Viability of Azure IoT Hub for High Velocity Large Scale IoT Data

Wajdi Halabi
John Hill
Ken Kennedy
Linh Ngo
Daniel Smith
Jason Anderson
Brandon Posey
Amy Apon

CLEMSON UNIVERSITY

Contents

- Motivation
- Azure IoT Hub
- Software and Architecture
- Experiments and Results
- Conclusion

Motivation

- Digital Plants and Industry 4.0
  - Sensors could send data to the cloud for analysis, allowing for predictive maintenance and status dashboards
- We studied the viability of Microsoft Azure IoT Hub for this task
- We worked closely with BMW partners to assess the needs of a manufacturing plant
- We used the Clemson supercomputer to generate a large workload that would resemble thousands of sensors

Azure IoT Hub: Overview

- IoT Hub is a managed service
- 1 IoT Hub can have multiple deployed instances called ***units***
- Units increase throttling limits of IoT Hub linearly
  - Example: 3 units = 3x higher throttling limit

Azure IoT Hub: Sensors

- Sensors have a unique ID
- They also have a secure access token
- They communicate with IoT Hub using the MQTT protocol*
- Sensors link to the IoT Hub, not any individual unit

*HTTPS and AMQP also options

Azure IoT Hub: Pricing

- IoT Hub offers 2 tiers: **Basic** and **Standard**
- Each tier has 3 editions (1-3)
- Pricing is a flat monthly rate based on tier, edition, units, region, and number of days provisioned

Azure IoT Hub: Throttling

- Throttling limits are determined by ***edition***
- Example:
  - Edition 3 (Basic or Standard)
  - 6000 send operations / sec per unit
  - 300,000,000 messages / day per unit

Azure IoT Hub: Partitions

- IoT Hub can have 4-32 partitions if created through the online portal
  - 128 possible through the Azure CLI
- Partition count doesn't affect price
- Partition count fixed
- Each sensor is hashed to a specific partition
  - More on this later

Software and Architecture: Supercomputer

- Multi-node high performance computing cluster
- Enabled us to emulate thousands of sensors with up to 10 nodes
- Pings from Palmetto to an Azure East US 2 VM had an average latency of 20 ms

Software and Architecture: Client Data Generator

- Represents a single physical sensor
  - Thousands of generators simulate thousands of sensors
- C++ for low memory footprint
- Accommodates parameters to mimic real sensor behavior
  - Will cover in our experiments

Software and Architecture: Generator Validation

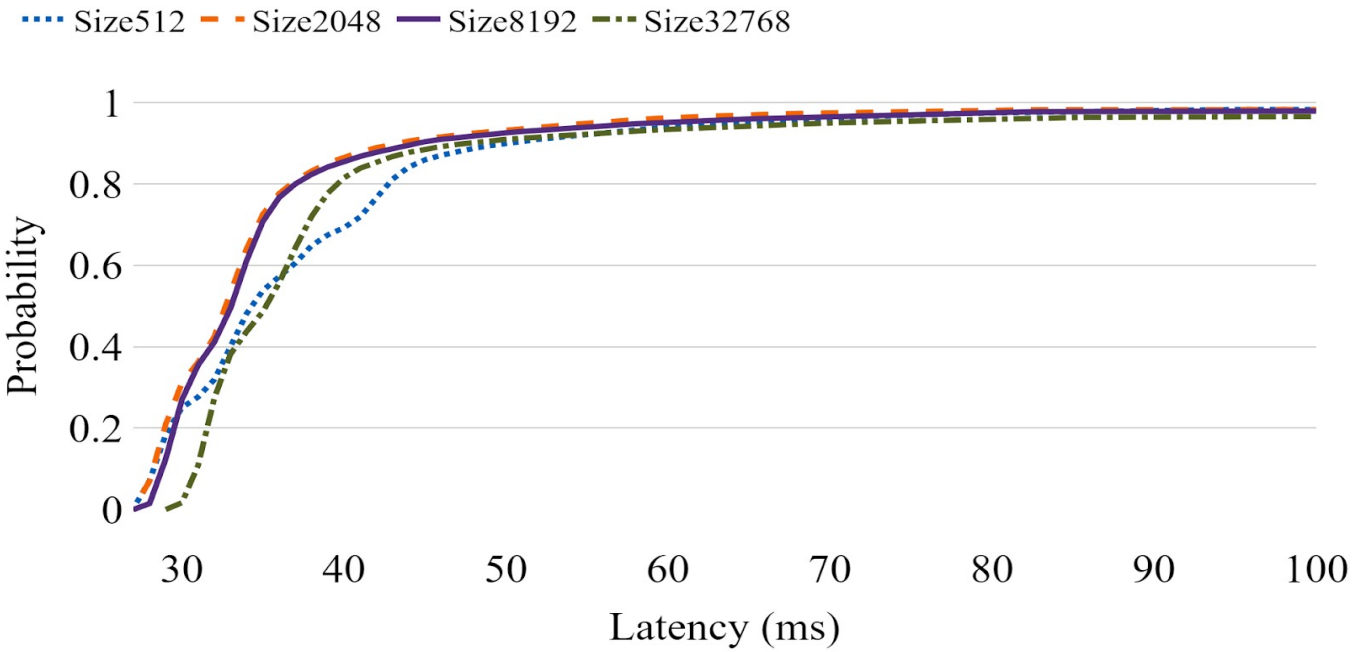
- Generator gives intermessage gap times that follow a statistical distribution
  - Constant or Pareto
  - Specified in parameters
- We used tcpdump to verify this by measuring packet send times
- We confirmed the distributions of the generated intermessage gap times were the same as those specified in the parameters

The Experiment Loop

- Sensor generates and sends JSON string to IoT Hub
- Send time is logged
- A new async thread is created for every response
  - Response time and status is logged
- Main thread sleeps between message sends
- After elapsed time, main thread sends another message
- Loop until all messages are sent
- Measure round trip latency after steady state achieved
  - First 5% of messages are dropped

Effects of Varying Message Sizes

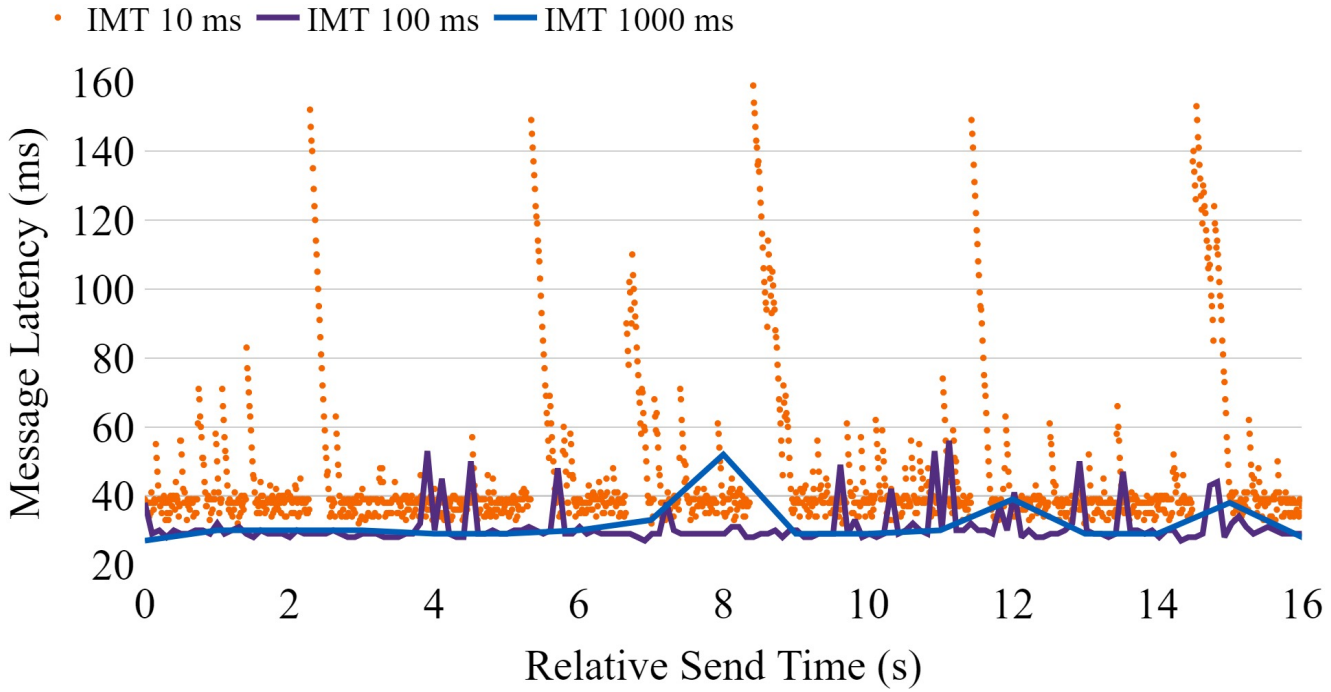
- *Var*: MsgSize=512B,2048B,8192B,32768B; Basic Edition 1, Basic Edition 2, Basic Edition 3
- *Const*: 10 sensors; IMT=200ms; RT=120s
- B1, B2, B3, if kept within the throttling limits, follow similar patterns
- 95% of messages have a latency less than 60ms



Latency CDF of 10 sensors, Edition 3, Constant IMT of 200ms

Effects of Varying Intermessage Gap Time

- *Var:* IMT=10ms,100ms,1000ms
- *Const:* 2048B; B3; RT=300s
- 100 ms and 1000 ms had few spikes while 10 ms had frequent spikes
- All messages from a single sensor go to the same partition, so messages were flushed

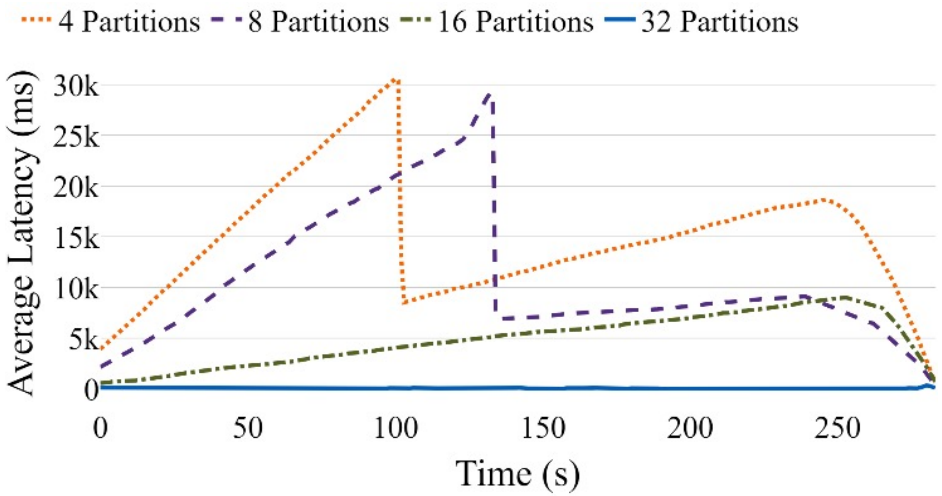


Latency of 1 sensor, 4 partitions, constant IMT distributions

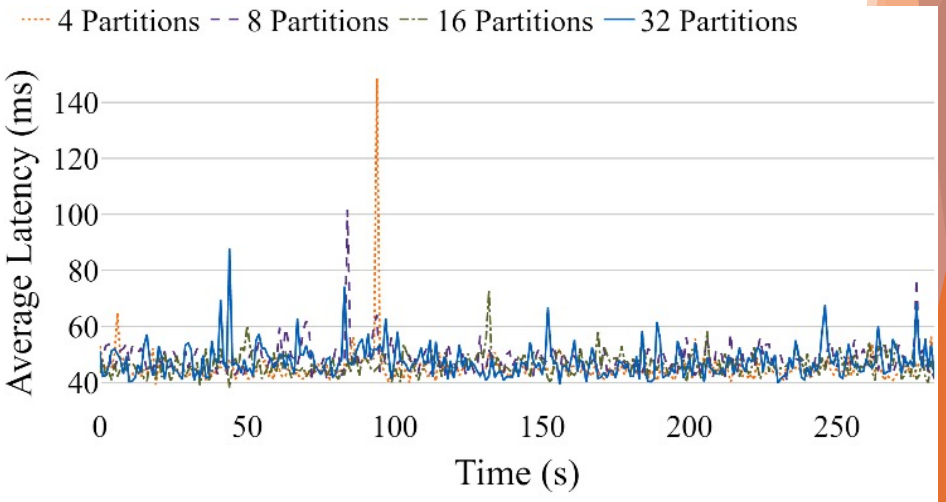
CLEMSON UNIVERSITY

Effects of Varying Partition Count

- *Var*: IMT=10ms,333ms; partition_count=4,8,16,32
- *Const*: 2048B; B3; RT=300s
- There is no load balancing between partitions, so in worse case all messages might go to one partition
- Confirms that IoT Hub is best equipped to handle large number of sensors sending at modest rate



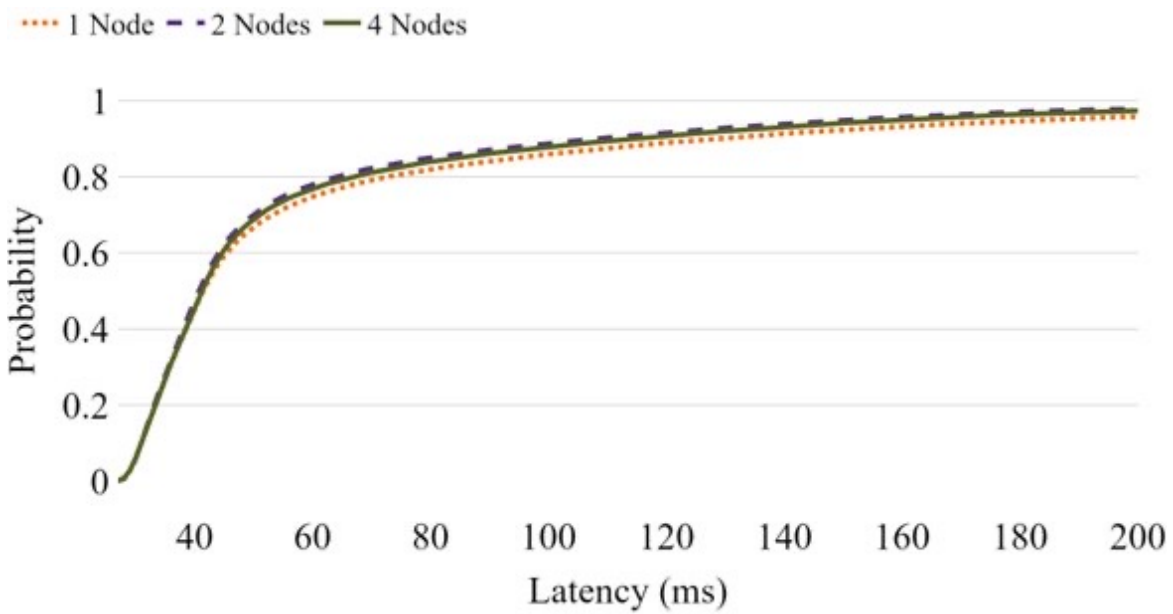
Group A Latency mean of 10 sensors, constant IMT of 10 ms
100 msgs/sec/sensor



Group B Latency mean of 1000 sensors constant IMT of 333 ms
~3 msgs/sec/sensor

Scaling Experiments

- *Var:* Sensors=2000,4000; IoTHubUnits=2,4; ComputingNodes=2,4
- *Const:* IMT=200ms; 2048B; B3 IoT Hub; 4 partitions
- 10,000 msg/s and 20,000 msg/s
- With multiple units, we kept below the throttling limit (6000 msgs/sec/unit), at 80%
- IoT Hub behaved stable, with 85% of messages with a latency < 100ms



Latency CDF of 2000 Sensors, Constant IMT=200ms, 4 partitions, 2 B3 Units

CLEMSON UNIVERSITY

Conclusion

- IoT Hub is designed to scale horizontally
  - Benefits manufacturing plants as more sensors connect to the cloud
- Individual sensors should avoid sending at high frequency
- Relatively simple to determine the configuration and flat rate cost of an IoT Hub deployment
- Our generator's source code, scripts, and data are public at github.com/aapon00/ltb2021

Thank you!
Questions?

- Wajdi Halabi, whalabi@clemson.edu
- Daniel Smith, dnsmith@clemson.edu
- John Hill
- Jason Anderson
- Ken Kennedy
- Brandon Posey
- Linh Ngo
- Amy Apon, aapon@clemson.edu

CLEMSON UNIVERSITY