

VG-Hub 制御・管理のクラウド化に関する検討

大條海渡¹ 川喜田佑介² 田谷昭仁¹ 戸辺義人¹ 横川慎二³ 市川晴久³

概要：我々は USB Power Delivery (USB PD) 規格に対応した USB Type-C のポートを搭載し、直流電力の合成と分配を行うバーチャルグリッドハブ(VG-Hub)の構築を行ってきた。本研究は VG-Hub の制御を行う制御アルゴリズムをクラウドに置くことで複数台の VG-Hub の管理を容易にすることと協調動作を行うことを目的としている。本稿では、AWS (Amazon Web Services)を用いた VG-Hub 制御クラウドのプロトタイプについて述べる。

1. はじめに

近年、家庭での利用も拡大しつつある太陽光発電は、曇りや雨の日には発電量が減少してしまうという問題点がある。このような太陽光発電の天候による発電量の変動を吸収するためには、余剰電力がある際にその電力を蓄電池に貯めておき、発電量が減少した際には蓄電池からの給電を行うような仕組みが必要となる。しかし現在の送電網は交流の電力網となっており、ノート PC やスマートフォンといった直流駆動の端末に対して給電を行う際、直流から交流に変換し、再び直流に変換する必要がある、交流と直流を変換する際には変換ロスが生じてしまうため非効率である。そのため、直流のマイクログリッドの研究開発も進められている¹⁾。

我々は、直流マイクログリッドの一形態として、USB PD 規格に対応した USB Type-C を用いて事業所内・家庭内で利用可能なバーチャルグリッドシステムを開発している²⁾。このバーチャルグリッドシステムでは、VG-Hub と呼ばれるハブを用いて直流電力の合成・分配する。

実用的な規模で運用するには、相互接続された複数の VG-Hub が必要となり、現状では各 VG-Hub 上で管理されており、複数台の VG-Hub を導入した際の管理が複雑になってしまう。そこで本研究は、制御アルゴリズムをクラウドに置くことで、VG-Hub の管理・制御のクラウド化を行い、複数台の VG-Hub がある際の管理、データ収集、解析を容易にし、協調動作を行うことを目的とする。本稿では AWS (Amazon Web Services) IoT と AWS Lambda を用いたプロトタイプについて述べる。

2. 関連研究

マイクログリッド¹⁾は1つの住宅街や都市を導入の対象としていたのに対し、我々の先行研究で提案したバーチャルグリッドシステム²⁾は、住宅などより狭い範囲での導入を想定しており、ユーザが自由に VG-Hub を追加・移動し、グリッドを構築することができる。さらに、我々は VG-Hub に内蔵された Raspberry Pi や DC/DC 変換器で発生する電力損失を定量化し、損失関数を定義した³⁾。またその損失関数を基に、負荷デバイスが接続されているポートの最大電

力値を計算し、あるポートで余剰電力が発生したらその余剰電力を他のポートに割り当てるといった制御を行った。しかし、この制御アルゴリズム自体は VG-Hub 上で動作し、本研究で行うような制御のクラウド化は行われていない。

3. クラウド型直流電力分配システム

VG-Hub の制御のクラウド化に当たっては、アマゾン社の AWS を用いた。VG-Hub と AWS を接続するために、AWS のサービスの1つである AWS IoT を用いた。また、制御アルゴリズムは同じく AWS のサービスの1つである AWS Lambda で記述した。

システムの全体像を図1に示す。VG-Hub 内部では、各ポートに Source と Sink の役割や最大電力値を割り当て、各ポートの情報を取得するポート制御プロセスと、AWS との橋渡しを担う通信プロセスが動く。制御の流れとしては、まず各ポートの接続状況、役割や最大電力値といった情報をソケット通信で通信プロセスに渡す。AWS IoT Core との通信は MQTT (Message Queueing Telemetry Transport) 通信で行う。AWS IoT Core 内のデバイスシャドウには、端末からのステータスを格納する“reported”と、アプリケーションからの要求を格納する“desired”の2つのメモリがあり、まずは VG-Hub がデバイスの初期状態をシャドウ更新用のトピック (\$aws/things/VGHub/shadow/update) にパブリッシュし、“reported”に記録する。また、AWS Lambda へ送信す

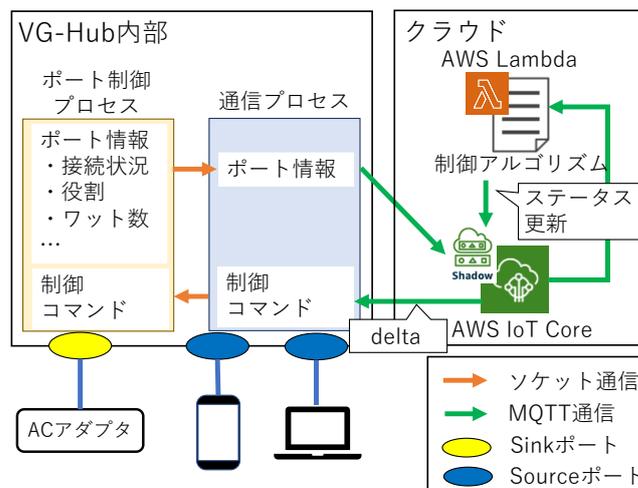


図1 システムの全体像

1 青山学院大学理工学部情報テクノロジー学科
 2 神奈川工科大学情報学部情報工学科
 3 電気通信大学i-パワードエネルギー・システム研究センター

る目的で別途用意したトピック (VGHub/info) にポート情報をパブリッシュする。AWS IoT Core 内でメッセージを受け取ると、IoT ルールによって AWS Lambda が発火し、受け取ったデータを基に新たな最大電力値を計算してシャドウ更新用のトピックにパブリッシュする。この際“desired”へ記録を行う。デバイスシャドウは“reported”と“desired”の差分をトピック (\$aws/things/VGHub/shadow/update/delta) にパブリッシュする。VG-Hub の通信プロセスは、このトピックをサブスクライブしており、その差分を埋めるように制御コマンドを生成し、ポート制御プロセスで制御を行う。

4. 実験

本章では、システムの実装とその結果について述べる。

4.1 実験

現在、複数台の VG-Hub がある際、相互に同期しておらず、各 VG-Hub が独立して電源からの直流電力を合成し、接続された負荷デバイスに分配を行っている。しかし、1 台の VG-Hub に接続することが出来る負荷デバイスには限りがあり、将来性を考慮すると複数台の VG-Hub を接続することが必要になると考えられる。本研究は、VG-Hub の制御をクラウドから行うアプローチをとった際に、制御にどれほどの時間を要するのかを計測することを目的としており、今回は AWS 上でクラウドから制御を行う制御システムを構築した。

AWS 上で動かす制御アルゴリズムとして、先行研究にて提案された損失関数から、VG-Hub に接続された負荷デバイスの最大電力値を計算し、更新を行うというものを AWS Lambda に実装した。今回は、VG-Hub の 3 番ポートと 4 番ポートに負荷デバイスとしてモバイルバッテリーを接続し、初期の最大電力値として 20W を設定した。

またこのシステム上で、VG-Hub の 3 番ポートにモバ

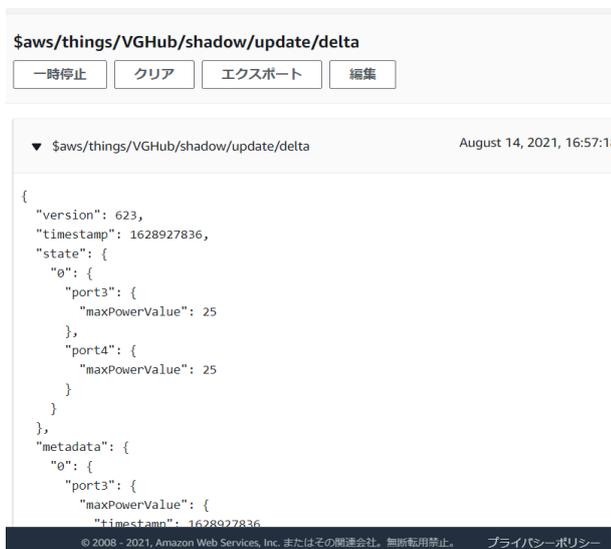


図 2 デバイスシャドウの“delta”の内容

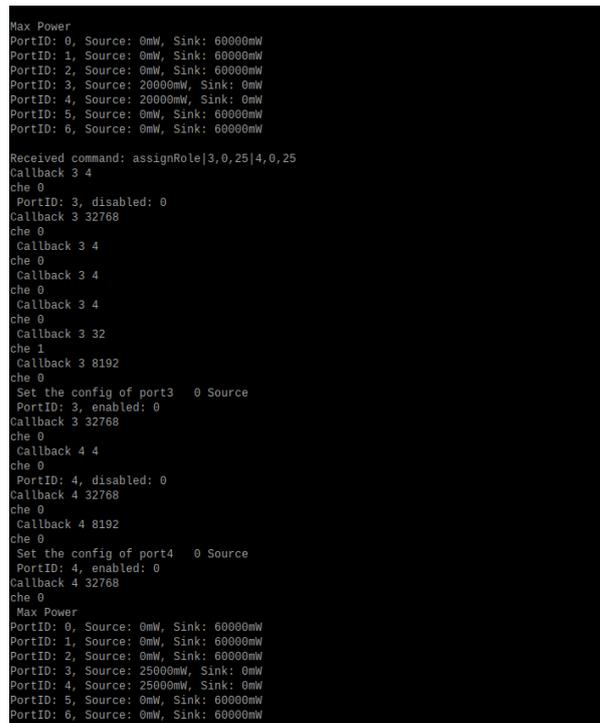


図 3 3 番ポートと 4 番ポートの最大電力値が更新されている様子

イルバッテリーが接続されている状態で、通信プロセスからデバイスの初期状態をシャドウ更新用のトピックにパブリッシュし、AWS Lambda 上で処理を行った後にデバイスシャドウの“reported”と“desired”の差分をサブスクライブするまでにかかった時間を Wireshark にて計測を行なった。

4.2 結果

AWS Lambda 上で 3 番ポートと 4 番ポートの新しい最大電力値が 25W と計算され、デバイスシャドウの“desired”の項目に設定された。ここで“reported”には 20W と設定されているため、“reported”と“desired”の間に差分が発生し、“delta”に差分が発生している項目が設定された。これは図 2 に示す。VG-Hub 内部の通信プロセスでは、“delta”の内容がプッシュされるトピックを購読しており、“delta”の内容が MQTT 通信で通信プロセスに送信される。通信プロセスでは、受け取った“delta”から 3 番ポートには 25W、

表 1 Wireshark での測定結果

計測回数[回]	開始時間[s]	終了時間[s]	差分[s]
1	20.69	21.95	1.254
2	0.000	1.255	1.255
3	0.000	1.256	1.256
4	0.000	1.273	1.273
5	22.96	24.22	1.264
6	25.34	26.65	1.314
7	0.000	1.208	1.208
8	0.000	1.268	1.268
9	0.000	1.307	1.307
10	13.04	14.28	1.234
平均			1.263

4番ポートには25Wの最大電力値を設定するという情報を取り出し、今度はソケット通信でポート制御プロセスにその情報を送信する。ポート制御プロセスでは、受け取ったポート番号と最大電力値を基に、3番ポートと4番ポートに25Wの最大電力値を割り当てる。3番ポートと4番ポートの最大電力値が20Wから25Wに更新されている様子を図3に示す。

また通信プロセスからデバイスの初期状態をパブリッシュし、デバイスシャドウの差分をサブスクライブするまでに要した時間をWiresharkにて計測を行なった。この計測結果を表1に示す。測定結果より差分の平均は1.263秒であり、これはインターネットアクセスなどの諸オーバーヘッドを考慮すると十分に高速であるといえる。

5. むすび

本稿では、複数のVG-Hubの制御・管理をAWS上で行うシステムの試作結果について述べた。現状では、各VG-Hubの値の取得・設定に留まっており、今後、複数VG-Hub間の協調動作を実装していく予定である。

参考文献

- 1) 一井啓介, 高原勇, 谷口守: 再エネ活用に向けたスマートグリッド直流化効果の検証. 第59回土木計画学研究発表会・講演集(2018).
- 2) Ichikawa, H., Yokogawa, S., Kawakita, Y., Sawada, K., Sogabe, T., Minegishi, A., and Uehara, H.: An Approach to Renewable-Energy Dominant Grids via Distributed Electrical Energy Platform for IoT Systems. IEEE Int. Conf. on Communications, Control, and Computing Technologies for Smart Grids (2019).
- 3) 高家和暉, 田村光汰, 戸辺義人, 川喜田佑介, 横川慎二, 市川晴久: USB-PDハブによる電力配信の効率化に向けた制御検討. 令和3年電気学会全国大会, (2021.3).