

PM473593

Fusion と Forge を用いた CNC ツールパスの最適化。IoT によるリアルタイムでの製造データ収集のデモ

加藤 丈博

オートデスク株式会社

学習の目的

- 問題領域の理解
- ソリューションの理解
- MQTT ブローカーのビルド
- IoT センサーデバイスのビルド
- IoT センサーファームウェアのビルド
- Forge アカウントへのモデルのアップロード
Forge Viewer Sample Project のカスタマイズ

説明

このクラスのデモでは、アルミニウムを加工する際にホビークラスのコンピュータ数値制御（CNC）ルータからリアルタイムの製造データ（切削温度）を取り込み、そのデータを製造する部品のビジュアルモデル上に可視化するという簡単な例を概観します。切削工具とワークピースの温度は、切削効率と工具の健全性を直接示しています。

マッピング温度は、切りくずの公称荷重を実現するための送り、速度、切込み深さの調整を検証し、提案するのに役立ちます。産業用ソリューションとはかけ離れていますが、このデモでは、Forge で構築されたクラウドベースの製造ソリューションの有用性を探るために、IoT 製造のパワーを簡略化したコンテキストで紹介します。

スピーカーについて



スピーカーについて

Nathan Skalsky

Nathan Skalskyは、オートデスクのForgeのシニアプロダクトマネージャであり、モノづくりを愛しています。自動化と簡素化により、人間の創造性をより効率的に表現できるワークフローと機能を実現しています。

IBMでは大きなシステムの一部の設計を担当し、その後、Fusion 360のソフトウェア開発マネージャーや、スタートアップも経験しました。現在はオートデスクでFusionのプロダクトマネージャをしています。



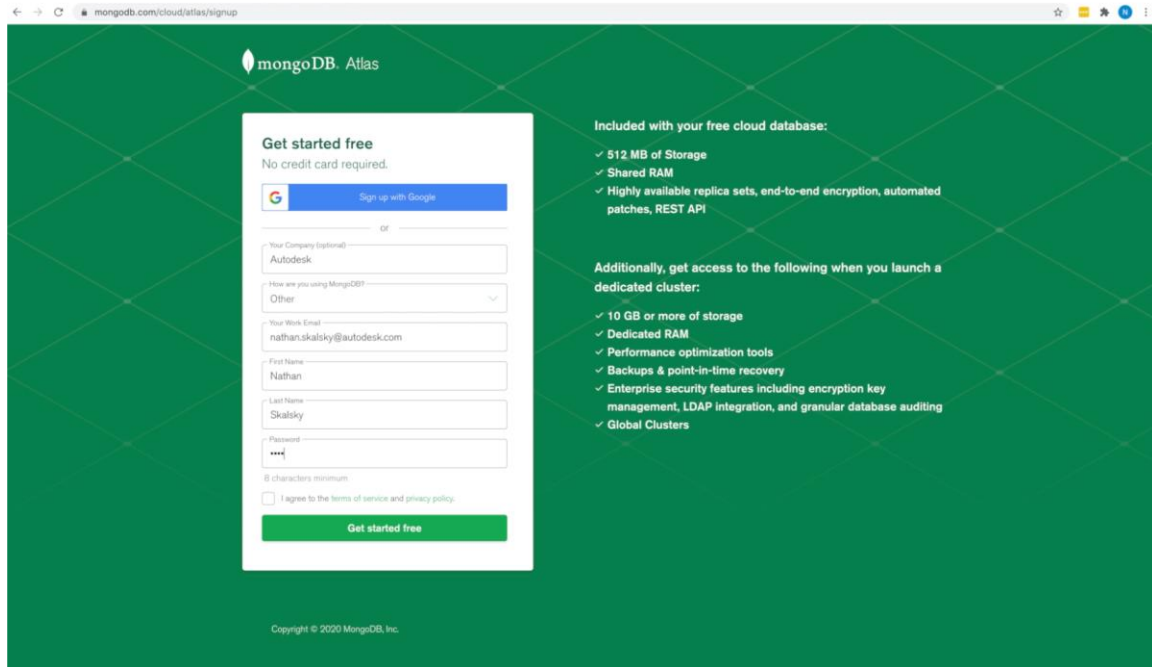
スピーカーについて

Varun Patil

Varun Patilは、フルスタック開発者であり、Forge開発チームの一員としてオートデスクで2年半以上勤務。

Forge APIを用いたサンプルプロジェクトの開発を通し、お客様の支援をしています。

データベースのデプロイ



mongodb.com/cloud/atlas/signup

Get started free
No credit card required.

Sign up with Google

OR

Your Company (optional)
Autodesk

How are you using MongoDB?
Other

Your Work Email
nathan.skalsky@autodesk.com

First Name
Nathan

Last Name
Skalsky

Password
•••••
8 characters minimum.

☐ I agree to the terms of service and privacy policy.

Get started free

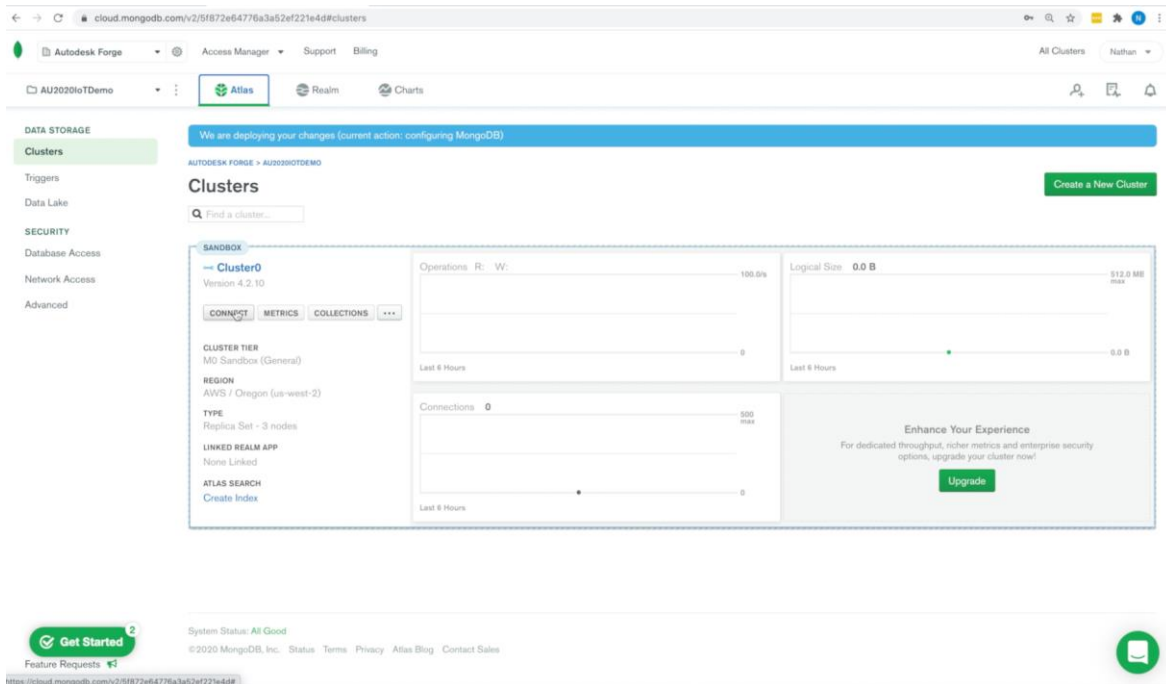
Included with your free cloud database:

- ✓ 512 MB of Storage
- ✓ Shared RAM
- ✓ Highly available replica sets, end-to-end encryption, automated patches, REST API

Additionally, get access to the following when you launch a dedicated cluster:

- ✓ 10 GB or more of storage
- ✓ Dedicated RAM
- ✓ Performance optimization tools
- ✓ Backups & point-in-time recovery
- ✓ Enterprise security features including encryption key management, LDAP integration, and granular database auditing
- ✓ Global Clusters

Copyright © 2020 MongoDB, Inc.



cloud.mongodb.com/v2/51872e64776a3a52ef221e4d#clusters

Autodesk Forge | Access Manager | Support | Billing

AU2020toDemo | Atlas | Realm | Charts

DATA STORAGE

- Clusters
- Triggers
- Data Lake

SECURITY

- Database Access
- Network Access
- Advanced

We are deploying your changes (current action: configuring MongoDB)

Clusters

Find a cluster...

SANDBOX

Cluster0
Version 4.2.10

CONNECT | METRICS | COLLECTIONS | ...

CLUSTER TIER
M0 Sandbox (General)

REGION
AWS / Oregon (us-west-2)

TYPE
Replica Set - 3 nodes

LINKED REALM APP
None Linked

ATLAS SEARCH
Create Index

Operations R: W: 198.0%

Logical Size 0.0 B 512.0 MB max

Connections 0 500 max

Enhance Your Experience
For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!

Upgrade

System Status: All Good
© 2020 MongoDB, Inc. | Status | Terms | Privacy | Atlas Blog | Contact Sales

Get Started | Feature Requests

https://cloud.mongodb.com/v2/51872e64776a3a52ef221e4d#

シンプルな MQTT ブローカのビルド (+MongoDB ブリッジ)

シンプルな MQTT ブローカのビルド (+MongoDB ブリッジ)

- 利用ライブラリ: mongodb, mosca (MQTT)
- 説明: ポート1888で MQTT メッセージを受信し、シンプルなセンサスキーマを用いて結果データベースに挿入

```

MQTTBroker.js
1 //MongoDB Connection
2 const MongoClient = require('mongodb').MongoClient
3 const dbConnectionString = MongoDB = "mongodb+srv://MQTTBroker:forme2no@cluster0.cnyhr.mongodb.net/
4 <dbname>?retryWrites=true&w=majority"
5 const client = new MongoClient(dbConnectionString, {useNewUrlParser: true, useUnifiedTopology: true})
6 client.connect(); // create persistent connection
7
8 //MQTT Broker
9 var mosca = require('mosca')
10 var config = {port: 1888}
11 var MQTTBroker = new mosca.Server(config)
12
13 MQTTBroker.on('ready', () => {
14   console.log('MQTT Broker Service Started!')
15 })
16
17 MQTTBroker.on('published', (packet) => {
18   message = packet.payload.toString()
19
20   if(!isNaN(message)){
21     console.log('Publishing Topic: ' + packet.topic.toString() + ' Value: ' + message)
22     var activeCollection = client.db('AU2020IoTDemo').collection('test_ESP12') //TODO: map collection
23     //name to cnc job name dynamically
24     activeCollection.insertOne({
25       when: new Date(),
26       temperature: message, // Sensor value, in this case (Temp), it will be degrees celsius (float)
27       sensor_type: packet.topic.toString().slice(0,4), // sensor type is only 'Temp' presently
28       sensor_name: packet.topic.toString().slice(5,25) // sensor name can be 'StockMaterial', 'Tool',
29       'spindle', 'StepperA', etc.
30     })
31   }
32 })
33 })

```

シンプルな MQTT ブローカーのテスト

- 利用ライブラリ: mongodb, mosca (MQTT)
- 説明:
 - MQTTPublish.js – topic/subtopic 文字列およびセンサーを元に、MQTTメッセージを送信
 - MQTTSubscribe.js – Will subscribe to receive 特定のtopic/subtopicを用いて、MQTTメッセージを受信するようにサブスクライブ

```

MQTTPublish.js
1 //MQTT client
2 var mqtt = require('mqtt')
3 var client = mqtt.connect('mqtt://localhost:1888')
4
5 //what topic?
6 var topic = 'Temp/Default'
7 var sensorValue = 0
8
9 client.on('connect', () => {
10   process.argv.forEach(function (val, index, array) {
11     if (index == 2) {
12       topic = val
13     }
14     if (index == 3) {
15       sensorValue = val
16     }
17   })
18   client.publish(topic, sensorValue.toString())
19   console.log('Sending Topic: ' + topic.toString() + ' Value: ' + sensorValue.toString())
20   return process.exit(22)
21 })

```

```

MQTTSubscribe.js
1 //MQTT client
2 var mqtt = require('mqtt')
3 var client = mqtt.connect('mqtt://localhost:1888')
4
5 //what topic?
6 var topic = process.argv.slice(2);
7
8 client.on('message', (topic, message) => {
9   message = message.toString()
10   console.log(message)
11 })
12
13 client.on('connect', () => {
14   client.subscribe(topic)
15 })

```

IoT デバイスのビルド

IoTデバイスのビルド

- 詳細はクラス動画とハンドアウトを参照
- BOM:
 - ESP8266 NodeMCU ESP-12E
 - DS18B20 デジタル温度計
 - 4.7k 抵抗
- 接続:
 - Temp Sensor.VCC = Vin on ESP board
 - Temp Sensor.GND = GND on ESP Board
 - Temp Sensor.Data = GPIO 5
 - 4.7k 抵抗は GPIO 5 および Vinに接続



IoT ファームウェアのビルド

IoT ファームウェアのビルド

- 詳細はクラス動画とハンドアウトを参照
- Nathanの4つの重要なIoTデバイスの機能
 1. アクセスポイントに接続するWiFiクライアント
 2. ユーザからのWiFi認証を受け付けるWiFiアクセスポイントとしての能力
 3. ユーザにコントローラのリセットを行うことを許可する機能 (2重リセット)
 4. MQTT クライアントおよびMQTTイベントへのサブスクライブ



```

Arduino File Edit Sketch Tools Help
CNC-IoT-TempSensor-ESP12 / Arduino
CNC-IoT-TempSensor-ESP12 $
}

void checkTempSensors() {
  // get the temps
  // update MQTT data with temp info + sensor type and name (MQTT topic/subtopic)
}

void updateMQTTData(float temp) {
  char mqtt_payload[100] = "";
  sprintf(mqtt_payload, "100, %f, %f", temp);
  Serial.print("MQTT Message: ");
  Serial.println(mqtt_payload);
  client.publish(MQTT_Topic, mqtt_payload, true);
  Serial.println("> MQTT data push");
}

void mqttCallback(char* topic, byte* payload, unsigned int length) {
  Serial.print("MQTT Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i=0; i < length; i++) {
    Serial.print(char(payload[i]));
  }
  Serial.println();
}

void connectMQTTclient() {
  while (!client.connected()) { //loop until connected
    Serial.print("***** Attempting MQTT connection...");
    if (client.connect(MQTTClientName, mqtt_username, "")) {
      Serial.println(" -> MQTT client connected");
    } else {
      Serial.print("failed, rc=");
    }
  }
}

Done compiling

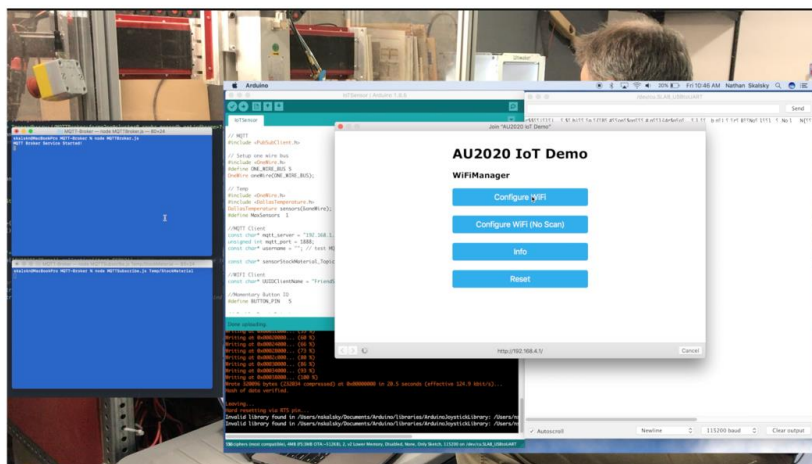
Sketch uses 311420 bytes (29%) of program storage space. Maximum is 1044464 bytes.
92
NodeMCU 1.0 (ESP-12F Module), 80 MHz, Flash, Legacy mode can return nullptr, All 50, 100MHz most compatible, 4MB PS

```

MQTT から database のユニットテスト

MQTTからdatabaseのユニットテスト

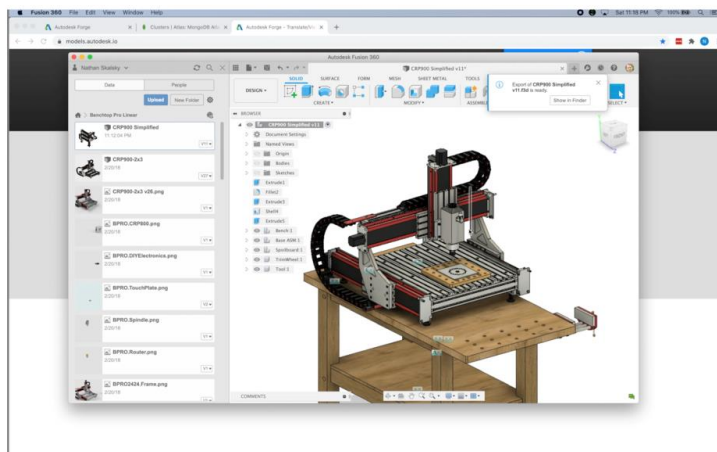
- 詳細はクラス動画とハンドアウトを参照



Fusion 360 モデルのアップロード

Upload Fusion 360 model to Forge

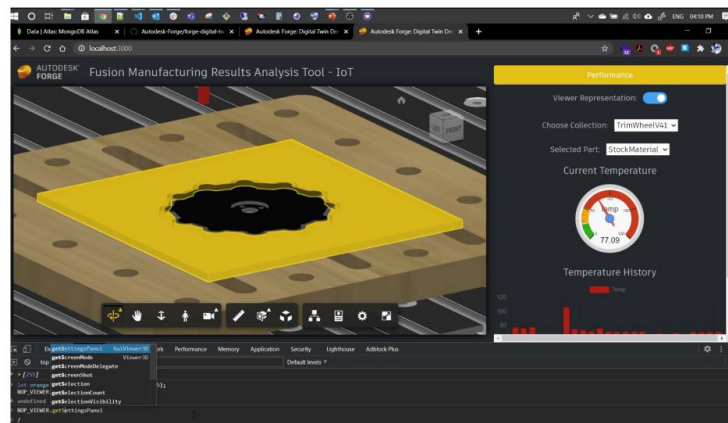
- ・ 詳細はクラス動画とハンドアウトを参照
- ・ Forge API認証を用いてモデルをmodels.autodesk.ioにアップロード



CNC データを表示するよう、Forge Viewer のコンフィギュレーションと拡張

可視化Webエクスペリエンスのビルド

- 詳細はクラス動画とハンドアウトを参照



まとめ

- データベースのデプロイ
- MQTT ブローカーのビルド
- IoTセンサーのビルド
- IoT ファームウェアのビルド
- MQTT → Databaseのテスト
- Fusion 360モデルのトランスレート
- Forge visualization サービスのビルド
- 結合
- まとめ



外部リソース・リファレンス

- サンプルプロジェクトのダウンロード > <https://github.com/Autodesk-Forge/forge-iot-cnc>
- Forge API ドキュメント > <https://forge.autodesk.com/developer/documentation>
- Forge Viewer > <https://forge.autodesk.com/en/docs/viewer/v7/overview/>
- MongoDB Atlas > <https://docs.atlas.mongodb.com/>
- NodeMCU ESP8266 > <https://www.instructables.com/NodeMCU-ESP8266-Details-and-Pinout/>
- MQTT > <https://mqtt.org/mqtt-specification/>
- デジタル温度計 (DS18B20) > <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>