# ”Leveraging Edge Compute for Cloud Application Scalability”

Tech Con 2025 Abstract 639

HPE Edge;

# Leveraging Edge Compute for Cloud Application Scalability

## Abstract

*Data sent inefficiently to cloud applications can negatively impact HPE's profitability by driving up cloud costs and shrinking margins. To address this, we propose a cloud-based data optimization service that partitions wireless access points sending data to cloud application based on data correlations, allowing APs in the same group to share common data with a designated leader, who consolidates it before sending it to the cloud. This reduces the number of messages sent to the cloud, lowering costs. Using simulations, we demonstrate that our approach can reduce the number of messages by 50%, enabling better scaling of cloud applications.*

## Problem statement

The IoT Operations (IoTOps) solution within Aruba Central is used for the deployment and management of Internet of Things (IoT) applications. IoT applications utilize technologies such as BLE, Zigbee, Wi-Fi RTLS, and other IoT proprietary standards with the goal of addressing customer use cases such as indoor localization, smart door locks, sensor and habitat monitoring, electronic shelf labels, etc. IoT applications can either be deployed on a dedicated virtual machine appliance per site or on individual Aruba wireless access points (APs) (see Figure 1). Due to cost concerns, the AP-based deployment mode (AP connector) is preferred by most Aruba Central WLAN customers. In a typical deployment, each Aruba Access Point (AP) is configured as an "AP connector" reporting IoT device data, IoT application-level metrics and logs to IoTOps periodically. Each AP connector can process data from around 500 BLE devices, 32 Zigbee devices and a few proprietary USB gateways. IoTOps displays time-series data for each IoT device such as signal strength, transmit/receive messages, etc. Additionally, IoTOps pre-processes the (RSSI/signal strength) data generated by the IoT devices before handing it over to other Central applications such as Asset Tracking, which are responsible for computing IoT device location.

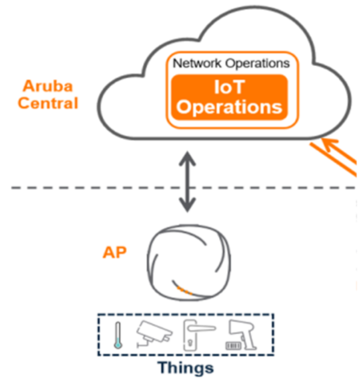


**Figure 1: Overview of AP connector-based IoTOps Deployment.**

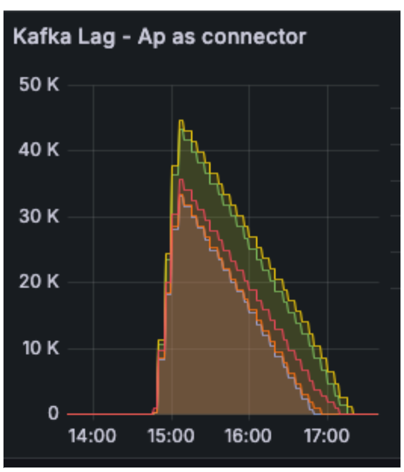


**Figure 2: Kafka lags observed when 10K APs send device reports to IoTOps. (X axis shows time and Y-axis shows the number of lagging/unprocessed messages.**

In customer deployments where we have thousands of APs sending data to IoTOps, we will start to see issues crop up with all APs sending their periodic and constantly changing status messages, and IoT data reports to Aruba Central. Figure 2 shows the Kafka data processing lag for the "ii-ae-devices" IoTOps service before and after a simulation of 10K APs sending device reports every 60s for 10minutes. Note that it takes significant time (nearly two hours) after a 10 minute run to clear the lag. This clearly indicates that the current deployment cannot scale to 10K APs with the existing resources. Furthermore, these numbers fall well short of the expected 20K scale number for AP connectors. The current IoTOps microservices configuration deploys 5 replicas to handle device report messages from AP connectors (2 CPUs and 2GB RAM per replica plus 1 CPU and 8GB RAM for 1 KairosDB instance), and 2 replicas to handle RSSI messages (0.75CPU and 2GB RAM per replica). For reference, a "m6g.2xlarge" AWS EC2 instance with 8vCPUs and 32GB memory costs USD240 per month for an On-Demand instance (~USD120 for standard reserved instances). Doubling or quadrupling the scale of these microservices will incur higher cloud costs as each instance of the microservice necessitates additional infrastructure instances of Kafka, PostGres SQL, Redis, etc.

While some data, such as application metrics and connector information, is unique to each AP, device reports are particularly problematic because much of the information they contain is duplicated by neighboring, overlapping APs that report the same device to IoTOps. Processing this redundant data places unnecessary strain on cloud resources and individual microservices. Reducing this duplication would enhance microservice performance, enabling the system to scale to a greater number of APs using the same resources. Furthermore, since cloud infrastructure costs are borne by HPE, reducing these costs would directly benefit the company's bottom line.

## Our solution

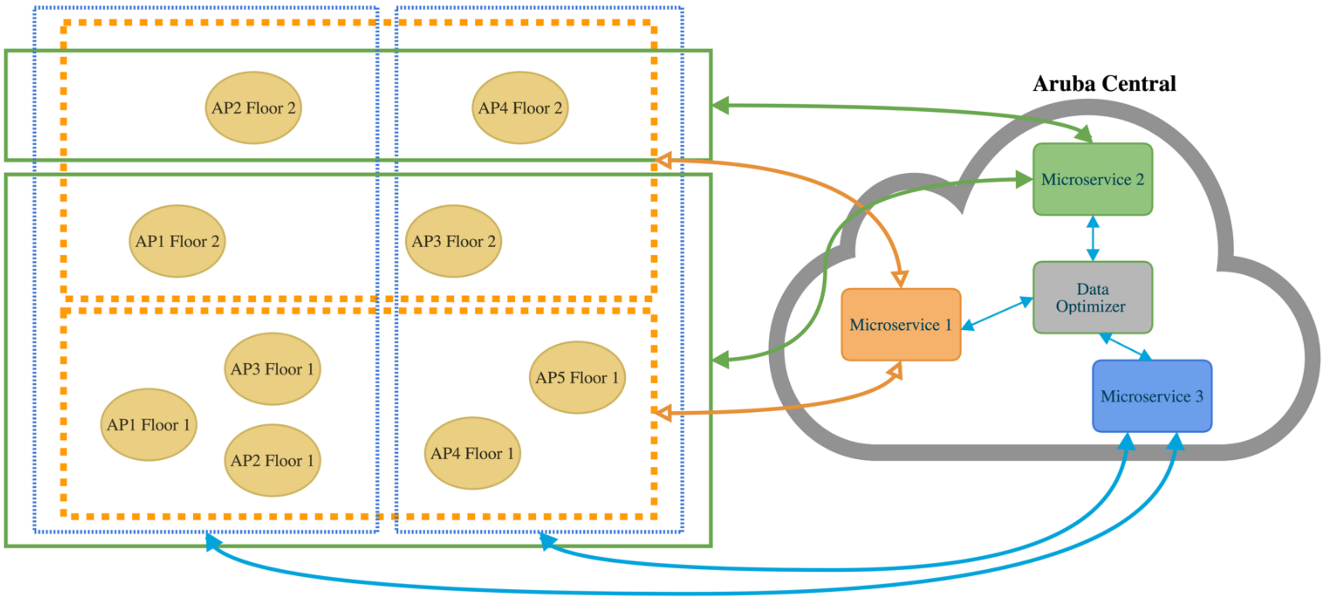


**Figure 3: Illustration shows the formation of multiple sets of groups with different member APs for optimizing data being transported to each cloud microservice.**

At a high-level, all the AP connectors in the network will be partitioned into groups based on the locality of information. Group members will exchange common data with a designated group member which is responsible for coalescing the data into fewer messages before forwarding to the cloud. Every AP connector is still responsible for forwarding the data that is unique to it to the cloud. While aggregating data on edge compute devices before forwarding to cloud is not a new concept, our solution expands this concept by taking into account the needs of multiple cloud applications simultaneously by forming overlapping functional groups based on the end consumer cloud applications. The details of our approach are as follows:

1. Each cloud application will begin by receiving data from individual AP. Each application will create a dataset of all the IoT devices (or sensor value attributes) categorized by AP. This dataset will be provided as input to our data optimization cloud service which will compute the data correlations between different APs. The resulting correlation matrix will be re-arranged into a number of groups such that within each group there is high degree of similarity between the members of that group. The data optimization service will return the groups with a designated group leader (AP with which other group members have highest degree of data similarity) and mark the common data attributes between leader and member APs.
2. Each cloud application will propagate the group information to the APs. For grouped APs, the "group leader" will operate an MQTT server service. Member APs will send common data to it over an MQTT connection. The "leader" will coalesce the data from group members with its own data reports and forward to the cloud application. Other group members will send their "unique" data separately to the cloud. Overall, we will reduce the number of data messages needed to be processed by the cloud since there is only one copy of the common/duplicate data being sent to the cloud.

Figure 3 shows an illustration of our approach highlighting the creation of multiple sets of groups specific to various cloud applications based on the similarity characteristics of data consumed by them. By having multiple applications taking advantage of our data optimization service, we can have multiple groups per application each minimizing data to their respective cloud application. The more applications that adopt our service, higher are the cost savings, as the same cloud infrastructure can scale to a bigger number of APs.

## Evidence the solution works

To show the effectiveness of our approach, we gathered BLE device data from multiple live QA testbeds in San Jose and Beijing. We simulated the behavior of our data optimizer service using scripts to process the BLE data and to create a similarity matrix from the data. Our service uses the similarity matrix to compute correlations between pairs of APs, and returns groups of APs with high degree of data similarity.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.00 | 0.36 | 0.70 | 0.34 | 0.37 | 0.32 | 0.32 | 0.41 | 0.30 |
| 2 | 0.24 | 1.00 | 0.41 | 0.68 | 0.69 | 0.88 | 0.86 | 0.71 | 0.81 |
| 3 | 0.72 | 0.63 | 1.00 | 0.64 | 0.65 | 0.61 | 0.59 | 0.61 | 0.57 |
| 4 | 0.29 | 0.88 | 0.54 | 1.00 | 0.91 | 0.89 | 0.85 | 0.73 | 0.83 |
| 5 | 0.33 | 0.91 | 0.56 | 0.93 | 1.00 | 0.93 | 0.90 | 0.78 | 0.87 |
| 6 | 0.22 | 0.91 | 0.40 | 0.71 | 0.73 | 1.00 | 0.92 | 0.75 | 0.87 |
| 7 | 0.22 | 0.90 | 0.40 | 0.70 | 0.72 | 0.94 | 1.00 | 0.76 | 0.90 |
| 8 | 0.36 | 0.96 | 0.54 | 0.76 | 0.80 | 0.98 | 0.97 | 1.00 | 0.96 |
| 9 | 0.20 | 0.82 | 0.37 | 0.64 | 0.66 | 0.85 | 0.87 | 0.72 | 1.00 |

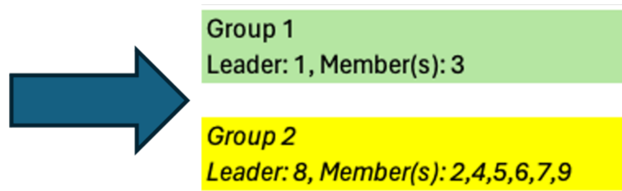


**Figure 4: The correlation matrix shows the similarity between BLE device data observed by the APs, and on the right we show the groups computed by our program.**

Figure 4 shows one example of the group creation using BLE device data as input by our approach on a 9 AP testbed. Table 1 summarizes the reduction in number of messages sent to the cloud for BLE device data. With ~50% drop in the total messages sent to the cloud application, our approach demonstrates that it can significantly reduce the message processing lag encountered by services such as “ii-ae-devices”. The reduced lag will easily translate to savings in cloud infrastructure costs by enabling microservices to better scale to higher number of APs, with fewer EC2 instances.

| Total APs | Number of Groups | Total Messages | | Potential Savings |
|---|---|---|---|---|
| | | Before Grouping | After Grouping | |
| 9 | 2 | 2655 | 1110 | 41% |
| 13 | 4 | 6308 | 3398 | 53% |
| 19 | 3 | 391 | 210 | 54% |
| 30 | 5 | 15015 | 8117 | 54% |

**Table 1: For each testbeds, the table shows the number of similarity groups computed using our approach, the number of device update messages sent to cloud before and after the grouping takes effect and the ensuing savings in terms of number of messages.**

## Competitive approaches

As mentioned in the “Problem Statement” section, customers can deploy a dedicated IoTOps virtual appliance per site which would aggregate the data from all APs at that site before forwarding to the cloud. However, this approach is a non-starter for many small and medium business customers that have simple IoT application requirements, or for large customers that have a small number of APs distributed across multiple sites. Managing an extra network appliance translates into more work for the customer. The AP Connector is quite appealing for IoT customers that have less compute intensive and distributed IoT applications, which is the case for most IoT applications.

## Current status

We will present our simulation results to our PLM and leadership team, with the goal of having the proposed changes adopted for implementation in upcoming Central/CNX sprints.

## Next steps

In addition to minimizing BLE device data reports and BLE device RSSI messages, our approach can be utilized to compute the Zigbee coordinator neighborhood information. Solving this problem will improve the usability of the IoTOps Zigbee doorlock integrations with Assa Abloy and Dormakaba. The AirRange CNX application, which uses AP barometer sensor data to to determine whether APs are co-located on the same floor, can benefit by consuming data aggregated using our approach. We will continue to identify cloud applications that can integrate with our approach to optimize cloud costs.

# References