Why You Need To Know About 'Computications'

Yaakov Stein
Forbes Councils Member
Forbes Technology Council
May 24, 2022,09:45am EDT

Yaakov (J) Stein is CTO at Allot.

What do you do with your computer? With my computer, I solve systems of equations, find optimal values of functions under constraints, separate intertwined signals, hunt for correlations between variables and approximate empirical data. In short, I use my computer for *computing*.

I bet that's a bit different from what most people do. Most people use their computers for video teleconferencing, sending and receiving email and streaming movies — in other words, for *communications*. What do most people do with their smartphones? At times, they use them for voice calls and an occasional SMS, but they run apps on them. The *communications* device par excellence is mostly used for *computation*.

While there was once a clear distinction between communications devices (such as telephones) and computation devices (such as desktop computers), this distinction has become completely blurred. We now universally use combined "computications" devices.

Take the internet of things (IoT), for example, which stresses the connectivity aspects, but IoT devices are also computers — in short, miniature computicators so integrated into our world that they are basically invisible.

The Road To Computications

This marriage of communications and computation was prefaced by a long engagement.

Remember fax machines? They were communications devices that ran sophisticated signal processing and image compression algorithms. The telephone network's switches run algorithms to convert tones into phone numbers. 3G phones could already run limited applications. However, it was the smartphone that obliterated the border between communications and computation; modern smartphones boast processing power that puts to shame the high-end computers of just a few years ago.

Strangely, this unification transpired for end-user devices well before it occurred in the heart of the communications network. While end-users were enjoying computications devices, the communications infrastructure was built of dedicated equipment such as hardware switches and routers. Of course, these boxes were chock full of processors running digital algorithms. These algorithms, though, were embedded into proprietary hardware and developed using different languages and tools than state-of-the-art software.

Computications In The Network

This dissimilarity between end-user computications and infrastructure communications led to a fundamental disconnect between their development timescales. While a new app can be developed in days, a new networking protocol could take months to years to standardize, develop and deploy. This situation soon became unsustainable.

It was not initially clear how communications and computation in the network itself could be unified. The solution came from the world of computation — *virtualization*. For years, programmers had been using virtual machines (VMs), software simulating physical computers. This concept was extended to the virtual network function (VNF), a piece of software emulating a physical network element such as a switch or router.

This idea of network functions virtualization (NFV) had everything going for it. It enabled new pure-software players to enter the telco fields, and it promised to save the business models of communications service providers (CSPs) who were suffering from over-the-top providers eating their breakfast. But NFV never really took off, perhaps because the available VNFs weren't sufficiently attractive financially (partly because initial VNFs were developed by the incumbent hardware vendors who may have priced VNFs so as to avoid cannibalization).

And then things became cloudy.

The situation changed when NFV started availing itself of cloud technologies. The pre-NFV model had proprietary embedded software tightly coupled to a proprietary hardware platform. The initial NFV model dictated generic software running on a commercial off-the-shelf server. The new model espoused generic software running in a cloud.

On paper, this should have elicited a small technological improvement — replacing memory-demanding VMs with lightweight containers and telco-style system management with cloud-native microservice orchestration — but the stars magically aligned, and NFV finally got the needed boost. Open-source communities sprung up, developing alternatives to network boxes and sidestepping standards development organizations. Startups materialized, offering pure software solutions for almost every network function. The hyperscalers (AWS and Microsoft) started addressing the NFV market, too.

And then came edge clouds. The major technological drawback to cloud-based solutions was the unavoidable latency involved in sending information to a distant data center. This was acceptable for management traffic and some control signaling but objectionable for time-sensitive user information. But with the decomposition of monolithic data centers into smaller, more distributed edge data centers, the final obstacle was removed.

5G inherits it all.

The major beneficiary of all these developments is 5G. No, not the present generation of 5G, which is actually just 4G with new spectral allocations and doesn't really merit the moniker. Full 5G will enable a plethora of new applications (such as smart-x, where x=home, utility, city, factory, etc.), and much of this is due to cloud-based computications.

The new 5G core network is designed around the cloud model, and the cellular base station itself has been decomposed to maximize the use of software. Basically, 5G only needs an array of antennas and some front-end circuitry to convert the radio frequencies into digital format, and after that, everything can be processed by various pieces of software, each of which can be run wherever there are resources to do so — from dedicated processors close to the cell site to edge clouds to centralized data centers.

In fact, 5G conveys computications to its logical culmination. Everything around us will be a communicating computer, and all will be connected by a ubiquitous communications infrastructure implemented on communicating computers.

Why bother coining a new word?

Who cares about this culmination of the trend of unifying communications and computation?

Anyone who provides or consumes digital services. When software functionality and communications go hand-in-hand, the time-to-market for new services is drastically reduced, the development and operating costs are slashed and the quality of experience significantly increases. These effects are even more marked for new *types* of services.

What I find amazing is how few people have even noticed this revolution and how most of you have survived until now without the word "computications."

Yaakov Stein
Yaakov (J) Stein is CTO at Allot. Read Yaakov Stein's full executive profile here.