May 2022

Distributed Computing Using Dynamic Local Network of Edge Devices

n/a

Distributed Computing Using Dynamic Local Network of Edge Devices

ABSTRACT

Conventional cloud computing operation involves direct communication between nodes in the cloud and individual client devices. Such an approach is inefficient for handling situations in which cloud nodes or client devices may be occasionally offline. This disclosure describes an architecture that provides operational flexibility by forming a dynamic local network of multiple client devices associated with a user that provides redundancy and avoids computing bottlenecks. The dynamic local network is used to distribute computations in a manner that can overcome the bottlenecks that result from a cloud node or client device being offline. When a cloud node is offline, one or more client devices can hold and run the required computation. Similarly, if a targeted client device is offline, a cloud node can allocate the task to another client device associated with the user. Such operations can lower feature subscription costs and reduce latency, thus enhancing the user experience.

KEYWORDS

- Cloud computing
- Distributed computing
- Edge computing
- Ubiquitous computing
- Ambient device
- Dynamic local network
- 6LoWPAN
- Low-power wireless personal area network

BACKGROUND

Conventional cloud computing operation involves direct communication between nodes in the cloud and individual client devices, such as smartphones, smart appliances, smart speakers, etc. Such operational implementation is used for edge computation of a wide variety of computational problems ranging from small to large. However, the approach is inefficient in situations in which cloud nodes or client devices are offline for any reason, such as crash, empty battery, etc. If a cloud node is offline, the client device is stuck until it is back online, for instance, after rebooting following a crash. While it is possible to handle such situations by providing redundancy via backup cloud nodes, such resources are not always available.

If the client device is offline, the download from the cloud occurs only after the device is back online, which can lead to a perceptible increase in latency if the client computation is needed for user interface (UI) elements. As a result, in current operational deployments of cloud computing, offline cloud or edge nodes can degrade the user experience (UX) because the single receiving device on the client side presents a bottleneck.

DESCRIPTION

This disclosure describes a cloud computing architecture that provides operational flexibility by leveraging multiple client devices a user owns. If the user permits, the various devices a user owns, such as a smartphone, a smart speaker, a smart thermostat, etc. are configured to form a dynamic local network to avoid bottlenecks inherent to conventional cloud computing architectures that have single cloud node to client branches.

The dynamic local network is used to pass computations in a manner that can avoid the bottlenecks that result from a cloud node or client device being offline. For instance, if a central cloud node is offline, a user-owned device that has sufficient amount of available computational resources can hold and run the computation to be allocated to a targeted client device. Similarly, if the targeted client device is offline, a cloud node can allocate the task to another user-owned edge device. Whenever the targeted client device is back online, the compute task or results can be transferred to it from the substitute edge device much faster because of their proximity.

Further, a fully-connected local compute network can be devised among user-owned edge devices that are capable of forming local low-power wireless personal area networks, such as IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN). Such networks of local devices that are compatible with the Thread protocol [1] (or other similar protocol) can add an extra layer of robustness in case any of the local devices fails in downloading or computing a task. In such cases, the task can be passed along a cycle within the local network to meet the needs of the client.

For instance, if the targeted client device, such as a smartphone, is not discovered because it is offline, another user-owned device within a fully-connected local compute network can serve as the integration point by receiving the compute task to be performed at the edge. If the task received from the cloud is sufficiently large, the integration point can offload the work to other devices within the local compute network via Thread or similar local network protocols with low power consumption. The individual tasks performed at the edge devices within the local network are aggregated by propagating the results back to the edge device that serves as the integration point. The results are then relayed to the targeted client device whenever it is online. Since the user-owned ambient edge devices within the local network perform the computation while waiting for the targeted client device to be online, the client device can receive the results without needing to perform the task, resulting in low UI latency from the user's point of view.

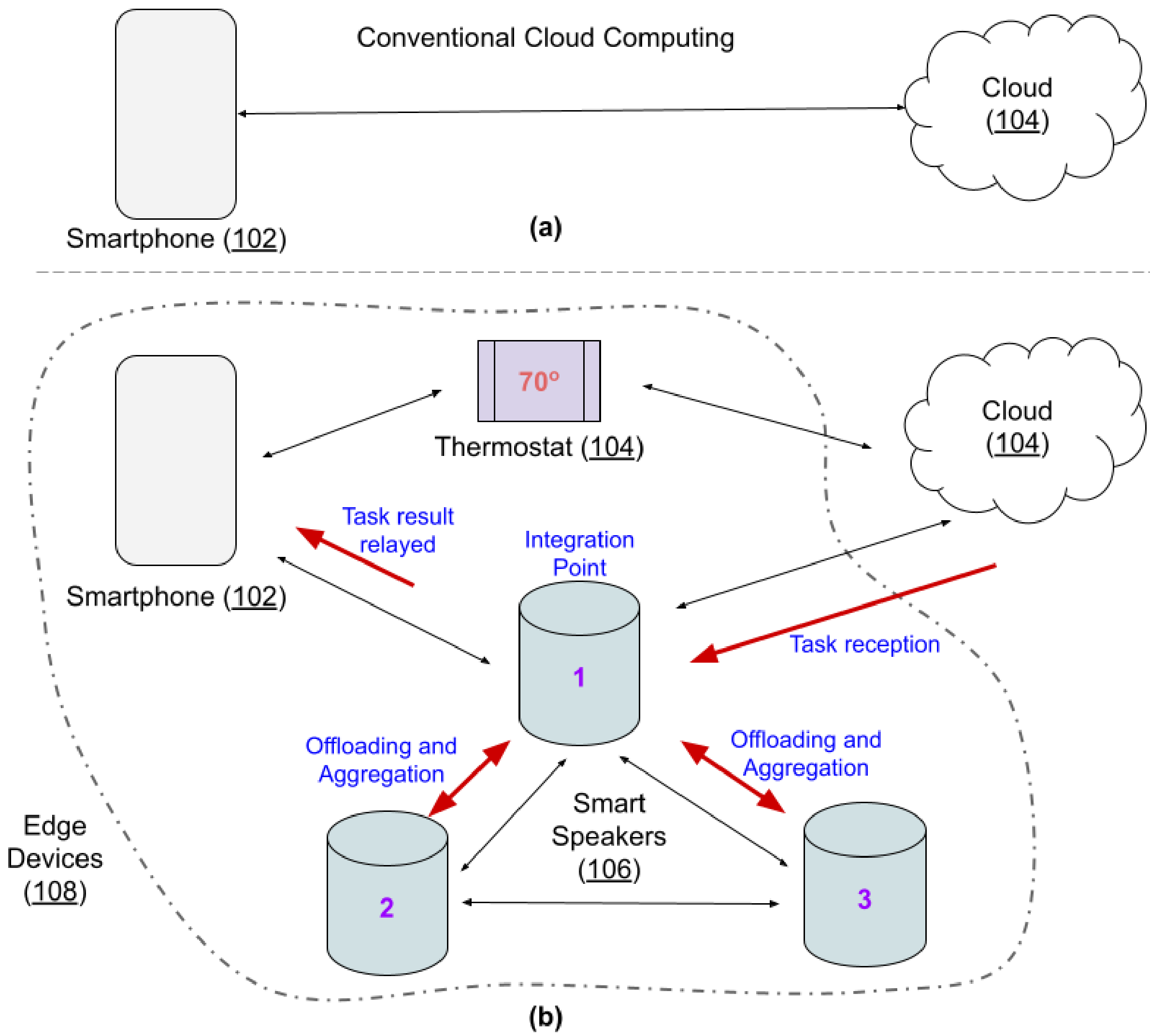

Fig. 1: Performing compute tasks on locally connected edge devices until target is available

Fig. 1(a) shows the conventional cloud computing approach limited to one-to-one communication between an edge device such as a user's smartphone (102), and the cloud (104). In contrast, Fig. 1(b) depicts an operational implementation of the techniques described in this disclosure, showing the communications occurring between the cloud and a local network of user-owned edge devices (108) such a smartphone, a thermostat (104), and three smart speakers (106).

If the smartphone happens to be offline when a compute task is sent from the cloud, the task can be received by one of the smart speakers (e.g., speaker 1) that acts as an integration point. If necessary, the task can be offloaded to other locally connected devices, e.g., smart speakers 2 and 3 in Fig. 1(b), and the results can be aggregated. The aggregated task computation is relayed to the smartphone when it is back online, thus creating a seamless and low-latency user experience.

The techniques can be implemented within any architecture in which individual users are associated with multiple static and/or portable edge devices. The use of dynamic, ambient local networks formed by user-owned edge devices as an extra holder for compute tasks can lower feature subscription costs since cloud resources are typically charged based on the computational resources used. Implementation of the techniques described in this disclosure user can improve reliability and can reduce operational latency from the user's point of view, thus enhancing the user experience.

CONCLUSION

1. "Thread (network protocol) - Wikipedia" available online at https://en.wikipedia.org/wiki/Thread_(network_protocol) accessed Mar 22, 2022.