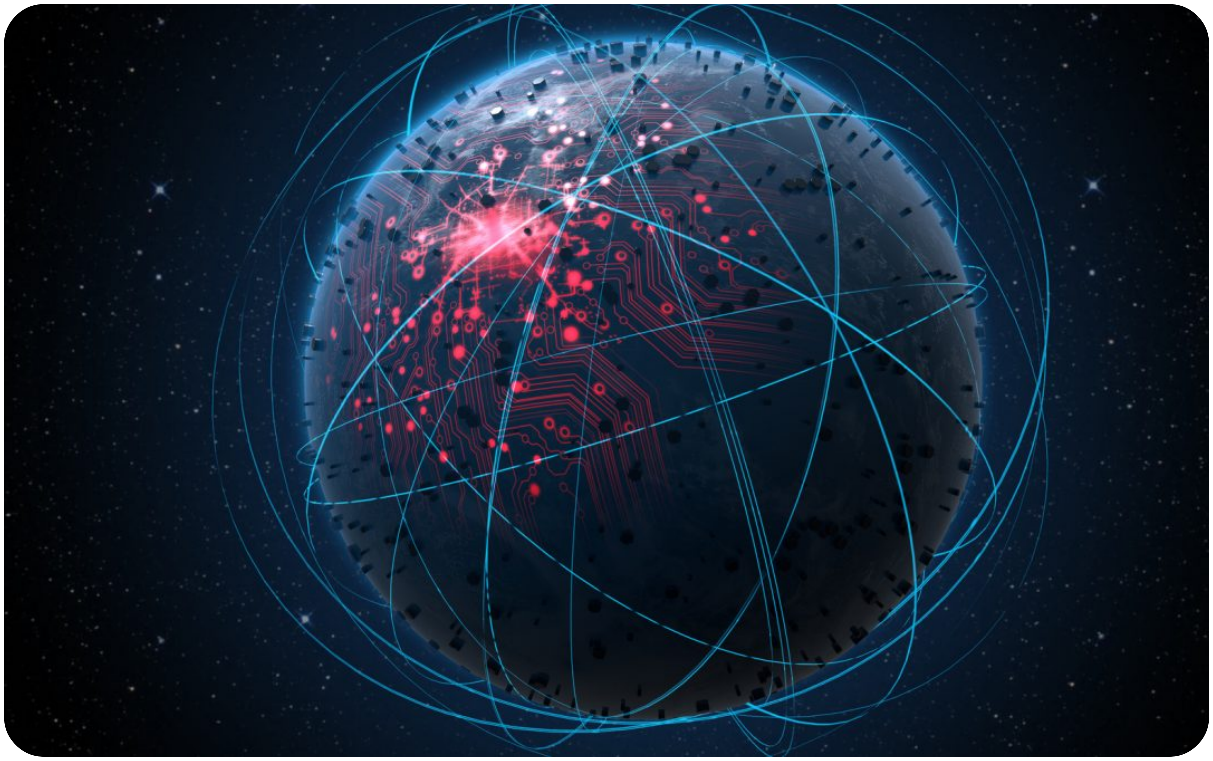
Computer Architecture Today

Informing the broad computing community about current activities, advances and future directions in computer architecture.

Geocomputers and the Commercial Borg

by Michael B. Taylor on Mar 27, 2017 | Tags: Datacenters, IoT, Mobile, Vision



The Desktop is Dead;
Long Live The Geocomputer
(and the Commercial Borg)

Michael B. Taylor, Professor, UC San Diego & University of Washington

Although the desktop is largely an after-thought market of modern computer companies, many computer architects conceptualize modern computers as essentially a miniaturized desktop with some attached I/O widgets. In that computer, we take an ISA-centric definition that takes us back to the halcyon days of the RISC/CISC wars. Servers are x86 and run Linux. Mobile devices are ARM and run iOS/Android. But our emerging applications do not run on a single machine on our desk. A single application executes across many machines of diverse types that are strategically positioned across the planet. It is no longer appropriate to characterize them by what ISA or operating system is used. It is time that we update our mental model of the architecture of the computer that these applications run on today.

Today's emerging modern computers, which I will call **geocomputers,** envelope the planet and have four key components that roughly map to datacenters, the internet, mobile networks, and mobile devices. Each of these has a specific role that connects computation and communication with the environment. But datacenters, mobile devices, etc., are just the physical implementation, like CMOS, or a Manchester adder in traditional machines. The geocomputer is a geocentric architecture that is layered on top of these components, linking *function* and *place* with a given implementation.

The Architecture of the Geocomputer

A geocomputer has four key components: the databrain, the mobot, the planetary network, and the mobot-net. I describe them in greater detail below:

Databrains are the brains of our planet-scale geocentric architecture, and are implemented using datacenters. A databrain is not just a collection of servers mounted in 42U racks with a network, it is a place. Geocomputer architects choose the places to minimize the cost of computation and storage by finding locales that have cheap energy, cooling, and land, as well as geopolitical and environmental stability. A databrain is a deployment of computation and storage proximate to world locations that have high activity (e.g., lots of people generating lots of data), to reduce bandwidth costs and latency.

The databrain's "ISA" replacement is the compatibility layer between the concrete datacenter structural resources (e.g. a composition of flash, disks, cores, networking, etc) and the distributed software that is running on top of it. Its basic primitives are clustered into classes of functionality with names like BigTable, Kubernetes, TensorFlow and FlumeJava. It provides the interface for *storing* data that it is collecting across the world via a planet-scale network. It is *learning* from this data (aka big data), *computing* on this data, and *optimizing* new strategies that are pushed out to its agents that are acting in the world.

Underneath this compatibility layer is a *mesoarchitecture* that transcends the bespoke, customized x86 hardware that is bought in massive quantities from Intel. There is a co-designed software/hardware architecture that determines how data is distributed and made persistent, how computation is distributed across racks of machines, and contains a variety of specialized chips and boards, such as Amazon's network switch chips, or ASIC Cloud chips, that optimize TCO for datacenter primitives running on the geocomputer. For those that cannot afford to build their own databrains, there is Amazon EC2, Microsoft Azure, and Google Cloud, who are willing to rent their databrain if you use their primitives.

Mobots, or mobile robots, are the moving parts of the geocentric computer. They are the I/O network that feeds the databrains, and provides distributed actuation. As they wander the globe, they are constantly *sensing*, *compressing* (to send sensor data to the datacenter over narrow links), and *reacting* (because it would take too long to process what's in the datacenter, or the cost of temporary outages is too high). These are the phones, the self-driving cars, the watches and the robots. (And cyborgs, but we will get to that in a second…) Mobots are very specialized for the application and do not tend to have standardized hardware/software compatibility layers, although mesoarchitecturally, they run things like Android, use ARM cores and have nice displays for doing I/O to other (soon-to-be-mentioned) mobot components.

Planetary Networks are not just the public internet, but many private internets criss-crossing the planet. Planetary networks are there to *span* the globe and *connect* all the nodes, and to *find home (i.e., the databrain)* for the world-wide data the geocentric machine is collecting. These links are placed strategically to minimize the cost of bandwidth, security, and reduce latency to key points of interest in the world.

Mobot-nets are the movable tethers that bridge the fixed places of the internet to the moving locations of the mobots. They are the flexible cord that follows the devices around the world. They are always *moving*, *tracking (e.g. GPS)*, and *evolving (e.g., with more or fewer nodes)*. In some cases, they may be a 3/4/5G cellular network; but they also could be bluetooth, underwater networks, a custom IoT network, or even flexible wired harnesses.

Borgs: Geocomputers that grow and "eat the world"

The Borg is a construct from Star Trek that is essentially a massive computer that is a conglomeration of inter-networked cyborgs. It travels the galaxy, conquering planets and assimilating humanoids and other lifeforms into its organism/machine conglomerate. Its purpose is essentially to conquer the galaxy. The Borg is a militaristic entity; its purpose is to accumulate resources acquired through warfare and enslavement, growing larger and larger as it sweeps the galaxy. As it encounters new targets, it announces to its victims:

"We are the Borg. Lower your shields and surrender your ships. We will add your biological and technological distinctiveness to our own. Your culture will adapt to service us. Resistance is futile."

Unlike in the Star Trek universe, there is not one huge borg in our world, rather, there are many. They conquer not through warfare but rather by commerce. There are big mature borgs, and there are small baby borgs. Many of them were incubated in the SOMA region of San Francisco. Who controls these *commercial borgs*, as opposed to the military borgs forecasted in films? Uber, for one. City by city, the Uber borg advances, gaining control not by violence but by economics. It assimilates new drivers into its fold and controls them through a planetary network, defeating the local Taxi industry neighborhood-by-neighborhood. Many of the assimilated are taxi drivers from the conquered companies but there are also other bystanders who are pulled into the machine.

The ideal mobot for Uber would be a self-driving car that would drive around 24/7, do exactly as commanded, and never listen to rap. But this purely electromechanical mobot is technically infeasible today. Instead, Uber employs a cyborg mobot: a cell phone controls a human, who in turn, with their advanced neural net capabilities, is able to transform the cell phone's control commands, plus sensor data from their eyeballs and ears, into control of the car and the passenger. (Uber has raided the CMU Robotics Center to build the research pipeline to eliminate the problematic "biological distinctiveness" of this mobot.) Uber's unique customization mostly extends to its mobot, using more conventional 3/4G for its mobot-net, the Internet for its planetary network, but its own custom datacenters for its databrain. But we can imagine one day, that Uber will launch its own mobot-net for tracking its mobots more accurately than GPS allows.

But let's not dwell too much on Uber. There's also Yelp, which, having not yet perfected a inorganic mobot that can taste food and take in restaurant ambiance, controls diners via an app, having them translate the sensor data from their tongues into prose. Compression ratios in today's Yelp mobots are extremely high, compressing 4K video from diner's eyeballs, as well as multivariate sense and smell data from their mouths and noses, to a short, clever ASCII paragraph that is uploaded to the Yelp databrain. Better "taste-to-text" hardware is surely on the roadmap for the Yelp research team.

It's the System, Stupid

What is responsible for the transformation of today's computers from the simple desktop to the planet-centric Geocomputer architecture? Marc Andreessen, venture capitalist, and founder of Netscape, wrote an op-ed called "Why Software is Eating the World" in 2011, suggesting that the next wave of tech applications will be those that escape the bounds of our desktop machines and instead engage the world. From this mandate comes our scale-out datacenters, and the vast network of phones feeding them data.

Although we often think of these applications as simply software apps we install on our phones, in fullness, these are special-purpose application-specific geocomputers that are being deployed. As technology progresses, it is hardware that will provide the underpinnings of future geocomputer advances. For as surely as software is eating the world, it is hardware that is doing the chewing. Moving into the future, computer architects must expand their concept of the machine beyond the traditional desktop ISA to the entire geocomputer architecture. Only by being fully engaged with the entire system can we discover hardware's particular special value in unlocking the end-to-end geocomputer application.

Thanks to Alvin Lebeck, Heiner Litz, and Reese Nguyen for providing feedback on this blog post.

Share this:

Facebook 22 | Twitter | LinkedIn

0 Comments ACM SIGARCH 1 Login

♡ Recommend Share Sort by Best

Start the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Be the first to comment.

ALSO ON ACM SIGARCH

SIMD Instructions Considered Harmful

6 comments • a month ago

ProfHankD — Clark may well have introduced some subword operations, and some were heavily used in SIMD

The Unreasonable Ineffectiveness of Machine Learning in Computer

9 comments • 7 months ago

Daniel Ángel Jiménez — Keshav, thanks for a great blog post! I'm actually teaching a class on this very topic:

When to Prototype? When to Simulate?

3 comments • a month ago

David PATTERSON — I'd be amazed if any computer architecture researcher thinks every idea should be evaluated

Back to the future for commodity processor design? Biased

2 comments • 6 months ago

John Alsop — There's still a strong argument to be made for shared memory in many application domains.

✉ Subscribe Add Disqus to your siteAdd DisqusAdd 🔒 Privacy

EDITOR: ALVIN R. LEBECK

Contribute to Computer Architecture Today

RECENT BLOG POSTS

Offensive Security Research in Computer Architecture Conferences

The Road to Success In Industry

Gender Diversity in Computer Architecture
Remembering Nathan Binkert
When to Prototype? When to Simulate?

ARCHIVES

October 2017 (2)
September 2017 (6)
August 2017 (2)
July 2017 (4)
June 2017 (6)
May 2017 (6)
April 2017 (4)
March 2017 (2)

TAGS

Accelerators Advice Architecture Cache Conference CPU Databases Datacenters Debugging
Diversity Emerging Technology Hardware Industry IoT ISA Machine Learning Memory
Methodology Mobile Modelling Near Data Computing Networking Operating Systems Opinion
Parallelism Performance Persistent Policy Programmability Programming Prototyping
Security Shared-Nothing Simulation Smartphone Specialization Storage Systems
Virtualization Virtual Memory Vision

SUBSCRIBE

RSS SUBSCRIBE

Get E-Mail Alerts, enter your Email:

SUBSCRIBE »

ACM SIGARCH

SIGARCH serves a unique community of computer professionals working on the forefront of computer design in both industry and academia. It is ACM's primary forum to interchange ideas about tomorrow's hardware and its interactions with software.

Join Us

Keep up-to-date with the latest technical developments, network with colleagues outside your workplace and get cutting-edge information, focused resources and unparalleled forums for discussions. Learn more...

About the Site

This site is maintained by volunteers working in many programs of ACM SIGARCH. We thank you for visiting! If you have questions about the site, please send a note to our information director.