Project Proposal:
Adaptive Twin for the Data Thoughput Optimization using TinyML

Many IoT use cases involve data collected at high frequencies. The frequency of this data can vary significantly depending on the type of devices used. In many scenarios, the historical data is timestamped according to the acquisition time rather than the sampling period. This method can lead to potential cost savings by reducing throughput, enabling batching, and buffering data over time. However, this approach is challenging for low-energy devices, which have limited energy and memory capacity.

AI-based Optimization

To finalize the best-fit throughput and find the optimal trade-off between cost savings and edge energy optimization, a tinyML model can be employed. This model can identify the optimal solution for achieving the desired throughput on a specific low-energy device.

Model Parameters

An optimal solution involves maintaining or approximating the desired throughput on an edge device by considering the following parameters:

Memory Usage: This parameter relates to the buffering of data from connected devices. Throughput and batching algorithms can result in high memory usage.

Energy Usage: This is associated with memory I/O and CPU operations, which are limited by the device's performance.

Edge Device Frequency: This parameter defines the sampling rate of field devices and can be either constant or variable over time.

Given a desired throughput, the edge tinyML model can perform inferences to identify the best-fit, maximum, and minimum throughput based on the aforementioned input parameters. A desired throughput can be achieved if specific requirements are met, and an optimal solution (if it exists) can be found and maintained over time using the Ditto framework as a stateful twin.

Digital Twin Integration

The Ditto framework is a platform-based service that provides a Digital Twin through a thing description to maintain both the desired and current states of the monitored edge device.

Desired State: This is the optimal target for throughput optimization, given the current energy and memory usage.

Current State: This represents real-time metadata provided over time at different frequencies according to the current tinyML throughput.

Control and Monitoring

The current values of optimal and current parameters are monitored over time using a Grafana dashboard. A command-based API is provided to adjust the current desired throughput value.

Several constraints are applied in the Ditto configuration:

Minimum and Maximum Values: These are defined by the tinyML model for the input throughput (desired value), considering current energy and memory usage.

Throughput Change Frequency: Changes to the desired throughput value are subject to a timeout constraint to prevent continuous adjustments. This timeout is determined based on feedback from the edge device, processed by the tinyML model's output.

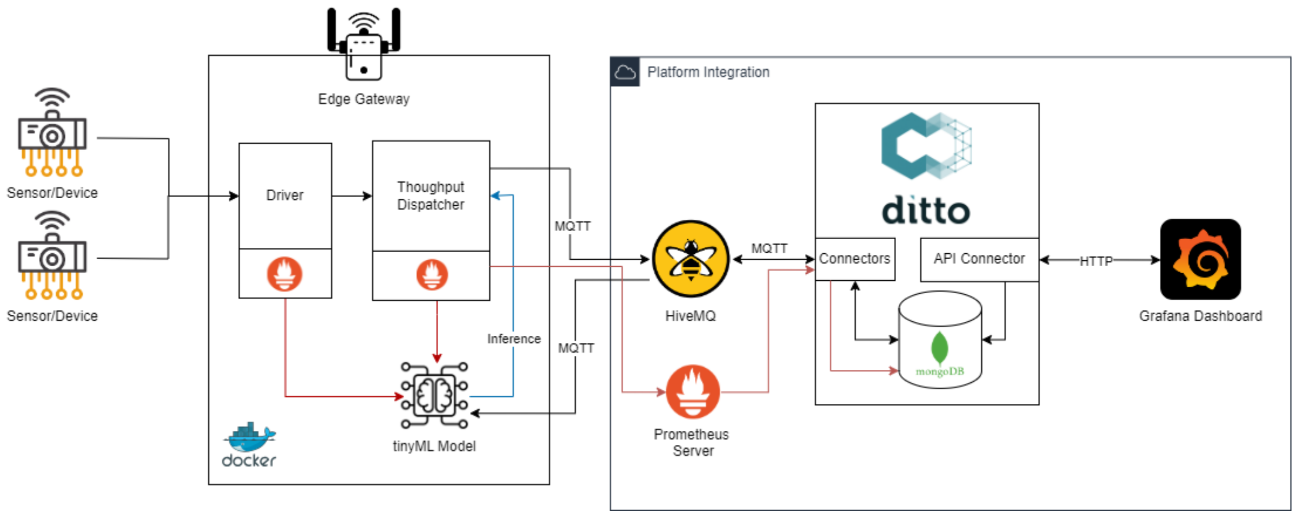

Figure 1 Architecture Overview

Architecture Overview

Here's a refined summary of the current architecture into four distinct layers:

Edge Device: Provides data at variable or fixed frequencies.

Edge Gateway: Acquires data and executes inference using the tinyML model to determine optimal throughput based on current average energy and memory usage.

Ditto Layer: Configures and maintains desired and current throughput values, while also managing real-time data for energy and memory usage.

Grafana and API Layer: Provides monitoring capabilities and command management to interface with the Ditto Twin.