the Availability Digest

www.availabilitydigest.com
@availabilitydig

Fog Computing Improves Application Availability

June 2017

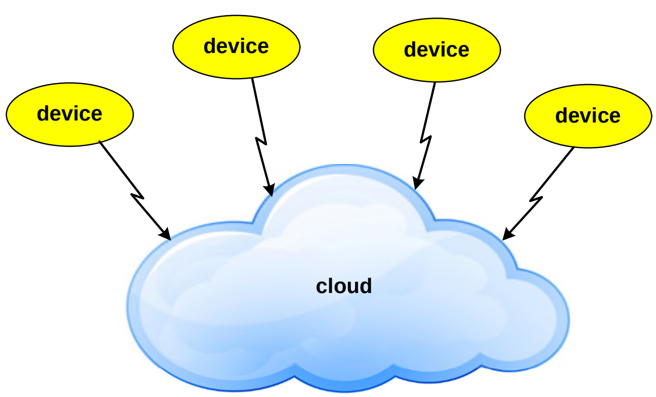
Fog computing is a decentralized computing architecture in which the applications and the data that they process are distributed in the most efficient places between the data source and the cloud. Fog computing[1] extends the services provided by the cloud to the edge of the network via ‘cloudlets’, bringing the advantages and power of the cloud closer to where the data is created. Fog computing allows data to be accessed more efficiently and processed more rapidly and reliably from it most logical location.

Equally important is the increase in availability that fog computing brings to applications. Should the cloud fail, an event that normally would bring down the applications, the distributed nature of fog computing can allow the critical aspects of application processing to continue via the cloudlets with little impediment.

The Limitations of Cloud Computing

Since the 1960s, computing has alternated between centralization and decentralization. In the 1960s and 1970s, centralized batch processing and time sharing dominated processing architectures. These techniques gave way to decentralized personal computing in the 1980s and 1990s. Cloud computing returned processing to a centralized paradigm in the 2000s. Now, fog computing is decentralizing computing.

Cloud computing provides the ability for devices to store data at off-site locations and to retrieve data from these locations. Two of the value propositions offered by cloud computing include the exploitation of economies of scale for large cost savings and the ability for organizations to avoid the costs of creating and running their own data centers.



Cloud Computing

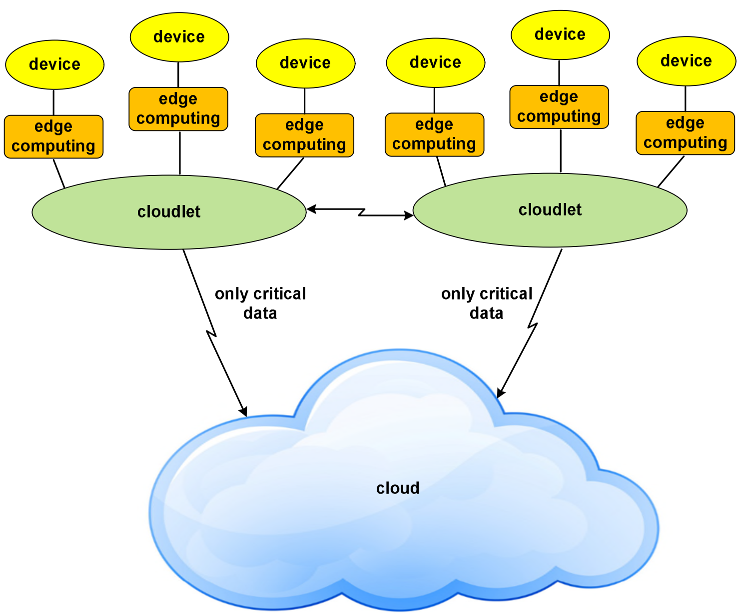
[1] ‘Fog computing’ is also called ‘edge computing.’

A problem with cloud computing is the bandwidth consumed by the myriad devices communicating with the cloud. The Internet of Things (IoT) depends upon cloud computing for the massive amount of processing required for hundreds of thousands of devices. Each of these devices connects directly to the cloud, consuming massive bandwidth in total. Furthermore, sending data to and receiving data from the cloud imposes tens of milliseconds of latency between the devices and the cloud, potentially compromising time-sensitive applications such as voice recognition.

Today, only about a third of the data collected by the growing array of IoT devices is analyzed at the source. The rest is transmitted in raw form to a cloud. This has got to change as the IoT grows. The problem is only going to increase as more and more objects connect wirelessly. This problem is solved by fog computing.

Enter Fog Computing

With fog computing, a significant amount of data processing capability is located at or near the devices themselves. Each device is typically monitoring a sensor and sends data only when the sensor data changes. Each device contains its own processing capability, which we call ‘edge computing.’ The edge computing processors control the devices and filter out data changes that are meaningless. When a change in the sensor data is perhaps meaningful, the data change is sent to a local ‘cloudlet’ for further processing. The edge processors convert the proprietary communication protocol used by the devices to the protocol used by the cloud.



Fog Computing

The Role of Cloudlets

Cloudlets can run arbitrary code. They preprocess the device data and send only critical data to the cloud. The devices can communicate with each other through the cloudlets to alert other devices of their status changes so that the other devices can take appropriate action. Thus, devices do not have to communicate with the cloud to discover status changes in other devices that may affect their behaviour. The result is that much less data is sent to the cloud, lowering the data storage requirements that must be provided by the cloud. Also, bandwidth, communication latency, and communication jitter are significantly reduced.

Fog computing significantly reduces the bandwidth consumed by device communication because device data is preprocessed locally. In addition, communication latency is virtually eliminated. Device data is processed and used in real time.

Advantages of Fog Computing

The proximity of cloudlets to the devices helps in several distinct ways:

- <u>Cloud services are highly responsive.</u>

- <u>The system is highly scalable.</u>

- <u>Privacy policies can be enforced.</u>

- <u>Cloud outages are masked.</u>

Let us explore these advantages of fog computing in more detail:

Highly responsive cloud services:

Because the communication paths are short, there is little latency, and response-time jitter and bandwidth requirements are significantly reduced when compared to communications with a distant cloud. Extensive computing capability is brought to within one hop of a device. This enables applications that are both computation intensive and latency sensitive.

Scalability through edge analytics:

The bandwidth required to feed information to the cloud is considerable lower if the device data is analyzed in the cloudlets. Only the critical information that is the result of this processing need be transmitted to the cloud. Therefore, many more devices or more complex devices can be accommodated in any given communication environment.

It has been estimated that in some cases, edge analytics can reduce the incoming bandwidth requirements to the cloud by three to six orders of magnitude for some applications. A good example is the preprocessing of video camera feeds. The cloudlet can process the raw video and send only pertinent events to the cloud. For instance, it might use facial recognition to search for the presence of certain individuals and send only those video segments to the cloud.

Privacy policy enforcement:

The cloudlet is the first point of contact for sensor data. Therefore, the cloudlet can enforce the privacy policies of the devices' owners prior to the release of their data to the cloud. Users should be able to prevent a subset of sensor data deemed sensitive from being transmitted to the cloud. Alternatively, data can be 'denatured' by a cloudlet to make it safe before sending it to the cloud. For instance, video images can be blurred. Sensor readings could be aggregated and only the aggregates sent to the cloud, masking the individual sensor readings.

The full data could be archived in the cloudlet for a given period of time in case an anomaly is discovered that requires more data.

Masking cloud outages:

Critical fallback services can be embedded into the cloudlets to let them mask to some extent the failure of the cloud. Because of the physical proximity of cloudlets to their devices, the survivability characteristics of cloudlets are closer to their associated devices than to the distant cloud. If the cloud should fail, the cloudlets are likely still to be functioning. Therefore, fallback services in the cloudlet can temporarily mask cloud inaccessibility.

During a cloud failure, cloudlets can perform the critical services normally provided by the cloud. Upon repair of the cloud failure, actions taken by the cloudlets are propagated to the cloud to resynchronize it with the changes in state created by the cloudlets.

Disaster Recovery

As described to some extent above, fog computing brings a degree of disaster recovery to applications. In a normal cloud environment, should the cloud fail, all the applications that are dependent upon the cloud also fail.

However, with fog computing, should the cloud or the network fail, critical processing tasks can continue. A cloud disaster is reduced to the loss of certain processing functions, but critical processing functions can be continued with the cloudlets.

The Amtrak Train Control System

In the 1980s, my company, Sombers Associates, Inc., built for Amtrak a Tandem-based train control system. The system allowed train controllers to manage train traffic along the Northeast Corridor from Philadelphia to Boston.

The status of every signal and the position of every switch, along with the position of every train, was sent to the central control system. Each train controller was provided a set of consoles that showed the track model for the particular portion of the rail system for which he was responsible. A train controller could control the signals and switches and could view the progress of all trains in his area of responsibility.

If the central system failed or if there was a major network outage, the train controllers moved to the basement and gathered around a massive board showing the entire train system. From there, the controllers could manually control signals and switches via toggle switches and observe the movement of trains.

Fog computing was not available back then. If it had been, the area of responsibility for each controller could have been presented on his consoles via a cloudlet. Only if a cloudlet failed would that particular controller have to go to the manual board to manage his section of track. Of course, a cloudlet requires a small data processing system in the field in the area in which it is managing. Such small computers are available today, but they did not exist in the 1980s. It would not have been possible then to implement the train control system with fog computing.

Today, this system would look totally different, with cloudlets supporting each train controller and communicating amongst themselves to pass trains from one track section to another.

Summary

Any business relying on storing data in someone else's data center should consider how their business might be affected by the future lack of bandwidth to access it. Fog computing takes a step forward in reducing the bandwidth required by classic cloud computing. In addition, it offers a degree of protection from cloud failures since critical tasks can still be carried out by the cloudlets.

Acknowledgements

Information for this article was taken from the following sources:

What is Fog Computing? And Why It Matters In Our Big Data and IoT World, *Forbes*; October 14, 2016.
All you ever wanted to know about fog computing, *TechRadar*; May 3, 2017.
The Emergence of Edge Computing, *Outook*; undated.
Fog Computing vs. Edge Computing: What's the Difference?, *Automation World*; undated.