Exploring the Impact of Workload Distribution in a Hybrid Edge and Cloud Application for Smart Grids

Otávio Carvalho, Manuel Garcia, Eduardo Roloff, Phillipe O. A. Navaux

Federal University of Rio Grande do Sul - Parallel and Distributed Processing Group (GPPD)

Table of contents

Introduction - Motivation

- **Smart Grids** potential to save billions of dollars in energy spending for both producers and consumers.

- **Internet of Things** potential economic impact.

- Technologies created for IoT are driving computing toward **dispersion**.
    - **Edge Computing**
    - **Cloudlets**
    - **Micro-datacenters**
    - **Fog Nodes**

Introduction - Main goals

- *Explore the potential performance improvements of moving computation from cloud to edge in a Smart Grid application*.

  1. What are the boundaries of our application architecture in terms of latency and throughput?
  2. To what extent is it possible to move our workload from cloud to edge nodes?
  3. Which strategies can be used to reduce the amount of data that is sent to the cloud?

Architecture and Implementation

- *Three-layered architecture:*
  - Cloud-layer
    - High latency processing.
    - Receives aggregated data from multiple edge nodes.
    - Composed by applications running on Linux VMs on Windows Azure.
  - Edge-layer
    - Low latency processing.
    - Receives data from multiple sensors and perform local processing.
    - Reduces the amount of data that needs to be sent to Cloud-layer.
    - Composed by ARM nodes (Raspberry Pi Zero W) connected to Wi-Fi.
  - Sensor-layer
    - Measurements only.
    - Produces a high amount of measurements that should be sent to Edge-layer for aggregation.
    - For evaluation purposes, our sensor measurements are pre-loaded into our Edge-layer nodes.

Architecture and Implementation

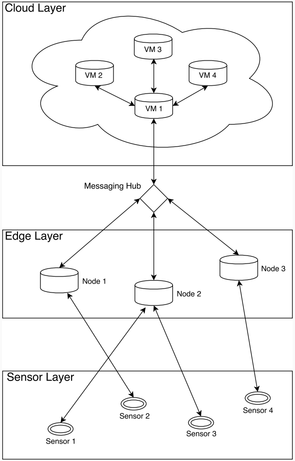

Figure 1: Architecture overview: Three-layered architecture

Evaluation - Communication

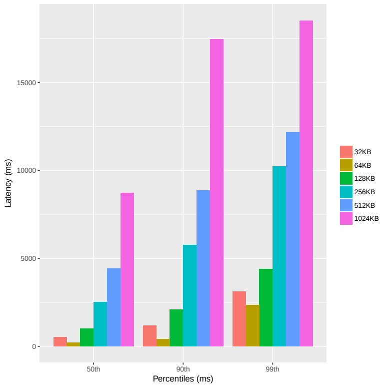

Figure 2: PingPong: Latency Percentiles by Message Sizes (32KB to 1MB)

Evaluation - Communication

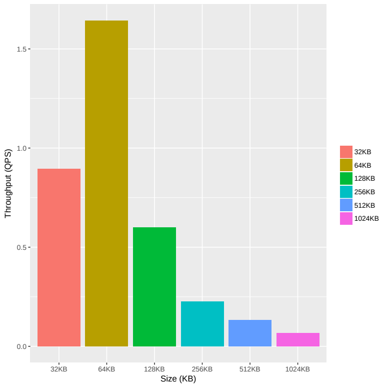

Figure 3: PingPong: Maximum Throughput by Message Size (32KB to 1MB)

Evaluation - Application concurrency

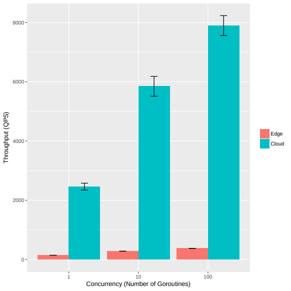

Figure 4: Concurrency Analysis: Impact of Goroutines usage on throughput (Edge and Cloud nodes)

Evaluation - Application scalability

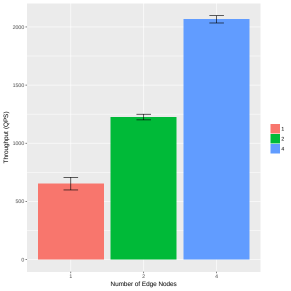

Figure 5: Scalability Analysis: Throughput with multiple consumers (1 to 4 edge nodes)

Evaluation - Workload windowing

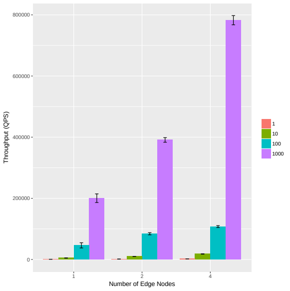

Figure 6: Windowing Analysis: Windowing impact on throughput (1 to 1000 messages per request)

Conclusion and Future Works

- Conclusion
  - The application was able to achieve a higher throughput by leveraging processing on edge nodes.
  - We were able to reduce communication with the cloud by aggregating data at edge level.
- Future Works
  - Study how other communication protocols (such as MQTT) would behave in this application context.
  - Explore techniques and models for adaptive workload scheduling.
  - Evolve the application architecture to a general framework for IoT.

Thanks! Questions?