# IoT機器における熱情報とCPU状態を用いた閾値アラートとプロセスの動的制御

多川 悠太[1,a)]　串田 高幸[1]

**概要**：近年，Alexa や Amazon Echo，AppleWatch を始めとする Internet of Things(以下 IoT と表記する) 関連機器が注目を集めてきている．一昔前まではなかった外出先から起動出来るエアコン，blutooth のイヤホン．人々の暮らしを豊かにする IoT．その IoT では，長時間利用における機器全体の発熱，CPU 本体の発熱や熱によるシステムダウンに関する問題があり，物理的な観点では空冷を行う方法が存在する．今回の論文では IoT 機器である RaspberryPi に着目し，RaspberryPi を使用している利用者に向けて本体のクロックダウンを気にすることなく長時間利用が可能となる提案を行う．

## 1. はじめに

今日，人々の生活を豊かにしている IoT 機器．IoT とはモノのインターネットのことで，田んぼの水位の測定，観測を自動化することやトイレの空き状態をスマートフォンで確認することが可能になっている．Krintz らは，プロセッサ (以下，単に CPU と呼ぶ) と周囲の大気温度との関係性を調べ，温度測定器として農業現場で利用することを目的としている [1]．温度測定器として RaspberryPi を利用し，リアルタイムの温度を測定し一週間のデータを取ったり、大気温度と RaspberryPi の CPU 温度を比較することで，局所的な水分の蒸発散量を予測することを可能にしている．

そんな中，学生にとって身近な勉強しやすい IoT 機器はシングルボードコンピュータでもある RaspberryPi である．RaspberryPi とは温度センサーや体感センサー，ビーコンセンサーを使って測定，観測を行い，データを分析する IoT 機器の一種である．だが RaspberryPi は測定や観測を行うという性質上，一週間や一ヶ月，長時間稼働し続ける必要があることが多い．当然稼働している間は CPU の熱が発生し，埃や水から RaspberryPi を守るために保護ケースに入っている場合が多いため，長時間の稼働によりその熱が籠ってしまう．そこで導入されている機能としてクロックダウンという機能が RaspberryPi には存在しているのだが，RaspberryPi にはクロックダウンの機能があるバージョンとクロックダウンの機能が無いバージョンがある．そのため，機能が無いバージョンを利用している利用者の中には急に RaspberryPi の電源が落ちてしまった，強制終了になった後に起動できなくなってしまったという報告が存在している．また，クロックダウンが機能として存在している RaspberryPi はどの様な基準でクロックダウンが行われているのか，またクロックダウンが解除されるのはいつなのかという問題も存在している．この様なことが度々起こるようでは RaspberryPi は状態を確認しに付きっきりで利用者が監視する必要がある．今回のレポートでは，その様なことがないように熱に関する項目を活用し監視者がその場にいなくとも RaspberryPi を使用することが出来るようアラート制御システムを提案する．

本論文では，２章で IoT 機器についての既存技術について紹介を行い，３章以降で今回の熱情報と CPU 状態を用いた閾値アラートとプロセスの動的制御を行うシステムについて報告する．

### 1.1 背景

前述したとおり RaspberryPi がクロックダウンしてしまい，原因以外のプロセスの活動も止まってしまう．それでは作業全体がストップしてしまうことになってしまうため今回の提案を行う．

また，今回の提案の環境は RaspberryPi3B+，研究室のサーバ上の VMware で構成されている．

### 1.2 課題

今回のレポートでは，IoT 機器を長時間使用している際，CPU から発せられる熱によるシステム全体のダウンで作業全体が中断されてしまうことを課題としている．作業全

[1] 東京工科大学コンピュータサイエンス学部
〒 192–0982 東京都八王子市片倉町 1404-1
[a)] C0117191

体が中断されてしまうと，放熱され，システムが復旧できるレベルになるまでダウンする前までにやっていた作業が出来なくなってしまう．そのため，このレポートでは熱に関したシステムダウンに関する提案を行う．

## 2. 関連研究

課題に関連する研究を記述する．RaspberryPi の温度についての対策や，負荷によって引き起こさプロセスに対する研究が存在している．

Brian らは RaspberryPi の CPU チップが 85 %の過熱状態に達成した時にクロックダウンしてしまうがオーバークロックを使用しない通常の設定であれば発熱を心配する必要がほぼない可能性を挙げている [2]．実験では，オーバークロック状況下で RaspberryPi を使用するという前提条件で RaspberryPi のケースがある場合と無い場合の比較を二分に一回温度センサーで計測することにより行っている．ケースなしでの過熱試験では，アイドル状態での平均温度は 45.5 度で稼働後 55 度に上がったことを挙げている．ケースありの実験もアイドル状態から調べていて，56 度であった．ケースを用いた状態での過熱試験では，最高温度が 68 度，その他は 66 68 度の間で推移していた．この様にケース付きで稼働させている RaspberryPi の温度上昇値はベンチマークのみの状態でも高く，この論文の実験気温である 23 度を上回る環境下であれば非常に高くなってしまうことが懸念される．

Zhengguo Sheng，Chinmaya Mahapatra らは産業用の無線センサーネットワーク，IoT について効率的な管理について述べている [3]．小型のワイヤレスセンサデバイスはバッテリー電力，処理能力と貯蔵能力，無線接続範囲と信頼性が制限されるという特徴があり，パフォーマンスの監視やセンサーノードへのコマンドの送信を人の手を介さずに時間の遅れがないようにリアルタイムでリモート管理するための通信のプロトコルを設計しなければならないことを挙げている．

また文中では IoT を使用した環境を指すスマートシティ，スマートホームについて述べられている．スマートホームは家電製品を始めとし，家の中に存在するデバイスが接続し合うことで，利便性の高い生活を提供する環境のことで，スマートシティは IoT を活用して，基礎インフラと生活インフラ，サービスを効率的に管理して経済発展目指す都市である．

IoT システムに存在している幅広いインテリジェント (データ処理を持つ機器) で小型のセンシングデバイスが存在しているが，すべて共通のアーキテクチャとネットワーク要素で共有しており，一般的な特徴と課題として 6 つに要約している．

(1) リモートデバイスと通信監理を介して信頼出来る通信を確保する必要，つまりセキュリティ面での問題がある．
(2) 現在の互いに異なるプラットフォームを使うのではなく，単一のプラットフォームとオープンプラットフォームのアプリケーションプログラミングインターフェース (以後 API と記す) でサービス機能を統合することでカスタマイズ性を上げることが出来るため重要度が高い．
(3) IoT ネットワークへの大規模な接続ためにアドレスリソース不足やアクセスの輻輳などの問題に対処する必要がある．
(4) 通信速度で遅延や再送が発生してしまい余計な負荷がかからないよう，QoS 要件を保証するために，異種アクセスの命名とアドレス指定に対処する必要がある．
(5) IoT サービスを実現できるようにするために，異種アクセスのネーミングやアドレッシングに対応する必要がある．
(6) 大規模な情報の蓄積，共有，マイニングに対応しなければならない．

そのため，今回は (6) に対しての対策としてサーバを利用することを提案で後述する．

Chandra，Krintz らは，この論文では IoT を農業に活かしている IoT を農業に活用するメリットとして，日照り対策，防霜，凍結対策を挙げ，例えばアメリカでは，霜害による損失は他のどの気象関連の現象よりも大きいとしている [1]．また大規模な防霜の対策には水の散布，風力発電機の使用，ヒーターの組み合わせから選ばれるか，その全てが必要になってくる．

従来のこの方法であるとかなりの手間がかかり，霜害が発生するタイミングを誤ってしまうと対策分費用が増え生産者にとってはコストパフォーマンスが悪くなってしまう．その上， どの作業においてもリアルタイムで正確に温度を測定し，予測することが求められ農業の土地環境の差異の要素である，地形の違い，周囲の構造物，地表面の覆い，植物の成熟度，近隣の水域が挙げられる。そして，膨大な数のパターンが存在する微気候を踏まえて的確に温度を測定することは，高コストで手間のかかることである．そこで IoT ベンダーが提供しているサービスを利用することで，データ抽出や高度な分析が可能となる．しかし，これらのサービスは高価であり，データをクラウドに送信する必要と定期的な分析のための定期料金を課しており，その結果として IoT の農業への導入は進んでいない．そこで，この論文では，オープンソースのクラウドウェアを使用しエッジクラウドを設計することで自己管理を可能としている．また微気候の温度を正確にリアルタイムに推定するために比較的安価である RaspberryPi を使用し予測を均一ではなく，局所的な温度の違いをリアルタイムで正確に低コストで予測することが出来ると主張している．

Tetsuya Oda らは，ノードの CPU 周波数と CPU 温

度に関する実験をしている [4]．実験では，CPU 周波数 100MHz，700MHz の二つについて比較していて，平均アイドル時間が 58.120 %，88.820 %で，CPU 温度の平均値は 49.107 度，49.487 度という結果になっている．平均スループットに関してもベンチマークを使用することで検証していて，CPU 周波数 100MHz，700Mhz の条件下で 505.8Kbps，511.2Kbps とどちらの実験，検証でも動作周波数 (またはクロック周波数) が高い方が CPU 温度上がりやすくなり，良い処理結果が得られることが分かっている．ただし，Raspberrypi の場合，動作周波数が高くなり過ぎると比例して CPU 温度も上昇し強制的にクロックダウンしてしまう．そのため，処理が重くなっているプロセスを突き止め，プロセスの停止と再会の制御を行う必要がある．

## 3. 提案

このレポートでは機器が壊れてしまう原因である熱に関して注目する．題では IoT 機器としているが今回は RaspberryPi として考える．理由として RaspberryPi には CPU が熱くなり過ぎると機器を守るために CPU の性能・稼働率を下げるクロックダウンが行われる．しかし，クロックダウンしてしまうと稼働中であるシステム，プロセス全体が通常以下の処理速度になってしまう．そのため，今回の提案では RaspberryPi に熱がこもりすぎてしまった場合に原因となっているだろうシステム，プロセスを特定し，それを一時的に停止させ，温度が正常に戻ったら一時的な停止をやめ，再開させる．追加的な機能として，RaspberryPi の CPU 自体の温度と周囲の温度から閾値を取り，温度が高くなっている旨のアラートを発生させる．また，閾値を越えていない時もサーバに状態の情報を送りいつでも確認が可能なように RaspberryPi からサーバに向けてデータを送信する．その具体的な図として挙げられるのが図 1 である．また図の矢印に振られている番号は実際にソフトウェアが稼働した時の実行手順であるため，以下に示し，図に存在するソフトウェアについても述べる．

(1) 閾値を超えた時の IoT 機器 (Raspberrypi) の熱情報，CPU 使用率等の負荷情報を送信
(2) 閾値を越えた時に限らず全ての IoT 機器の熱情報，CPU 使用率等の負荷情報を送信
(3) 一定の間隔でサーバに保存されている負荷情報をいつでも確認出来る．

まず，IoT 機器である RaspberryPi のソフトウェアであり熱や CPU の情報 (以下，二つをまとめて負荷情報と表記する) を使用者に通知する機能を持つ Load Notification(以下，LON と表記する) と負荷情報の受け渡し機能を持つ Passing load information(以下，PLO と表記) についてである．

先に挙げた LON は，図 1 で (1) の機能を担っていて，閾値を超えた熱や CPU 熱の負荷があった際に，それぞれ温

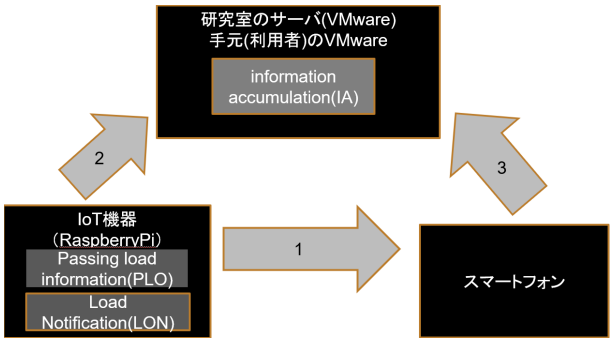


図 1 提案するソフトウェアの構成図

度が何度であったのか，CPU の使用率が何%あるのかという情報を送信する．次の PLO は (2) の役割を担うことに加え，LON に渡す負荷情報を一定の間隔で取得する．また閾値以下の情報を機能は LON が行う．

次にサーバ (または VM) 上に情報を蓄積する information accumulation(以下，IA と表記) についてだが，これは一定の間隔で情報を送信する PLO から受け取った情報を蓄積するためのもので，RaspberryPi の MicroSD に蓄積する方法とは異なり，情報を蓄積することが出来る．そのため，利用者のスマートフォンからサーバへアクセスすると slack に通知が来た前後の状況を確認することが可能となっている．図 1 の 3 でスマートフォンからアクセスがあった際に機能するソフトウェアである．

## 4. 実装と評価

### 4.1 実装

今回の論文では，提案する前の段階であるが実装の下準備として取り掛かれた部分を挙げていく．

まず、閾値を超えた際に通知する slack について取り組み，その結果として，RaspberryPi でプログラムをコンパイルした際に特定のメッセージを送信することが可能になるようにした．

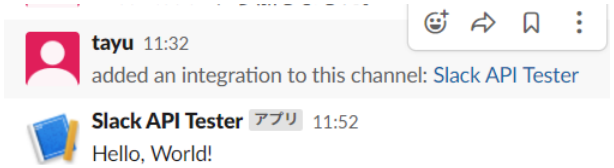


図 2 RaspberryPi でコンパイルした際に slack に文字が表示されるプログラム

### 4.2 実験環境

今回は実装出来ていないが，実装を行おうとして考えていた環境について述べていく．IoT 機器には RaspberryPi3B+を使用している．この RaspberryPi3B+には自身の CPU 温度を記録するプログラムを組む．記録を送信するサーバは，研究室に存在しているサーバを利用し，記録

を保持するデータベースを構築する．このデータベースはスマートフォンからもアクセス出来，二日間の期間の記録を保持する．また，RaspberryPi からスマートフォンに slack，Gmail を使用しアラートを送信するプログラムも組み込む．

### 4.3 評価

実装で行ったプログラムが図 3 である．実際には，このプログラムを応用し，閾値を超えた時だけ温度を送信するプログラムを作成する．

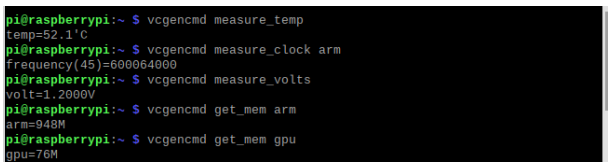


図 3 RaspberryPi の温度，CPU 情報を取得するプログラム

## 5. 議論

ここでは今後実装する必要があるソフトウェアについて図 1 を見ながらそれぞれ議論する．まず，IoT 機器である RaspberryPi のソフトウェアであり熱や CPU の負荷情報を使用者に通知する機能を持つ LON と負荷情報の受け渡し機能を持つ PLO についてである．

先に挙げた LON では，図 2 を発展させ，閾値を超えた熱や CPU 熱の負荷があった際に温度が何度であったのか，CPU の使用率が何％あるのかといった情報を送信出来るようにしていく必要がある．次の PLO では LON に渡す負荷情報を一定の間隔で自動取得するようにする必要がある．また閾値以下のものを弾く機能は LON が行う．

次にサーバ (または VMware) 上に情報を蓄積する IA についてだが，これは一定の間隔で情報を送信する PLO から受け取った情報を蓄積するためのもので，RaspberryPi の MicroSD に蓄積する方法とは異なり，情報を蓄積することが出来ると考えられるが，検証が必要である．利用者のスマートフォンからサーバへアクセスすると slack に通知が来た前後の状況を確認することが可能となっている．しかし，蓄積するという機能自体も未だ実装出来ていないため，RaspberryPi からの情報を受け取れるように実装を行う必要がある．

今回のレポートでは，実装，評価をあまり行うことが出来なかった．しかし，世の中でコロナウイルスが蔓延して外出出来ないという状況下に置かれ分かったこととして，オンライン講義を始め，オンライン説明会，オンライン面接，オンライン会議等，様々なことに IT の技術が必要とされているためスマートシティやスマートホームといったものに利用されている IoT の研究は必要なことであると考える．

## 6. おわりに

今回の提案では IoT が抱える熱という問題に対する課題についての一つの解決案を提示した．次回のレポートでは今回の続きである実装や評価を行い新たな課題、問題を浮き彫りにして解決をしていく．

### 参考文献

[1] Krintz, C., Wolski, R., Golubovic, N. and Bakir, F.: Estimating Outdoor Temperature from CPU Temperature for IoT Applications in Agriculture (2018).
[2] Dye, B.: Distributed computing with the Raspberry Pi, *K-State Electronic Theses* (2014).
[3] Zhengguo Sheng, C. M.: Recent Advances in Industrial Wireless Sensor Networks Toward Efficient Management in IoT, *IEEE* (2015).
[4] Tetsuya Oda, L. B.: Experimental Results of a Raspberry Pi Based WMN Testbed Considering CPU Frequency, *IEEE* (2016).