COLUMN: Focus on Community

The Fogging of the Cloud

Christine Miyachi
Xerox Corporation

Fog and edge computing decentralize cloud computing but often depend on centralized cloud servers. There is an active consortium supporting architecture and standards as fog computing complements cloud computing.

A few jobs ago, I worked as a control systems engineer. The machines I worked with moved and then oscillated. My job was to remove the oscillations as quickly as possible, so I designed software to move a machine and minimize oscillations, using a proportional, integral, and derivative (PID) controller.[1] Figure 1 shows a typical oscillation.

Working on control systems, you learn that everything oscillates. One oscillation in the computing world has been the swing from centralized to decentralized. Before the word “cloud” was used, we had mainframes (centralized), but then PCs became popular (decentralized). We have the Internet (decentralized), but we have large governing bodies like the Internet Corporation for Assigned Names and Numbers (ICANN) controlling domain names among other things (centralized). We have blockchain-based contracts and cryptocurrencies (decentralized), as well as governments and banks that want to regulate them (centralized). We have proprietary software like UNIX (centralized) and open-source software like Linux (decentralized). We have cloud servers that are centralized, and now we have edge and fog computing, which decentralize the cloud. Will the oscillations ever end? Fog computing helps to solve the problem of getting data where it is needed quickly and reliably, although there are still tradeoffs.[2]

Edge computing is not new. Back in 2000 when the recession hit and I was in graduate school, there was a local company that seemed to be recession-proof: Akamai.[3] The company originated from a challenge by Tim Berners-Lee, the inventor of the World Wide Web, who asked his colleagues at MIT to create a faster, better way to deliver newly rich complex webpages. In 1998, Akamai was incorporated and launched products to help companies deliver content robustly, in part by locating servers near where the actual web content loads into a browser. The content was mirrored or duplicated at various physical locations around the world to aid in quicker delivery. Today, the website says concisely, “Akamai is the edge.” The company founders may not have foreseen the explosion of IoT (Internet of Things), but today Akamai offers a wide variety of edge-computing products.[4]

The structure of cloud-computing hardware physically near where it is needed has been around for some time. Decentralizing computing and storage made sense in the mid-1990s for the same reason it makes sense today. Doing the computation where you need it can be more secure, faster, and more efficient. And yet centralizing some operations in cloud servers is still a valuable activity. Will the oscillations ever stop?

Copublished by the IEEE CS and IEEE ComSoc
2325-6095/18/$33.00 USD ©2018 IEEE

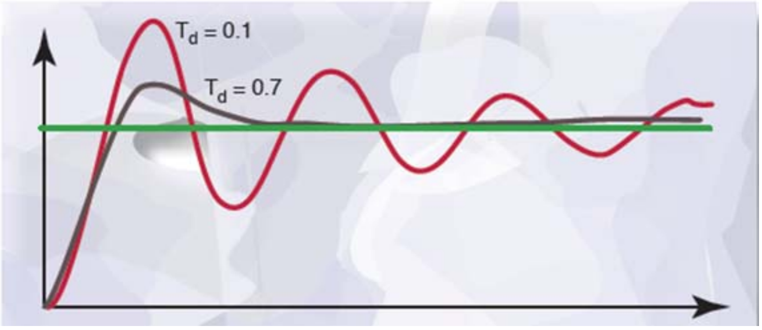


Figure 1. Oscillations of a PID controller.[5]

DEFINITION

Fog computing is defined as "an architecture that uses edge devices to carry out a substantial amount of computation, storage, and communication" locally, as well as routed over the Internet backbone. It "most definitively has input and output from the physical world, known as transduction" (defined as the process of directly drawing conclusions about new data from previous data, without constructing a model).[6]

Edge computing is defined as "a method of optimizing applications or cloud-computing systems by taking some portion of an application, its data, or services away from one or more central nodes (the 'core') to the other logical extreme (the 'edge') of the Internet, which makes contact with the physical world or end users."[7]

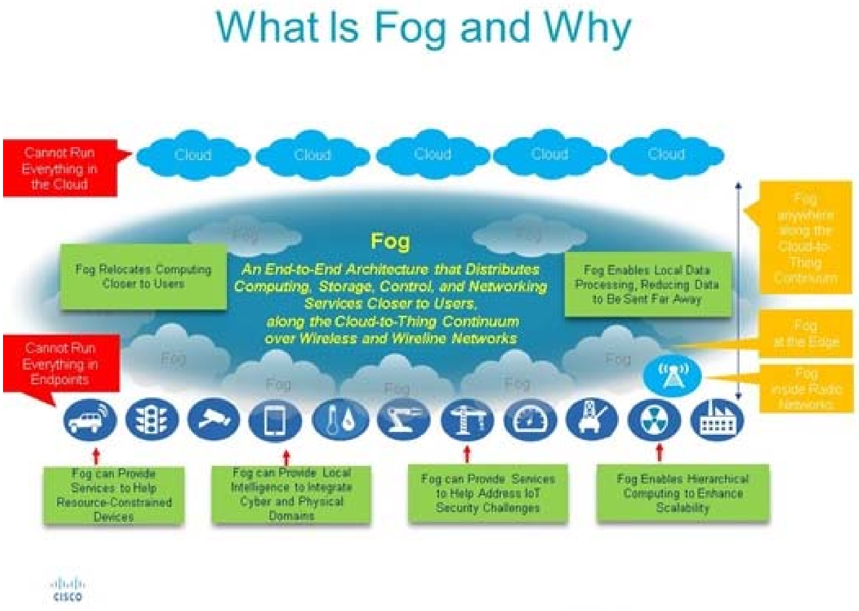
Both fog and edge have the concept of providing services closer to where a thing or user needs it. David King, CEO of FogHorn Systems says, "Many in industry indeed use the terms fog computing and edge computing (or edge processing) interchangeably. Edge computing is an older expression that predates the fog-computing term. By way of background, Cisco created the term fog computing years ago to describe a layer of computing at the edge of the network that could allow pre-processed data to be quickly and securely transported to the cloud."[8]

The National Institute of Standards and Technology (NIST) says there is a difference: "Fog computing is often erroneously called edge computing, but there are key differences. Fog works with the cloud, whereas edge is defined by the exclusion of cloud and fog. Fog is hierarchical, where edge tends to be limited to a small number of peripheral layers. Moreover, in addition to computation, fog also addresses networking, storage, control, and data-processing acceleration."[9]

To understand this difference, see the diagram presented by Tao Zhang from Cisco Systems to the IEEE Cloud Computing group in 2016 (see Figure 2). His diagram shows that solutions use both centralized and decentralized services.

Dr. Zhang points out in this figure that not everything can run in the cloud (because it wouldn't get delivered in time, and so on) and not everything can be run at the endpoints (the full set of resources such as storage and computing power do not exist). The cloud, the edge, and the "things" need to work together to function. Zhang gave the example of the self-driving car, which needs to process a tremendous amount of data in real time to fulfill its purpose. The car won't have all the resources on board to do all the computing, storage, and networking it needs, but it can do some of it. It will rely on fog computing for some functions and the traditional cloud systems for the rest. The car will need all that data analyzed in real time to drive.

When considering IoT, cloud storage is required for these devices to transfer their data aggregate. Device intelligence and increasing capabilities now allow for some processing to take place at the IoT device. There is a large amount of data generated by IoT, and it can be too large to transfer over networks reliably and efficiently. There is consumption of resources that could be avoided (like network bandwidth and storage). The Moore-Nielsen Paradigm[10] states that data that gets accumulated at the edge is faster than the network can push into the core. An example is a jet engine that produces 20 terabytes of data for each flight hour. Additionally, storage and processing power are becoming very cheap, so devices can now have storage and processing power built in. An added benefit is that these devices can do some filtering to reduce the amount of data before being transferred. Finally, if the data is sensitive, it is better to keep that data locally than to send it over the network. For example, medical records being analyzed locally rather than transferred over a network are less likely to be stolen on a network.[11]



What Is Fog and Why

Cannot Run Everything in the Cloud

Cloud Cloud Cloud Cloud Cloud

Fog

An End-to-End Architecture that Distributes Computing, Storage, Control, and Networking Services Closer to Users, along the Cloud-to-Thing Continuum over Wireless and Wireline Networks

Fog Relocates Computing Closer to Users

Fog Enables Local Data Processing, Reducing Data to Be Sent Far Away

Fog anywhere along the Cloud-to-Thing Continuum

Fog at the Edge

Fog inside Radio Networks

Cannot Run Everything in Endpoints

Fog can Provide Services to Help Resource-Constrained Devices

Fog can Provide Local Intelligence to Integrate Cyber and Physical Domains

Fog can Provide Services to Help Address IoT Security Challenges

Fog Enables Hierarchical Computing to Enhance Scalability

Figure 2. Fog computing presented by Tao Zhang from Cisco Systems—the company that coined the phrase “fog computing.”

The second definition of fog computing is “a scenario in which a huge number of heterogeneous (wireless and sometimes autonomous), ubiquitous, and decentralized devices communicate and potentially cooperate among themselves and with the network to perform storage and processing tasks without the intervention of third parties.”[12]

Finally, the OpenFog Consortium (OFC) has the simplest definition: “a horizontal, system-level architecture that distributes computing, storage, control, and networking functions closer to the users along a cloud-to-thing continuum.”[13]

A CONSORTIUM IS STEPPING UP

With all this cooperation and computing occurring at the edge, standards are going to drive innovation. One consortium that has gained momentum in the past two years is the OFC. Members define their work as “centered around creating a framework for efficient and reliable networks and intelligent endpoints combined with identifiable, secure, and privacy-friendly information flows between clouds, endpoints, and services based on open standard technologies” (www.openfogconsortium.org/).

One of the OFC’s biggest efforts so far is to produce an open architecture platform for fog computing. The goals of the architecture can be summed up with the acronym SCALE:[13]

- Security: additional security to ensure safe, trusted transactions
- Cognition: awareness of client-centric objectives to enable autonomy
- Agility: rapid innovation and affordable scaling under a common infrastructure
- Latency: real-time processing and cyber-physical system control

- Efficiency: dynamic pooling of unused local resources from participating user devices

The principles of the OpenFog Architecture build upon the pillars shown in Figure 3.

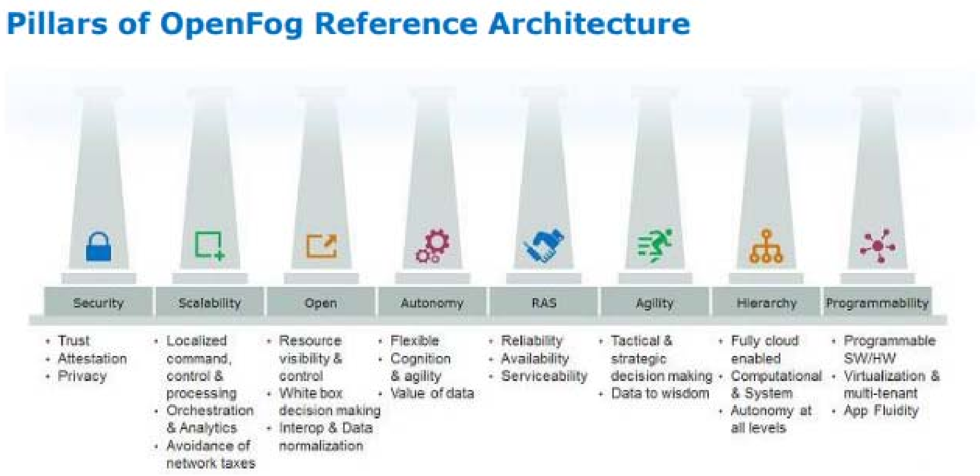


Figure 3. Pillars of the OpenFog Architecture.

The architecture has a goal of creating standards so that many different applications can work together to solve a problem. The OFC gives a compelling use case of a traveler who does the following:

- Leaves from home and drives to the airport
- Parks in the long-term parking garage
- Takes bags to the airport security checkpoint
- Drops off bag at counter
- Checks in through security and proceeds to boarding gate
- Upon arrival, retrieves bags
- Proceeds to a rental-car agency and leaves the airport

All kinds of mishaps can happen:

- The vehicle entering the airport is stolen.
- The passenger's name is on a no-fly list.
- The passenger leaves his luggage unattended someplace in the airport.
- The passenger's luggage doesn't arrive with the flight.
- The luggage is scanned and loaded on the plane, but it is not picked up by the correct passenger.
- An imposter steals or switches a boarding pass with another passenger and gets on someone else's flight.
- The passenger takes someone else's luggage at the arrival terminal.

Developing a solution woven together by various applications with data that can be analyzed immediately is the purpose of the OpenFog Architecture. The use case states: "Catching these possible threats requires an extensive network of surveillance cameras across the outbound and inbound airports, involving several thousand cameras. Approximately 1 terabyte of data per camera per day must be transmitted to security personnel or forwarded to local machines for scanning and analysis." The use case points out that law enforcement will need data from multiple systems about the passenger's trip, and the video and data must be integrated with a real-time threat assessment and remediation system.

In an edge-computing system, video and traveler data is transmitted back to the cloud, making the ability to assess a problem in real time more difficult. By contrast, in a fog-computing system, the data would be collected and analyzed where it was collected, and the cost of data transfer would be reduced. Only analyzed data would be transmitted, with sensitive data remaining in place.

STANDARDS

The IEEE and the OFC are working together to create a new standard: the IEEE Std. 1934-2018 (IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing).[14] The IEEE Standards Association has adopted the OFC's reference architecture as a standard for fog computing and will release more details at the Fog World Congress conference, which takes place October 30 through November 1 in Santa Clara, California. The IEEE Communications Society and the OFC are jointly hosting this first multi-day conference on fog computing and networking.

Furthermore, NIST—a US government agency—has developed "The Fog Computing Conceptual Model," which lays out a set of definitions.[9] NIST adds a new term called "mist computing," which is shown in Figure 4.

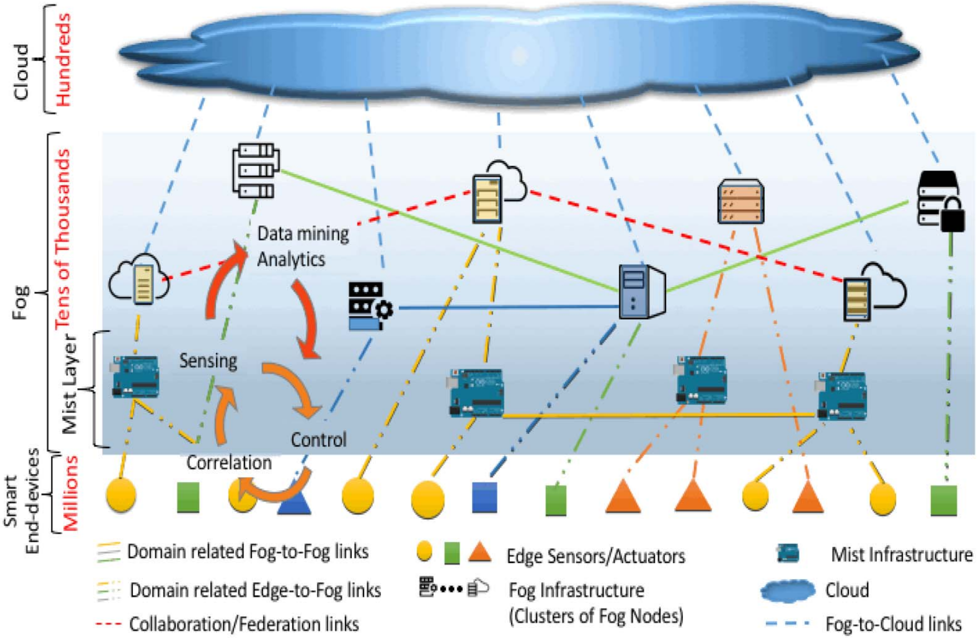


Figure 4. The NIST definition of fog computing and mist computing.

NIST defines mist computing as "a lightweight and rudimentary form of computing power that resides directly within the network fabric at the edge of the network fabric, the fog layer closest to the smart end-devices, using microcomputers and microcontrollers to feed into fog-computing nodes and potentially onward towards the cloud-computing services." NIST clarifies that "network fabric" is defined as a network topology where components pass data to each other through interconnecting switches.

CONCLUSION

After working with centralized cloud-computing systems, engineers found the need and the technology to complement it and decentralize with fog computing. The structure of fog computing requires agreed-upon standards to move forward. The OFC is a community of participating companies that is quickly developing standards to develop this technology. The consortium's website provides some white and technical papers that provide an invaluable resource for understanding and building fog-enabled applications. Like many consortiums related to the cloud, its funds come from member companies, and only individuals from those companies can contribute. Perhaps we can create a community in which individuals can also contribute and comment on standards without being part of a member company, further decentralizing. Fog computing aligns with the principles of a decentralized computing model that originally inspired the Internet. While cloud computing is centralized, fog will enable a paradigm shift and will only be possible with community involvement. Fog computing may be the perfect balance of centralized and decentralized concepts, and a stable PID controller for the computing world may be found.

REFERENCES

1. "PID Controller," *Wikipedia*, Wikimedia Foundation, 2018; https://en.wikipedia.org/wiki/PID_controller.
2. J. Weinman, "The 10 Laws of Fogonomics," *IEEE Cloud Computing*, vol. 4, no. 6, 2017, pp. 8–14.
3. "Company History: How Akamai got its start," *Akamai.com*; www.akamai.com/us/en/about/company-history.jsp.
4. "IoT Edge Connect," *Akamai.com*; www.akamai.com/us/en/products/web-performance/iot-edge-connect.jsp.
5. P. Avery, "Introduction to PID Control," *Machine Design*, 2009; www.machinedesign.com/sensors/introduction-pid-control.
6. "Fog Computing," *Wikipedia*, Wikimedia Foundation, 2018; https://en.wikipedia.org/wiki/Fog_computing.
7. "Edge Computing," *Wikipedia*, Wikimedia Foundation, 2018; https://en.wikipedia.org/wiki/Edge_computing.
8. D. Greenfield, "Fog Computing vs. Edge Computing: What's the Difference?," *Automation World*, 2016; www.automationworld.com/fog-computing-vs-edge-computing-whats-difference.
9. M. Iorga et al., "Fog Computing Conceptual Model," *NIST Special Publication 500-325*, 2018.
10. M. Enescu, "Paradigm Shift with Edge Intelligence," *Open at Cisco*, 2014; blogs.cisco.com/openatcisco/paradigm-shift-with-edge-intelligence.
11. *Web Connectivity and Security in Embedded Systems*, Coursera; www.coursera.org/learn/iot-connectivity-security/lecture/bYkez/fog-computing.
12. S. Yi, Z. Qin, and Q. Li, "Security and Privacy Issues of Fog Computing: A Survey," *10th International Conference on Wireless Algorithms, Systems, and Applications (WASA)*, 2015; www.cs.wm.edu/~zhengrui/papers/wasa15-fog.pdf.
13. *The OpenFog Consortium Reference Architecture: Executive Summary*, 2017; www.openfogconsortium.org/wp-content/uploads/OpenFog-Reference-Architecture-Executive-Summary.pdf.
14. *IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing*, standard IEEE Std 1934-2018, The IEEE Standards Association, 2018.

ABOUT THE AUTHOR

Christine Miyachi is a systems engineer at Xerox Corporation and holds several patents. She works on Xerox's Extensible Interface Platform, which enables developers to create applications that work with Xerox devices by using standard web-based tools. Miyachi graduated from the University of Rochester with a bachelor's degree in electrical engineering. She holds two MIT degrees: a master's in technology and policy/electrical engineering and computer science and another master's in system design and management. Contact her at cmiyachi@alum.mit.edu.