

機械学習を利用したパケットキャプチャによる消費電力推定手法

中山 心太†, 上西 康太†, 三浦 愛美‡, 小田 哲†

公立はこだて未来大学大学院 システム情報科学研究科

† NTT 情報流通プラットフォーム研究所

† {nakayama.shinta, uenishi.kota, oda.satoshi}@lab.ntt.co.jp

‡ 公立はこだて未来大学

‡ g2111036@fun.ac.jp

Estimation of electric energy consumption using machine learning and packet capture.

Shinta Nakayama†, Kota Uenishi†, Manami Miura‡, Satoshi Oda†

† NTT Information Sharing Platform Laboratories

† {nakayama.shinta, uenishi.kota, oda.satoshi}@lab.ntt.co.jp

‡ Future University Hakodate

‡ g2111036@fun.ac.jp

概要 近年、半導体集積回路の電源管理技術の進歩により、IT 機器の動作時とアイドル時の消費電力の差が拡大している。そのため、同一のコンピュータであっても実行されているプログラムによって消費電力に大きな差が生じるようになった。また、データセンター事業者は、空調制御やフロアプランを行うためにコンピュータの消費電力を把握する必要がある。コンピュータの消費電力の把握には電力計や CPU の利用率を利用した方法がある。しかし、遠隔監視可能な電力計は高価であり、すべてのコンピュータに設置することは現実的ではない。また、CPU の利用率はモニタリングソフトウェアを導入する必要があるため、データセンター事業者が顧客環境に導入することは難しい。そこで通信パケットから消費電力を推定する手法を提案する。この手法は、まず一部のコンピュータの消費電力と通信パケットを取得し、機械学習を行い消費電力と通信パケットの関係性のモデルを構築する。次に、構築されたモデルを利用して他のコンピュータの通信パケットから消費電力の推定を行う。これにより低コストで顧客環境に影響を与えずに、リアルタイムにコンピュータの消費電力を推定することができる。本研究では提案手法の有効性を評価するために評価実験を行い、通信パケットと消費電力の間に正の相関があることを確認した。

キーワード 電力推定, 機械学習, パケットキャプチャ

1 背景

東日本大震災以来、あらゆる電力消費に対して厳しい目が向けられており、クラウドもその例外ではない。この「クラウド」という言葉は、幅広い用途で使われているが、そのクラウド環境を実現する実体はデータセンターであり、多くの場合サーバ群である。これらのサーバ群はデータセンター全体として多くの電力を消費するため、DEMS (Data center energy management system) に代表されるようにデータセンターを運営する立場にとって、データセンターの消費電力を知ることは重要である。

データセンターにおいて、その電力消費量は大きく分けて、IT 機器を運用するための電力、空調など IT 機器を冷やすための電力、設備を維持するための電力（照明など）の三つに分類される。これらの消費電力を減らし、いかに効率のよいデータセンターを作るかについて、DCiE (Data Center infrastructure Efficiency), PUE (Power Usage Effectiveness), DCP (Data Center Productivity) などいくつかの指標が提案されている。特に、クラウド事業者にとってより少ない消費電力で多くのクラウドサービスを提供することが望ましい状態である。

一方、半導体集積回路の電源管理技術の進歩によ

り、IT 機器の動作時とアイドル時の電力量の差が拡大している。たとえば Intel 社の Core i7 では、負荷にあわせて利用するコアの数や、動作周波数、動作電圧を動的に変更することで消費電力の削減を行っている。そのため単純に IT 機器の個数に定格消費電力を掛けた値では、実際の消費電力から乖離してしまう。したがって、データセンターの空調制御やフロアプランなど活かすには、IT 機器ごとの消費電力をリアルタイムに知ることが重要である。

IT 機器ごとに消費電力を計測するためには、一般的にクランプメータが利用される。クランプメータは電源ケーブルに取り付けて実際に流れている電流から発生する磁束を観測する。データセンターにおいてクランプメータを用いる場合は、ただ観測するだけでなく、観測した結果を別システムのネットワークを用いて収集し、一覧で管理できるように運用するのが一般的である。そのため、すべての IT 機器に対してクランプメータを設置し、センサーネットワークを構築するためには多くの投資が必要となる。

また、低コストで利用している IT 機器の現在の消費電力を簡易的に知るには、CPU 利用率を観測しそこから消費電力を推定する方法が考えられる。利用している端末、サーバの型番が分かり、CPU の利用率が分かれば、ある程

度正確な消費電力量が得られる。しかし、CPU利用率は、OS上でソフトウェアを実行しなければ得ることができない。そのため、自社設備であれば容易に導入は可能だが、クラウドのように、消費電力を知りたい事業者と中で動いているソフトウェア（OS）を管理している事業者が異なる場合では導入が難しい。そのため、低コストで顧客環境に影響を与えない電力計測手法が求められている。

2 既存手法

データセンタにおける電力計測の既存手法は、電力計を使った方法、CPU利用率の計測、パケット流量の計測が存在する。

2.1 電力計による計測

電力計を利用することでサーバの消費電力を直接計測することができる。たとえばHP社のHPインテリジェントパワーディストリビューションユニット（iPDU）[1]は、電源タップにRJ-45のポートが付いており、有線LAN経由で電源ポートごとの消費電力を監視することができる。しかし、このような監視可能な電力計付きのタップは高価であり、消費電力収集用のネットワークを敷設しなくてはならないため、すべてのサーバの消費電力を計測するためには費用がかかる。

2.2 CPU利用率の監視

CPUの消費電力は省電力機能により時々刻々と変化し、またCPUはサーバの消費電力の変動部分の大半を占める。そのため、CPUの利用率が計測できれば、サーバの消費電力が予測可能である。

Fanらの研究[2]では、消費電力を計測しながら任意のCPU負荷をかけられるベンチマークプログラムを実行し、消費電力を次の式で近似している。

$$P_{idle} + (P_{busy} - P_{idle})(2u - u^r)$$

P_{idle} は無負荷の際の消費電力、 P_{busy} はCPUリソースをすべて使っている際の消費電力、 u はCPU利用率、 r は係数である。Fanらの実験では $r=1.4$ となっている。

Fanらのように自社設備¹であれば、CPU利用率の監視ソフトウェアを導入することは容易である。しかし、クラウド事業者のように、OSレイヤより上が顧客環境であった場合、顧客にCPU利用率の監視ソフトウェアの導入を求めることは難しい。

2.2.1 パケットの流量

静的コンテンツをホスティングしているWebサーバや、ファイルサーバ、ルータ、スイッチ[3]のような単純な機能のサーバであればパケットの流量と処理の量は比例する。したがって、パケットの流量とCPU利用率の比較表があれば、前述のCPU利用率の監視と同等の方法で消費電力を推論することができる。

この手法は、顧客環境にソフトウェアを導入しなくとも、ルータやハブからポートミラーを行い、サーバに流入するパケットの量を監視することで電力を推定することができる。

しかし、動的なコンテンツをホスティングしている

サーバやリレーショナルデータベースサーバなどでは、クエリと処理の量は一致しない。負荷が少ないクエリもあれば、1つのクエリで長時間計算を行うものまで存在する。そのため、複雑なシステムが動いているサーバにおいて、パケットの流量から消費電力を求めることは難しい。

3 提案

サーバの消費電力はCPUの利用率に強く依存する。そしてCPUの利用率はOS上でどのアプリケーションがどのように動作しているかに強く依存する。そのため、前述のように、パケットの流量から消費電力を直接推定することは難しい。

しかし、パケットのポート番号やペイロードまで確認することで、サーバ上で動いているアプリケーションの動作を推定することができ、消費電力を推測できる可能性がある。

たとえばオープンソースの侵入検知/防止システム（IDS/IPS）のsnort[6]では、パケットのポート番号やペイロードを解析することで、サーバ上で動作している数百のアプリケーションを分類可能である。しかしsnortを利用して消費電力を推測する場合、それぞれのアプリケーションごとにパケットの流量と消費電力の関係性を測定する必要がある。

そこで、本稿では機械学習によりパケットから消費電力を推測する手法を提案する。提案手法の構成を図1に示す。提案手法では少数のサーバで消費電力の測定を行い、ルータやスイッチからポートミラーによりパケットを収集する。そして、消費電力とパケットの組を教師データとして機械学習を行い、消費電力とパケットの関係性のモデルを構築する。そして構築されたモデルにより、他のサーバの消費電力を推定する。

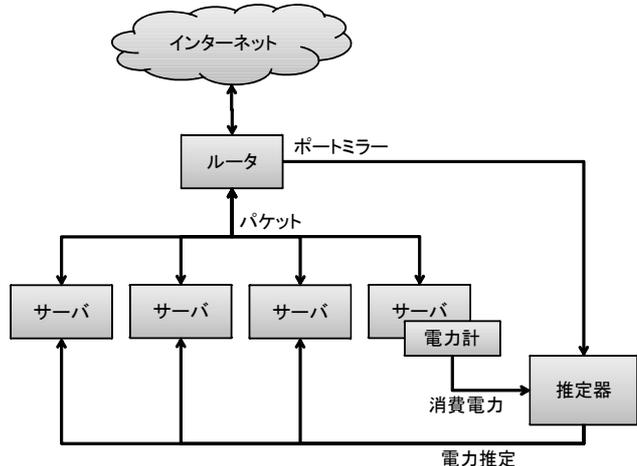


図1:提案手法の構成

本提案手法と既存手法との比較を表1に示す。提案手法は機械学習による推論を行うため、正確性を犠牲にしている。提案手法は電力計と推定用のハードウェアを必要とするものの、電力計をデータセンタ全体に配置するよりも低コストで実現できる。また、提案手法は電力計とルータのポートミラーを利用しているため、顧客環境に影響を与えずに消費電力を計測することができる。提案手法は、同等環境のサーバが多数並んでいる場合、設

1 FanらはGoogle社の設備で検証

置する電力計の数が少なくてよく、正確性、コストの面で効果的に働くと考えられる。

表 1:既存手法との比較

| | 電力計 | CPU 利用率 | パケット流量 | 提案手法 |
|----------|-----|---------|--------|------|
| 正確性 | ○ | △ | × | △ |
| コスト | × | ○ | ○ | △ |
| 顧客環境への影響 | ○ | × | ○ | ○ |

4 実装

提案手法の検証のために、一台環境で実験を行った。

4.1 実験環境

検証実験の構成を表 2 に示す。

表 2:検証実験構成

| | |
|------------|-------------------------------------|
| ノートPC | Let's Note R7 |
| CPU | Intel(R) Core 2 Duo U7500 (1.06GHz) |
| OS | Windows 7 |
| 環境情報収集ユニット | 宮川製作所 MTC501 |
| 交流電流計 | 宮川製作所 SY111-CM |
| パケットキャプチャ | Wireshark |

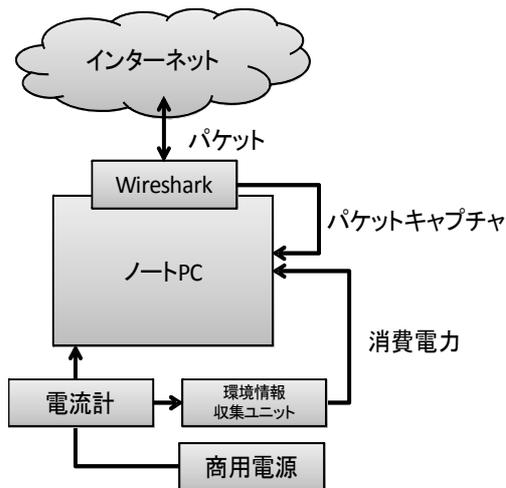


図 2:検証実験環境

実験環境の構成を図 2 に示す。ノート PC をインターネットに接続し、Wireshark[4]を用いてパケットキャプチャを行い、ノート PC 内のハードディスクに記録した。また、クランプメータ式の電流計を用いて電流値の計測を行った。さらに、バッテリーへの充電の影響をなくすためにバッテリーを取り外して計測を行った。電流計で計測された電流値を環境情報収集ユニットに蓄積した。そして、環境情報収集ユニットから 10 秒ごとに平均消費電力をノート PC 内に蓄積した。

4.2 機械学習プログラムの実装

機械学習のプログラムの実装には筆者らが開発中のオンライン機械学習フレームワークの Jubatus[5]を利用した。このような数値の予測問題には本来は回帰分析を利用すべきであるが、Jubatus には回帰がまだ実装されてい

ないため、分類器を利用した。また学習アルゴリズムには Passive Aggressive 法を採用した。

分類のためのラベルには、電力計で計測された 10 秒間の平均消費電力を利用した。また、学習のための特徴ベクトルは、10 秒間に流れたパケットから、Source IP, Destination IP, プロトコル, パケット長の配列を利用した。

分類ラベル付の学習データを入力し機械学習を行い、消費電力とパケットの関係性のモデル構築を行う。次に推定データを入力し、パケットから消費電力を推定し予測値を出力させた。

5 予備実験

5.1 機械学習の妥当性検討

機械学習の妥当性を検討するために、2011/09/06 に計測したデータを学習データと推定データに利用し実験を行った。実験結果を図 3 に示す。

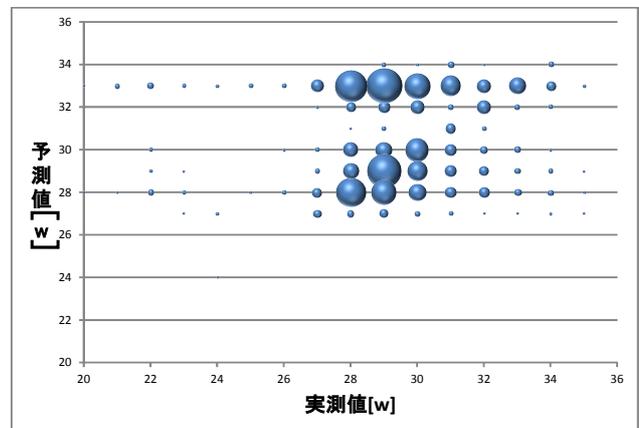


図 3:学習データと推定データを同一にした予備実験結果

5.2 予備実験考察

予備実験の結果から、予測値が 33w に引きずられていることが多いことが分かった。予測値が 33w になっている箇所について、パケットキャプチャを確認したところ、アイドル状態で ARP パケットを受信していることが多いことが分かった。

マシンがアイドル状態になっている際、基本的には外部から ARP パケットを受信するのみである。そのため、アイドル状態において、過去に ARP パケットを受信した際の電力を推定していると考えられる。

ARP パケットは L2 の MAC アドレスと L3 の IP を紐づけるために、定期的にネットワーク上のマシンから送出されるパケットである。そのため、ARP パケットを受信したとしても受信したマシンの内部状態に影響を与えることはない。本来であれば機械学習により ARP パケットとマシンの内部状態と無関係であるという学習が行われるはずである。しかし学習量が少ないため ARP パケットとマシンの内部状態が無関係であるという学習が行えなかったと考えられる。

5.3 ARPパケットの除外による精度向上

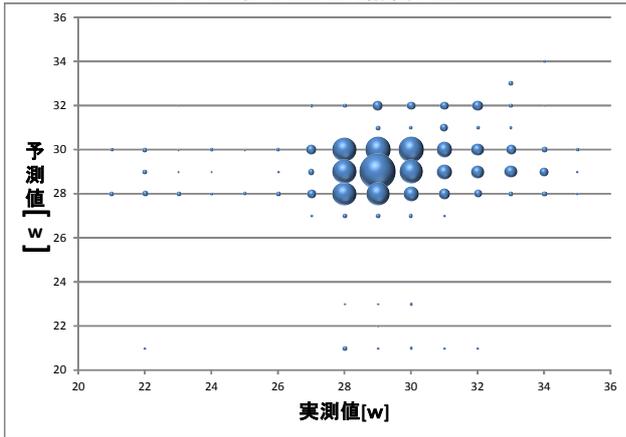


図4:ARPパケットの除外による精度向上

予備実験の考察から、ARPパケットにより精度が悪化している可能性があることが分かった。そこで、マシンの内部状態に影響を与えないARPパケットを学習データおよび推定データから除外し実験を行った。実験結果を図4に示す。33wの誤推定のラインが消え精度向上がうかがえる。

6 実験

2011/09/04に計測したデータと、2011/09/06に計測したデータを用いて、実験を行った。別の日に取得したデータを用いることで、図1にある同一構成で別のサーバという環境を再現した。また、予備実験の結果からARPパケットは除外して実験を行った。

6.1 実験1

2011/09/04に計測したデータを学習データとし、2011/09/06に計測したデータを推定データとして実験を行った。実験結果を図5に示す。

実験結果から、31wと推定されるデータが極端に少な

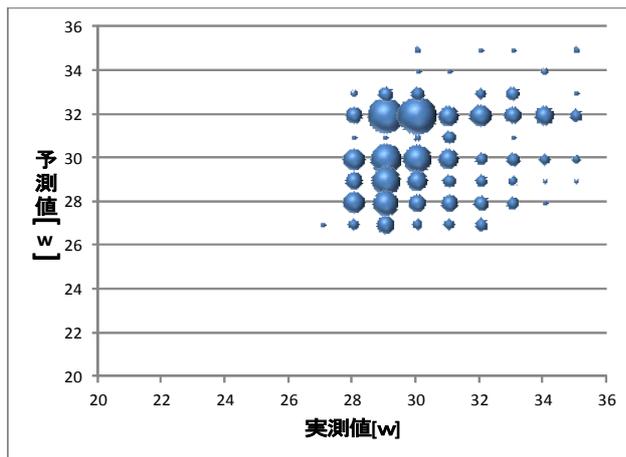


図5:学習データ2011/09/04, 推定データ2011/09/06の実験結果

く、32wと推定されるデータが多いことが分かった。

6.2 実験2

2011/09/06に計測したデータを学習データとし、2011/09/04に計測したデータを推定データとして実験を行った。実験結果を図6に示す。

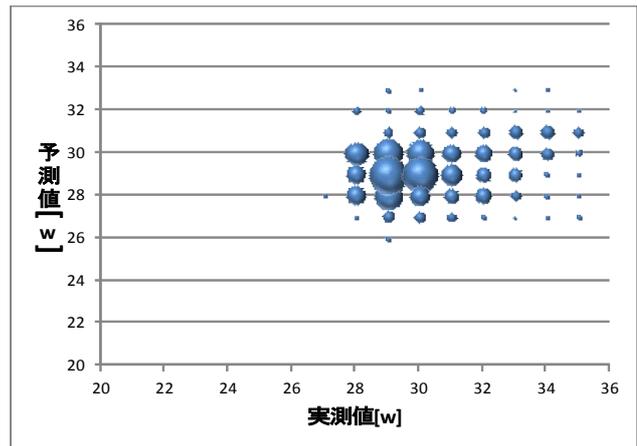


図6:学習データ2011/09/06, 推定データ2011/09/04の実験結果

実測値30w以上においては、実測値に対して推定値が低くなっていることが分かる。

7 考察

図5と図6の実験結果ではいずれも弱い正の相関が見受けられる。またいずれも実測値が34wなど領域では、予測値が実測値よりも小さくなってしまっているケースが多い。これは実測値が大きな領域において学習データが少なく、また29w付近での学習データが多かったため、適切な分類ができなかった可能性がある。そのため教師データとして利用するパケット群に偏りがある場合、教師データに補正が必要であると考えられる。

8 まとめ

クラウド環境を実現するデータセンタにおいて、細かい粒度で消費電力を知ることはデータセンタ事業者にとって重要である。本研究では通信パケットと消費電力との関係を機械学習することで、サーバ内のテナントに影響を与えずに消費電力を推定する手法を提案した。

提案手法の実験を行い、通信パケットと消費電力の間には、弱い正の相関があることを確認した。これにより通信パケットから消費電力を推測できる可能性を見出した。また、推定精度の向上が今後の課題である。

9 今後の課題

本実験ではWiresharkを利用してノートPC自身のNICのパケットキャプチャを自身に保存した。そのため、どのような通信が行われてもディスクI/Oが発生する。したがって通信量とディスクI/Oが相関を持つ環境になってしまっている。正確な実験を行うためには、図1に示したスイッチングハブやルータのポートミラーの機能を利用し、別端末でパケットキャプチャを行う必要がある。

本実験では機材調達の都合からノートパソコンを用いて評価実験を行った。そのため、消費電力の変動が少なく、また上で動作されていたアプリケーションもウェブブラウザやExcelといった、クライアントユースのものであった。今後は実運用されているサーバ機を用いて、ウェブアプリケーションやRDBMSを動作させて実験を行いたい。また、同一の端末を時間分割して擬似的に別の端末として取り扱い実験を行った。そのため、機種間の消費電力の差を機械学習で埋められるかどうかは明確では

なく、追加の検証が必要である。

本実験では一日のデータを教師データとして入力し、その後また別の一日のデータを推定するというバッチ処理的な利用をした。提案手法はオンライン機械学習を用いて実装されているため、リアルタイムに学習しながら推定を行うことができる。図1に示したような環境は、同一構成のマシンがロードバランサで分散されているため有効に働く可能性がある。そのため、複数のマシンを利用して、リアルタイムに学習を行いながら、リアルタイムに推定を行う実験を行いたい。

本実験は連続値の推定に対して分類アルゴリズムを利用した。このような問題は回帰分析アルゴリズムを利用すべきである。Jubatus への回帰分析アルゴリズムの実装は近日中を予定されており、実装され次第追試を行う。

参考文献

- [1] “HP Intelligent Power Distribution Unit – Overview & Features”,
<http://h18004.www1.hp.com/products/servers/rackandpower/powersupplies/ipdu/index.html>
- [2] Xiaobo Fan, Wolf-Dietrich Weber, Luiz Andre Barroso,
“Power Provisioning for a Warehouse-sized Computer” 2007
- [3] “ルーター等エネルギー消費効率及びその測定方法”,<http://www.meti.go.jp/committee/materials2/downloadfiles/g80917b03j.pdf>
- [4] “Wireshark・Go deep.”, <http://www.wireshark.org/>
- [5] “Jubatus : Distributed Online Machine Learning Framework – Jubatus 0.1.0 documentation”, <http://jubat.us/>
- [6] “Snort :: Home Page”, <http://www.snort.org/>

