

# 高効率ファシリティマネジメントを実現する ジョブ電力予測手法の提案

鈴木成人<sup>1</sup> 平岡美智子<sup>2</sup> 白石崇<sup>1</sup> クリシュパエンジ<sup>1</sup> 山本拓司<sup>1</sup>  
福田裕幸<sup>1</sup> 松井秀司<sup>2</sup> 藤崎正英<sup>1,2</sup> 宇野篤也<sup>3</sup>

**概要**：大規模な HPC システムは冷却ファシリティに大きな電力を消費している。冷却ファシリティの設定とその反映にはタイムラグがあるため、冷却対象である計算機の電力変動を予測することで、高効率なファシリティ制御を実現できる。従来の計算機予測手法は、ジョブ投入時に得られるジョブ属性情報から、過去に類似の属性情報を持つジョブを検索し、その電力を予測値とする。この手法では、類似の属性情報・電力を持つジョブが過去にないと、必然として予測が失敗する。そこで、この予測結果を補正するために、ジョブの属性情報に加え、予測時点までの電力値を利用する 2 つの機械学習モデルを開発した。“電力相関トピックモデル”は、選抜された 10 個の類似ジョブ候補の中から、電力値を使用して最も予測対象に近いジョブを再選択する。“可変リカレントニューラルネットワークモデル”は、その時点までの電力値の長さによって最適な時系列モデルを動的に選択し予測を行う。「京」コンピュータの運用情報で検証を行い、2 つの特徴の違うモデルを統合することで、平均相対誤差 5.7% の高精度予測を実証した。

## 1. はじめに

演算性能の向上に伴い、近年の HPC システムは従来よりも多くの電力を消費している。2011 年の TOP500 において 1 位にランクされた「京」コンピュータにおいては、10PFlops の演算時に 12.6MW の電力を消費する。現在、最も電力を消費する HPC システムである天河二号は、61PFlops 動作時に 18.6MW の電力を消費する[1]。2021 年には、エクサスケールの HPC システムの実現が見込まれているが、HPC システム消費電力の制約によって実現が阻まれると報告されている[2]。それゆえ、より低電力のコンピュータハードウェア[3]や高効率ファシリティマネジメントが強く求められている。

ファシリティマネジメントの取り組みとして、データセンターの冷却ユニットの予測制御が研究されている[6, 7, 8]。冷却ユニットを高効率に運用するため、冷却対象である計算機システムの消費電力に合わせて、冷却量を制御する。冷却ユニットの熱伝達時定数は、計算機の電力の時定数よりも遥かに長いので、測定された計算機電力を使用して冷却ユニットを制御すると、冷却ユニットにエネルギー損失が発生する。そこで、将来の計算機電力を正確に予測し、これに合わせて冷却ユニットを制御することで、高効率なファシリティ運用を実現できる。

本稿では、HPC システム向けの計算機電力予測技術を提案する。クラウドや基幹系システムを運用しているデータセンターと異なり、HPC システムの電力はメガワット単位で急峻に変化するため、変化に追従した正確な計算機電力の予測が課題となる。

従来の計算機電力予測手法は、計算機上で実行される個々のジョブの電力を予測し、その合計値を計算することで計算機の電力を予測する。個々のジョブの電力予測では、予測対象ジョブにキューに入力される時点でジョブが所有

している“属性情報”（例：実行ノード数、ユーザ ID、ジョブ名）が類似のジョブを、過去すでに実行済みのジョブの中から検索し、その電力値を予測値とすることでジョブの電力を予測していた（3.1 節）。これらの手法は、計算機電力の制御を目的とした Power-aware-scheduling と呼ばれる技術群の文脈で議論されていた[4,5]。本稿では特に急峻な電力変動に対応するため、ジョブの属性情報だけでなく、その時点までのジョブ電力（ジョブ電力履歴）の両方を用いる予測システムを検討した（3.2～3.4 節）。

図 1 に提案する HPC システムの予測電力制御のブロック図を示す。HPC システムのユーザはジョブスケジューラにジョブを投入する。ジョブスケジューラはジョブを順に計算機で実行し、その結果は計算機電力へ反映される。予測システムは、ジョブ属性情報とジョブ電力履歴から、将来の計算機電力を予測する。冷却ユニットは、計算機電力の予測値に合わせて冷却量を事前に制御することで、冷却ユニットの電力を削減する。

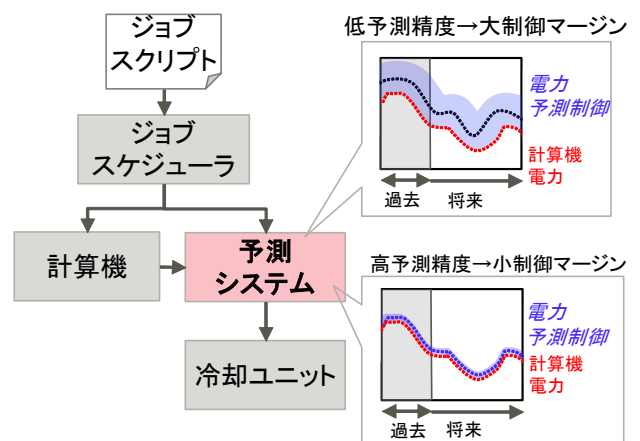


図 1 提案する電力予測制御を搭載した HPC システム。

1 株式会社富士通研究所  
2 富士通株式会社

3 国立研究開発法人理化学研究所 計算科学研究センター

次に、目標とする予測先期間と予測精度を示す。予測先期間は冷却ユニットの熱伝達時定数に依存する。冷却ユニットでの熱伝達時定数は最大で 30 分である[7, 8, 9]。したがって、将来 30 分までの計算機電力を予測し、予測結果に基づいて冷却ユニットを制御することにより、熱伝達時定数を考慮した最適な冷却ユニット制御を実現できる。電力予測の精度は、予測制御で取るべき制御マージンに影響する。電力予測の精度が低いと大きな制御マージンが必要になり、電力削減効果が減ずる。今回は、冷却ユニットの電力削減率の目標を 10%と定め、これを実現する予測精度として、相対誤差 12%以下を目標とした。

本稿の残りでは、キュージョブ予測モデルとジョブ実行中予測モデルを組み合わせた高精度な総ジョブ電力予測システムを提案する。2 章では、「京」コンピュータでキューに入れられたジョブと実行中ジョブの比率に関する分析結果から、実行中ジョブ予測の重要性を示す。3 章は提案する予測モデルを示す。3.1 節は、ジョブスクリプトから抽出されたジョブ情報を活用して、トピックモデルに基づくキュージョブ予測モデルを説明する。3.2 節では、ジョブ情報とジョブ電力履歴を使用した実行ジョブ予測モデルについて説明する。これらを組み合わせた統合予測モデルを 3.3 節で説明する。4 章では提案した統合予測モデルの評価結果を示す。3 か月の「京」コンピュータ運用実績を予測し、相対誤差 5.7%の高精度予測を実証した結果を説明する。5 章で関連研究を示し、6 章で結論とする。

## 2. ジョブ分析

予測したい将来のある時点の総ジョブ合計電力は、現在キューに入っているジョブの電力と現在実行中ジョブ電力の合計と考えることができる。図 2 は、ジョブの状態の例を示している。現時点では実行中ジョブ (Job2, Job3, Job6, Job7, Job8) の合計が総ジョブ合計電力となる。Job9 と Job10 は現在キューに入っているジョブだが、現在～予測先時間の間に実行状態に遷移する。

ここで、キュージョブと実行中ジョブでは得られる情報が異なる。キュージョブは、キューに入力される時点でジョブが所有している“属性情報” (例：実行ノード数、ユーザ ID、ジョブ名) を得ることができる。一方で、実行中ジョブは“属性情報”に加えて、その時点までのジョブの“電力履歴”を得ることができる。

計算機電力予測の既存研究では、“属性情報”のみを使用したキュージョブの電力予測しか提案されていなかった。既存研究が Power-aware-scheduling のために検討された技術であり、ジョブ実行前の時点でその電力を予測する必要があるためである (詳細は 5.関連研究を参照)。一方で、冷却ユニットの制御では実行中ジョブの予測結果を制御に反映させることができる。本論文の実行ジョブ電力予測では、“属性情報”と“電力履歴”の 2 つの情報を使用することで、

従来よりも高精度の予測手法を検討した。

本論文では、各ジョブ状態から得られる情報を最大限活用することで、キュージョブ・実行中ジョブの高精度な予測モデルを開発した。

- キュージョブ予測モデル:ジョブがキュー状態にあるときにジョブスクリプトを使用して各ジョブの電力を予測する (3.1 節)
- 実行中ジョブ予測モデル:ジョブが実行状態にあるときに、ジョブスクリプトと電力履歴を使用して各ジョブの電力を予測する (3.2 節)

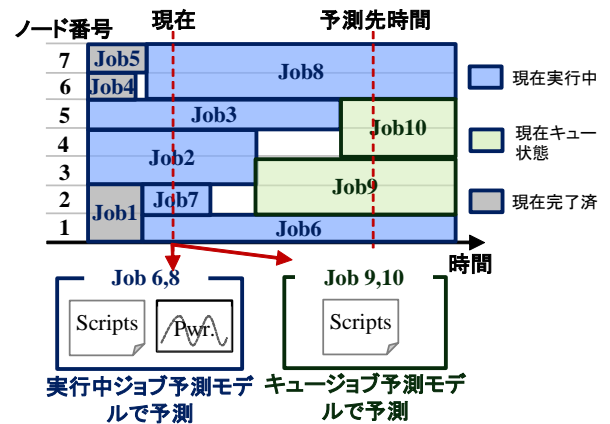


図 2 電力予測対象のジョブ状態の例。

実行中ジョブを予測する重要性を検証するため「京」コンピュータの運用情報を分析した。図 3 は、2017 年の 7 月から 9 月までの合計ジョブ電力におけるキュージョブと実行ジョブの電力比を示している。横軸は、予測する将来の時間を示している。電力比の評価は 5 分ごとに行われ、各期間の平均電力比が示されている。

現時点(予測先時間が 0)においては、計算機電力は 100% が実行中ジョブで構成されている。予測先時間を長くすると、キューに入れられたジョブの電力比が増加する。ここで目標である将来 30 分間の電力予測 (1 章で説明) に着目すると、総ジョブ合計電力の 87%が実行中ジョブで構成されている。将来 30 分間の総ジョブ合計電力予測では、実行中ジョブ予測の精度が総ジョブ合計電力予測の精度に大きく寄与することを示唆している。

本知見が他の拠点においても妥当であるか調査を行った。キュージョブと実行中ジョブの電力比は、その拠点におけるジョブの平均実行時間に依存している。平均実行時間が長いほど、ジョブ全体の実行ジョブの割合が大きくなる。「京」コンピュータを使用した分析では平均実行時間は 210 分であった。[11, 12, 13]によると、他の大規模 HPC サイトでは、平均実行時間はそれぞれ約 30 分, 110 分, 360 分だった。この値から他拠点でも将来 30 分の実行中ジョブの電力比率が高いことが推定される。

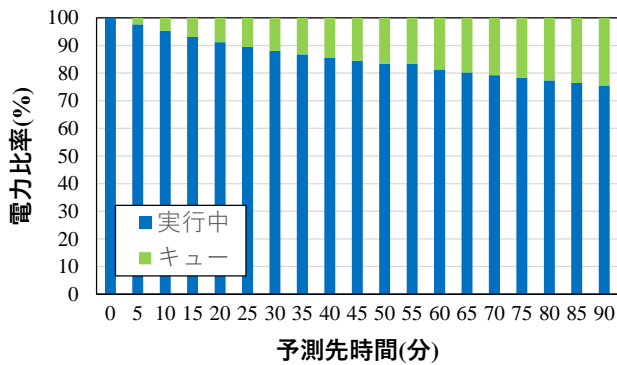


図3 予測先時間に対するキュージョブと実行中ジョブの電力比依存性.

### 3. 電力予測システム

本章では、ジョブ属性情報のみを使用した従来の電力予測モデル (TPC, PROB) について説明し、提案するジョブ属性情報とジョブ電力履歴の両方を使用する予測モデル (CTPC, VRNN と 2 つを統合した INTEG モデルについて説明する。

図4は、提案する電力予測システムの全体構成図である。トピックモデル (TPC) と確率モデル (PROB) を含む、キュージョブの電力予測モデル (3.1 節) を開発した。キュージョブ予測モデルは、ジョブスクリプトを使用することにより、ジョブ開始から完了までの電力を予測する。また、ジョブ実行中電力予測モデルも開発した (3.2 節)。ジョブ実行予測モデルは、過去のジョブ電力履歴と TPC 出力を使用し将来のジョブ電力を予測する。実行中ジョブ予測モデルは、“電力関連トピックモデル” (CTPC, 3.2.1 項) と“可変リカレントニューラルネットワークモデル” (VRNN, 3.2.2 項) の 2 つのコア予測モデルで構成されている。

CTPC は、ジョブ電力履歴を比較することにより TPC から出力された 10 個の候補から予測結果を再選択する。ただし、10 の候補に類似の電力を持ったジョブがない場合、予測は必然として失敗する。一方で、VRNN は過去すでに実行された多くのジョブ電力を使用して予測することができる。ただし、正確な予測結果を獲得するには長い電力履歴が必要となる。ジョブが実行に遷移した直後の状態では CTPC の平均予測精度は VRNN と比較し高いが、実行に遷移してから時間が経過するに従い、多くの過去実行済みジョブの電力履歴によって訓練された VRNN の平均予測精度が優位となる。この特性を活かし、2 つのモデルの予測結果の内、いずれかを動的に選択する統合予測モデル (INTEG, 3.2.3 項) を開発した。

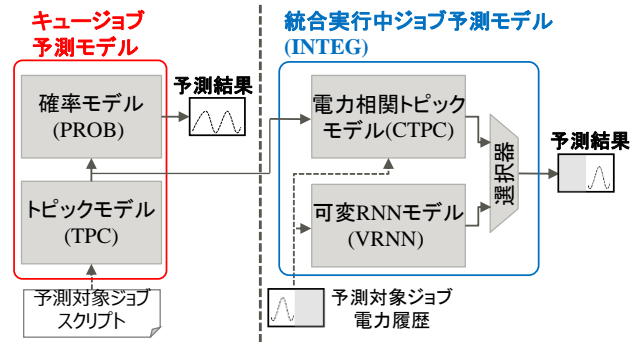


図4 提案する電力予測システムブロック図.

#### 3.1 キュージョブ予測モデル (TPC+PROB)

本節ではトピックモデル[14]を使用したキュージョブの予測方法について説明する。開発したモデルはトピックモデル (Topic model: TPC) と確率モデル (Probabilistic model: PROB) の 2 段階で構成されている (図5)。

トピックモデルは広く使用されている自然言語処理のアルゴリズムである[15]。トピックモデルのアルゴリズムを用いて予測対象のジョブと類似の属性情報を持った過去ジョブを検索し、その電力値を予測値とすることで予測を行うのが、TPC である。

TPC の類似手法が既存研究で提案されている。同様のジョブ属性情報を持つジョブは似た電力であることが報告されており[16]、スクリプトから得られるジョブ情報を使用した機械学習により、過去のジョブから最も類似したジョブを選択する電力予測手法が提案されている[17]。このアプローチでは、事前に属性情報の中で特に予測精度に貢献する特定のジョブ情報を分析し、その知見に基づいて、選択に使用する各ジョブ情報の“重み”を手動で調整していた。しかし、各ジョブ情報の“重み”は拠点毎に異なるため拠点毎の手動分析が必要となる。また、一度分析しても、投入されるジョブの傾向が変化すると再度手動での分析が必要となる。

そこで、各ジョブ情報に適切な重みを自動的に設定する確率モデルを開発した。確率モデルの訓練では、すでに実行が完了した予測対象ジョブと TPC モデル出力ジョブの各ジョブ属性情報とジョブ電力履歴を比較する。以下にその手順を示す。

- i. 予測対象ジョブ X と TPC 予測結果ジョブ Y の各ジョブ属性情報を比較し、一致するかどうかを判断する。
- ii. 動的タイムワーピング(DTW)メソッドを使用して、ジョブ X とジョブ Y のジョブ電力履歴を比較し、それらの類似性に基づいて成功の確率をスコアリングする。
- iii. i および ii の結果に基づいて、確率モデルは各ジョブ属性情報の重みを計算する。

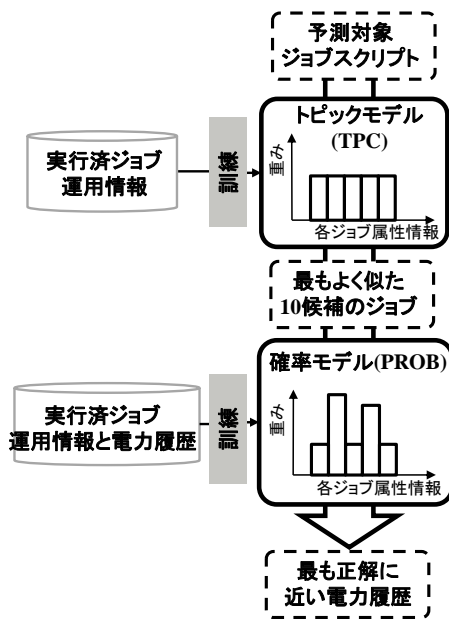
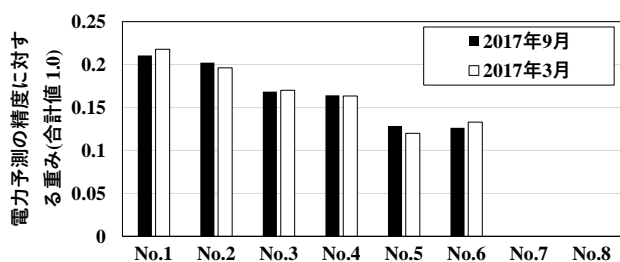


図5 提案するキュージョブ予測モデル (TPC + PROB) .

図6は、確率モデルで計算された「京」コンピュータ上の各ジョブ属性情報の重みを示している。No.2 要求ノード数と No.6 キューの両方がジョブのサイズを表現したジョブ属性情報である。No.2 は No.6 よりも詳細な情報を含むため、予測に対する寄与度も No.2 の方が大きいことが確認できる。また、No.7 ジョブ投入時間や No.8 ジョブ ID は、そのジョブに固有のジョブ属性情報であり、予測精度に一切寄与していないことが確認できる。

確率モデルによる予測では、得られた重みを使用して予測結果ジョブを変更する。まず TPC モデルから予測対象ジョブに対して属性情報が類似の候補を 10 個出力させ、その 10 個の各属性情報が予測対象に一致するかどうかを判定し、一致した属性情報に対して得られた重みでスコアリングする。スコアリングで1位となったジョブの電力履歴を確率モデルの予測結果として出力することで予測を行う。



No.	各ジョブ属性情報	詳細
No.1	要求実行時間	ユーザが指定する最大ジョブ実行時間
No.2	要求ノード数	ユーザが指定する最大要求ノード数
No.3	ユーザ	ジョブを投入したユーザ名
No.4	グループ	そのユーザが属するグループ
No.5	ジョブ名	ジョブスクリプトの名前
No.6	キュー	ジョブの規模感(small, large, huge)
No.7	投入時間	ジョブ投入時間
No.8	ジョブID	ジョブID

図6 確率モデル (PROB) による「京」コンピュータの各ジョブ属性情報重み分析結果.

「京」コンピュータでの 2017 年 7 月から 9 月までのすべてのジョブの時系列電力を積算し、計算機電力の予測結果を評価した。訓練は、予測時点から過去 3 か月のデータを 15 分間隔で学習した。予測対象と近い時間に実行されるジョブは、似た属性情報と電力履歴を持っている[10]ことがわかっているため、可能な限り短い間隔でモデルを更新することが望ましいためである。これを実証するため、学習間隔依存性を評価した。間隔を 15,30,60 分に変更すると、2017 年 9 月の総ジョブ合計電力相対誤差は 20.1%, 20.8%, 22.4% となり、学習間隔を短くすることで予測精度が向上することを確認した。

表 1 は、TPC と TPC+PROB モデルを使用した総ジョブ合計電力予測の相対誤差を示している。TPC+PROB モデルは、TPC モデルのみと比べて 3.1%精度が向上し、18%の平均相対誤差を達成しました。“重み”による予測結果の再選択が総ジョブ合計電力予測精度に寄与することを確認した。

表 1 ジョブ総電力予測結果の相対誤差比較

	7月	8月	9月	合計
トピックモデル (TPC)	17.8 %	22.0 %	23.6 %	21.1 %
キュージョブ予測モデル (TPC+PROB)	16.1 %	17.9 %	20.1 %	18.0 %

### 3.2 実行中ジョブ予測モデル

本節ではジョブ実行予測モデルを紹介する。3.2.1 項では、改良されたトピックモデルについて説明する。3.2.2 項では VRNN について説明する。3.2.3 節では、これらが統合されたモデルについて説明する。

#### 3.2.1 電力関連トピックモデル (CTPC)

TPC モデルの 10 候補の出力 (3.1 節で説明) 結果の中から、その時点までの予測対象の電力履歴を使用して 1 つを再選択する実行中ジョブ電力予測モデルを開発した (Correlated Power TPC:CTPC, 図 4)。再選択アルゴリズムは、ジョブ電力のタイムステップである 5 分ごとに動的に実行される。このアルゴリズムにより、キュージョブ予測モデルで失敗したジョブの一部が成功する。CTPC の再選択アルゴリズムの詳細は付録に示す。

TPC+PROB (3.1 節で説明) と CTPC でジョブ毎の予測精度を比較した。2017 年 7 月～9 月で「京」コンピュータのすべてのジョブを予測し、各ジョブの予測の成功率を比較した。図 7 は、平均の成功/失敗率を示している。図 7 に示すように、成功率は電力と経過時間の両方の基準として評価した。CTPC モデルは 5 分ごとに予測を動的に実行するが、開始から 30 分のタイミングでの評価結果を示してい



る。成功率は、TPC+PROB と比較しCTPC を採用することで成功率が3.8%向上した。

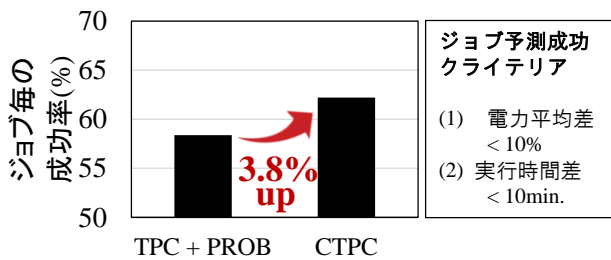


図7 キュージョブ予測モデル (TPC+PROB) と電力相関トピックモデル (CTPC) の予測精度比較

一方で、CTPC でも全ジョブの40%近くが依然として失敗している。図8は、各候補までのいずれかのジョブが予測成功した割合を示している。10番目の候補以降は成功率が飽和しており、候補の数を増やしても精度は改善されないことがわかった。予測精度をさらに向上させるためには、ジョブ情報のみから予測をするモデルと別のアルゴリズムの予測モデルが必要となる。

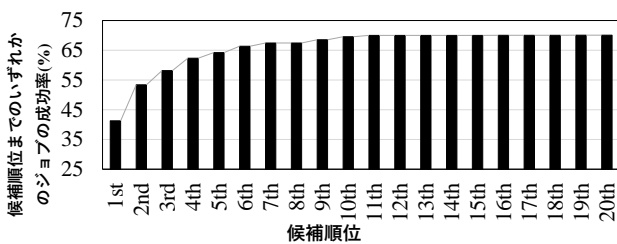


図8 キュージョブ予測モデル(TPC+PROB)と電力相関トピックモデル(CTPC)の予測精度比較

### 3.2.2 可変リカレントニューラルネットワークモデル (VRNN)

本節では、実行中ジョブ予測モデルの別アプローチとしてニューラルネットワーク(Neural Network: NN)を活用した予測モデルを提案する。NNの大きな利点はネットワークが巨大なデータから自動で特徴を抽出し、データ類似性を発見できることである。NNの一種であるリカレントニューラルネットワーク(Recurrent Neural Network: RNN)は、言語翻訳[19]や音声認識[20]が主な用途だが、近年は時系列データの予測技術として使用され始めている[21,22]。既存研究として、長い履歴データを使用してその波形自体の傾向分析を行っていた[22]。HPCにおける1つのジョブ電力波形は比較的短いため波形自体の傾向分析は用をなさない。そこで、過去のジョブの中から、予測対象に類似した電力波形の識別を課題として設定し、これを解決する予測モデルとしてRNNを使用して検討した。

まず「京」コンピュータのジョブ電力波形を分析した。これは、RNNで適切な学習を行うためには、過去データセ

ットに一定数の類似波形が含有されている必要があるためである。2017年の4月から12月までの約200,000個のジョブに対して、K-meansクラスタリング[23]を使用した波形分類を試みた。ジョブの最大実行時間は3日間であり、電力取得のステップは5分毎であるので、ステップ数は864である。864次元のK-meansにより、ジョブ電力の類似性に基づいてジョブパターンをクラスタ分類した。

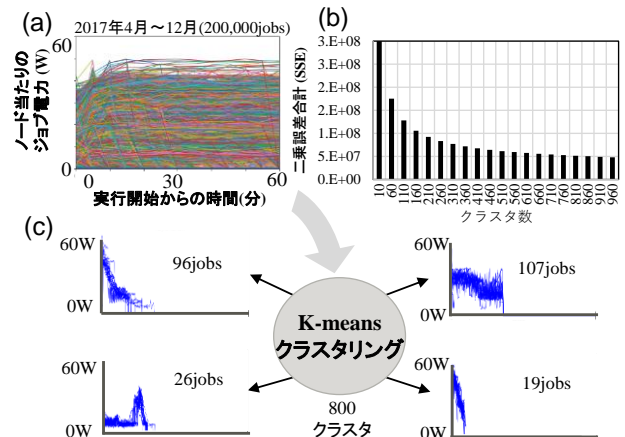


図9 キュージョブ予測モデル (TPC+PROB) と電力相関トピックモデル (CTPC) の予測精度比較

図9(a)は、分析したデータセットの実行開始～実行60分後までの電力波形を示している。縦軸はノード毎の電力を示す。[24]に示すように、ノード電力値はCPU監視温度と吸気流温度から定式化された電力値を使用している。最大電力値は58Wであり計算ノードの電力仕様と同じであり、定式化が適切であることを示している。

図9(b)は、エルボー法[25]を使用したクラスタ数毎の二乗誤差の合計(Sum of Squared Error: SSE)を示している。エルボー法による分析から分類数の適正值は800クラスタ程度であることがわかった。図9(c)は、800個にクラスタ分けされた内、4クラスタの電力波形の例を示している。クラスタ内の波形はよく似ていることが確認できる。800個のクラスタそれぞれには平均して250個の類似した電力波形を含有することになるため、RNNで適切な学習を行うのに十分なサンプル数が存在することを確認できた。

図9(a),(c)から電力波形は特徴的且つ急峻な変動を持った予測難易度が高い波形である。RNNモデルは乱雑な電力波形から予測結果を識別する必要がある。この課題を解決するため、可変サイズのRNNベースの電力予測モデル(Variable RNN:VRNN)を提案する(図10)。

VRNNは、予測先の時間間隔毎にサブモデルを有する予測モデルである。学習では、予測対象時間間隔の前までのすべての電力履歴をX、予測対象の電力履歴をYとして、各サブモデルで訓練される。ジョブ実行開始からすべての電力履歴を学習に使用することで、RNNに接続された各ニ

ューロンが電力履歴の変化タイミングを把握できるようにネットワークを構成する。ジョブ開始からの各電力履歴は、リカレントノードに並列に入力され、開始から特定の時間までの時系列間の関係が各々のサブモデルで学習される。ネットワークのサイズは、ジョブの実行時にジョブの電力波形全体を入力できるように拡張されていく。図 10 の左では、予測に使用できる電力履歴はほとんどないため、電力波形を適切に識別することは比較的困難であるが、時間が経過するに従って、ネットワークサイズと電力履歴が拡張され、予測精度が向上する。VRNN の訓練と予測アルゴリズムの詳細は付録に示す。

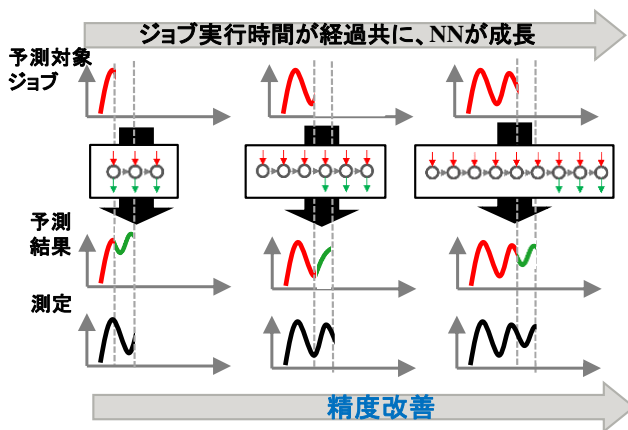


図 10 VRNN 電力予測モデルの概念図

VRNN モデルの有用性を確認するため、サブモデル毎の予測精度評価結果を図 11 に示す。2017 年 4 月の「京」コンピュータで実行された 24,991 ジョブを訓練に使用し、2017 年 7 月～9 月までのすべてのジョブを予測した。予測期間は 30 分に固定されている。正規化された RMSE である二乗平均平方根誤差率 (RMSEP) の全予測の平均値を評価指標とした。

$$RMSEP = \frac{1}{P_{max}} \frac{1}{m} \sum_{x=1}^m \sqrt{\frac{1}{n} \sum_{i=1}^n (A(x)_i - P(x)_i)^2}$$

$A(x)_i$  は、タイムステップ  $i$  でのジョブ X の測定電力である。 $P(x)_i$  は、時間ステップ  $i$  でのジョブ X の予測電力である。 $P_{max}$  はノードの定格電力 58W である。 $m$  は予測した総ジョブ数である。RMSEP は、すべてのジョブで平均化された、予測と実際の電力の正規化された RMSE を示している。横軸は、ジョブの開始からの電力履歴である。RMSEP は、0～90 分で大幅に改善される。この結果は、より長い入力データを持つ、より長いサブ RNN モデルは予測精度が高いことを示している。RMSEP は 90～180 分でほぼ安定する。これは、開始から 90 分以上の電力履歴で十分な予測精度が得られることを意味する。この結果から入力電力履歴に応じてモデルサイズを変更する VRNN の利点を確認した。

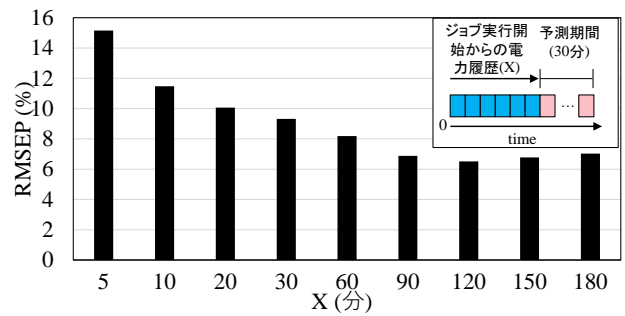


図 11 サブ RNN モデル毎の予測精度比較

### 3.2.3 統合実行中ジョブ予測モデル (INTEG)

本項では、CTPC (3.2.1 項) と VRNN (3.2.2 項) が統合された実行中ジョブ予測モデル (Integrated run-job prediction model: INTEG) を説明する。CTPC と VRNN の予測結果には異なる特徴がある。VRNN は膨大な電力波形から学習を行うことで正確な電力予測を可能にするが、電力履歴が少ない状態では十分に能力を発揮できない。一方で、CTPC は電力履歴が少なくても比較的精度良い予測が可能となる。CTPC と VRNN の 6 つの予測結果例を図 13 に示す。図の左側 (ジョブ A, B, および C) は、CTPC の予測結果が VRNN の予測結果よりも予測精度の点で優れているジョブの例である。10 の候補の中に予測対象に類似した電力波形がある場合、CTPC は電力変化点を正確に予測する。図の右側 (ジョブ D, E, F) は、VRNN の予測結果が CTPC の予測結果よりも予測精度が優れているジョブの例である。この例では、10 の候補に成功データがなく CTPC の予測が失敗しても、VRNN モデルは非常に正確に結果を予測する。VRNN は、過去の膨大な電力波形から学習されるため、その予測が完全に失敗することはない。

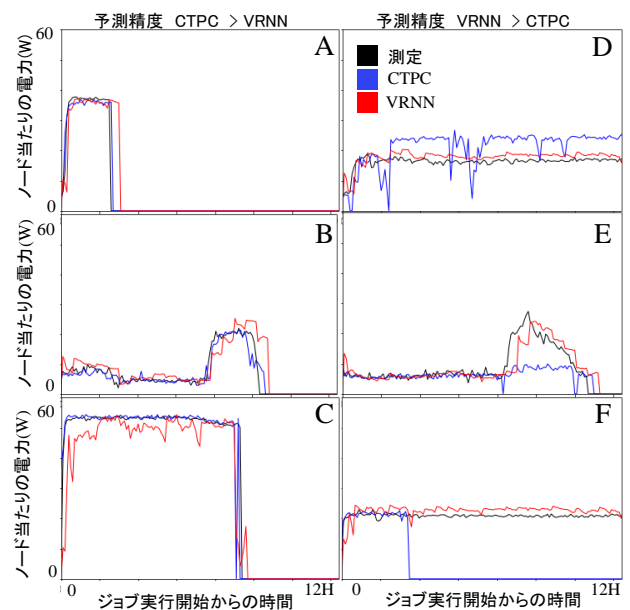


図 13 6 つのジョブにおける予測例 (実測と CTPC, VRNN の予測結果)

これら2つの予測モデルの予測結果から適切な方を選択するINTEGを使用することで、さらに予測精度を向上させることができる。図12は、ジョブ実行中予測モデル(INTEG)全体の概念図である。VRNNの予測結果とCTPCの予測結果は、アルゴリズム1と同様のアルゴリズムで5分毎に比較され、適切な予測結果が選択される。

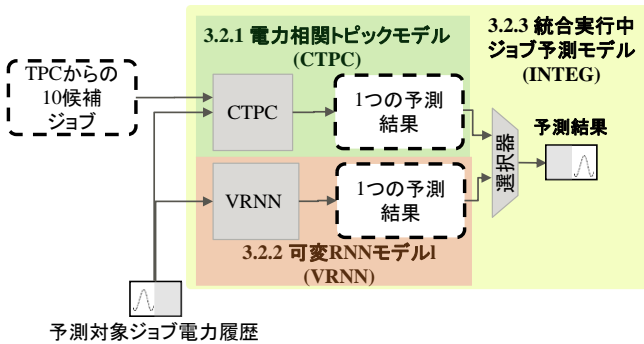


図12 CTPCとVRNNから構成される統合実行中ジョブ予測モデル

図14は、各予測モデル(CTPC, VRNN, INTEG)でのジョブ実行開始から各時間の平均相対予測誤差の比較を示している。評価は「京」コンピュータで2017年7月から9月に実行されたすべてのジョブの予測結果で行った。各モデルの訓練間隔は表3に示されている。

VRNNの相対予測誤差は、予測先時間が伸びるにつれて減少する。これは3.2.2項で説明した通り、VRNNが長い電力履歴から波形を認識できるようになるためである。一方、CTPCの相対予測誤差は、予測時間ステップが増加するにつれてわずかに増加する。予測の再選択は10個の候補内で行われているため、正解がその中に存在しない場合では予測が失敗し続け、相対誤差が大きくなっていくためである。タイムステップが30分まではCTPCの方が相対誤差が小さいが、以降VRNNと逆転する。VRNNはタイムステップ120分まで相対誤差が改善する。これは、この期間には訓練に使用できるジョブが少なく、モデルは少量のジョブから正解を認識すればよいので、予測難易度が比較的低位だと考察できる。詳細な分析は今後の取り組みとする。

INTEGの相対誤差はCTPCとVRNN両方の結果よりも小さくなっている。統合モデルからの予測誤差は、15分後に6%未満の相対誤差となり30分後に安定する。また、ジョブの開始時でも、INTEGは8%未満の相対誤差で予測できる。INTEGでは、CTPCとVRNNの選択は、5分毎に各予測タイミングで動的に行われる。INTEGの相対誤差曲線は非常に滑らかである。アルゴリズム1における過去30分の電力タイムステップの平均値を使用した選択をすることで、選択されるモデルが安定し、INTEG相対誤差のタイム

ステップ依存性は滑らかになる。

本節を要約する。ジョブの開始直後はCTPCの予測結果がINTEGによって頻繁に選択される。少数の電力時間ステップから電力形状を識別できないVRNNは、この期間中は不正確な予測結果を出力する。タイムステップが30分より後では、VRNNの予測結果が頻繁に選択される。ただし、この期間にCTPCが選択されることもある。これは、VRNNが回帰モデルの一種であり、一時的にわずかなエラーが発生することがあるためである。この原因は訓練データセットのノイズが原因であると推定される。一方で、CTPCは10個の候補の再選択アルゴリズムであり、同じユーザが同一のジョブを実行することがあるため、予測結果は実際のデータと完全に一致することがある。

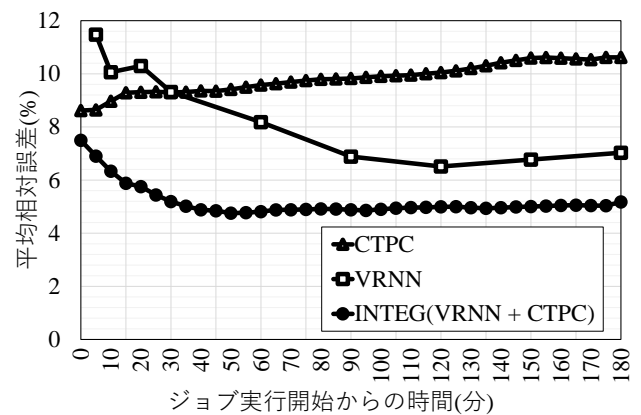


図14 6つのジョブにおける予測例(実測とCTPC,VRNNの予測結果)

#### 4. 評価

本章では、開発した予測システムで「京」コンピュータの総ジョブ合計電力を予測し、その評価結果を示す。表2に訓練と予測に使用した計算環境を示す。VRNNは訓練のみにNvidia GPU P100を使用した。

表2 本予測に使用した計算機仕様

CPU	Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz 2sockets, 14core/socket
Memory	2400 RDIMM 128GB
System disk	SAS 12Gbps, 10krpm
GPU	NVIDIA P100 2slot

表3の訓練条件の表に、訓練データ・訓練間隔・訓練時間を示す。2017年7月から9月までの全ジョブの電力予測を行った。この3か月の総ジョブ合計電力は、2017年に最も標準偏差が大きい予測が最も困難な3ヶ月である。キュージョブ予測モデルは、過去3か月のすべてのジョブを訓

練に使用した。キュージョブ予測モデルは、15分毎に過去3ヶ月すべてのジョブを訓練データとし、15分ずつずらしながら訓練される。学習された1つのモデルで将来15分間に実行されるジョブの電力を予測する。CTPCは、キュージョブ予測モデルの予測結果を使用するため訓練の必要はない。本評価ではVRNNモデルは1度だけ訓練し更新しない。VRNNの学習は、1ヶ月(4月)のすべてのジョブ電力をGPUで訓練させ78時間を要した。訓練期間(4月)と予測期間(7月から9月)は時期が離れている。訓練期間が予測期間に近いほど予測精度が高くなると想定できる。時期が離れてはいるが本評価のVRNNによる予測の相対誤差は5.7%と目標に対して十分小さい。

表2 訓練条件

	キュージョブ 予測モデル	実行中ジョブ予測モデル	
		CTPC	VRNN
訓練データ	過去3ヶ月 すべてのジョブ (6~8万 jobs)	-	2017年4月 すべてのジョブ (2万5千 jobs)
訓練間隔	15分	-	1度だけ
訓練時間	5分	-	78時間

図15に「京」コンピュータの計算機電力の予測評価結果を示す。評価期間は2017年7~9月で、予測対象の総ジョブ数は81,500個だった。すべてのジョブ電力が予測され、予測された電力を実際の実行開始時刻に揃えて和を計算することで予測計算機電力を算出した。測定された計算機電力も同じように算出された。相対誤差が使用し、予測した計算機電力と測定された計算機電力を比較評価した。

図15の縦軸は、取得した相対誤差と合計平均相対誤差の月毎の平均を示しています。4つのモデルの予測結果を比較した。

1. **TPC (従来)**: TCPモデル(3.1節)でキュージョブと実行中ジョブの両方を予測した結果
2. **TPC + PROB**: TCP+PROBモデル(3.1節)でキュージョブと実行中ジョブの両方を予測した結果
3. **TPC + PROB and CTPC**: キュージョブはTPC+PROBで予測し、実行中ジョブはCTPC(3.2.1項)で予測した結果
4. **TPC + PROB and INTEG**: 3のジョブ実行予測モデルをCTPCからINTEG(3.2.3項)に変更して予測した結果

図15を見ると、提案されているすべてのアルゴリズムの予測精度がTPCよりも優れている。TPCとINTEGを比較すると、予測精度は14.4%向上した。毎月の相対誤差でも同じ傾向の改善が見られた。改善は予測期間に依存していなかった。提案する予測モデルでは平均相対誤差5.7%で

の予測を確認した。予測誤差は、拠点や予測期間長、精度評価関数に依存すると考えられる。他拠点の運用情報を所持していないため直接の比較はできないが、本取り組みは長期間の大規模なHPCシステムの運用情報を使用して実証された最も正確な計算機電力予測システムと考えられる。詳細は関連研究(5章)で説明する。

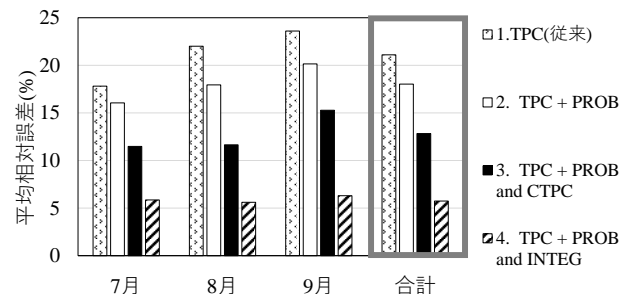


図15 4つの予測モデルによる総ジョブ合計電力予測結果比較

図16にジョブ総測定電力と予測電力の例を示す。ここで示している電力値は瞬時値ではなく30分間の平均値である。これは、30分間のジョブ電力平均値が冷却ユニット制御の制御指標であるためである。電力の変化が小さい期間(7月24~26日)では、キュージョブ予測モデルとキューと実行中ジョブ予測モデルの両方で高精度な予測が可能である(C, D)。キュージョブ予測モデルの予測が失敗した場合、ジョブの完了まで予測は失敗し続け、エラーは大きくなる。実行中予測モデルを使用することで補正できる(A, B)。

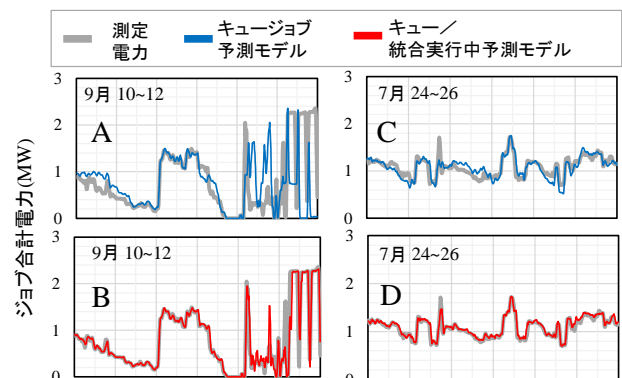


図16 総ジョブ合計電力予測結果の例  
A, C: TPC 予測結果 (9月10~12, 7月25~26)  
B, D: TPC+PROB+INTEG 予測結果 (A, Cと同じ期間)

## 5. 関連研究

機械学習モデルを使用してキュージョブの電力を予測し、予測結果からジョブのスケジューリングを行う、Power-aware-scheduling技術が提案されている[4]。100,000個のジ



ジョブで訓練されたランダムフォレストモデルは、Aurora コンピュータの運用情報を使用して、予測対象ジョブに類似した過去実行済ジョブを検索し、検索したジョブの電力を予測結果とする。ジョブ毎の平均予測誤差は 5%と計算された。評価は、時系列のジョブ電力ではなく、平均電力値で行われている。予測評価に使用したジョブ数は、943 および 714 個だった。一方、本評価では、3 か月すべてのジョブ (81,500 ジョブ) に対して予測を行う大規模な評価を実施している。Luna コンピュータにおいて電力キャッピングのための統計モデルによる消費電力予測が報告されている[26]。統計モデルのパラメータは、階層型ベイジアンモデルを使用して決定される。決定されたパラメータとシステム電力履歴が入力された統計モデルでシステム電力を予測する。Power-aware-scheduling に動的学習器を使用した電力予測が報告されている[10]。IBM Blue Gene/Q の運用情報で検討された。動的学習機は、単純なルールで電力を予測する。このルールでは、同じユーザのジョブの以前のジョブ電力が予測電力として使用される。この傾向は HPC 拠点毎に異なるため、この技術をそのまま他の拠点に適用することはできない。本論文では、この課題に対し確率モデルによる自動重み調整を行っている。同じアプリケーションのジョブが同様の電力履歴を持っているという知見から開発された、アプリケーション予測技術が報告されている[27]。Word2vec モデルはスクリプト内のハッシュとシンボルから 328 種類のアプリケーションにジョブをクラスタリングする。2016 年 9 月～2017 年 3 月の「京」コンピュータで実行された 273,121 ジョブで評価され、約 92%の精度で予測を実証した。この研究は、本論文のようなジョブ電力予測ではなく、アプリケーションの予測にのみ言及されている。上記は、キュージョブの電力を予測する既存研究である。

HPC の実行中ジョブの"時系列"電力予測はほとんど行われていない。時系列予測の初期研究では、フィードフォワードネットワークや多種の線形統計モデルよりも RNN の方が予測精度で優位であることが示されている[28]。その後、多種の RNN ベースモデルが、短時間で小刻みな変化を有する外国為替レート[29]、通信工学のカオス時系列[30]、または株価[31]などの特徴の異なる時系列データに対して、それぞれ開発された。また近年では、RNN の一種である Long-Short Term Memory (LSTM) が汎用時系列予測として報告されている[22]。

本論文と既存 RNN モデルの違いは、既存 RNN では個々の時系列に対して個々モデルを適応している点である[32]。既存 RNN は長期間の過去の波形履歴から学習され、その将来を予測する。例えば、天気予報や株式市場では毎月・毎年といった長期間の波形履歴からその傾向を学習する。一方で、本論文の VRNN モデルは、ジョブ電力の分・時間といった短期間の波形履歴から訓練される。VRNN は、波形の傾向を学習するのではなく、類似の過去電力波形から

類似性を学習するように開発されており、既存研究とは異なるモデルである。

株価予測のためのマルチブランチ LSTM が報告されている[33]。株式市場データの部分 (各銘柄・各期間) 毎に LSTM で学習するモデルである。各 LSTM は、K-means によって分類された 4～5 セットのクラスタ化データから学習される。この方法は、ジョブ電力予測に適さない。ジョブプロファイルの形状が 400～800 種類あり、ジョブが終了するまで K-means で識別できないためである。提案する実行中ジョブ予測モデルは、事前クラスタリングなしで将来のジョブ電力を特定できる。

## 6. 結論と今後の取り組み

大規模 HPC システムの冷却ユニットを予測制御するために、高精度計算機電力予測システムを開発した。「京」コンピュータにおける 3 か月間のすべてのジョブを予測し、その合計である総ジョブ電力において平均相対誤差 5.7% の高精度予測が実証された。トピックモデル (TPC) と確率モデル (PROB) で構成されるキュージョブの電力予測モデルを開発した。キュージョブ予測モデルの利点は、パラメータを未調整で容易に導入できることである。また、電力関連トピックモデル (CTPC) と可変リカレントニューラルネットワークモデル (VRNN) を組み合わせた統合実行ジョブ予測モデル (INTEG) を提案した。特徴の違う 2 つモデル (CTPC,VRNN) は、互いの欠点を補完するように統合され、高精度の予測を達成した。さらに、本論文は、我々の知る得る限り、大規模 HPC システムの合計電力を予測するために、キュージョブの電力予測と実行中ジョブの電力予測を組み合わせた最初の試みである。

本予測システムを冷却ユニットの制御に適用することで、冷却ユニットの制御マージンを減らし、その電力を削減できる。例えば、3 MW の冷却ユニットがある HPC システムに予測制御を適用する。相対誤差は 21.1%と 5.7%であり、それぞれ 0.63 MW と 0.17 MW の制御マージンが必要となる。したがって、予測精度が向上することで、0.47 MW (冷却装置の電力の 16%) を削減できる。

今後の取り組みとして、提案された予測モデルを他の HPC システムに採用し、その汎用性を確認する。また、冷却ユニットとジョブスケジューラの電力制御実験を行なう。

## 参考文献

- [1] TOP500 Lists. <https://www.top500.org/lists/2018/11/>
- [2] The race to exascale: A story of superpowers and supercomputers. <https://www.datacenterdynamics.com/analysis/superpowers-supercomputers-and-race-exascale/>
- [3] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In Proceedings of the 44th Annual International Symposium on Computer Architecture, pp. 1-12.

- ACM, 2017.
- [4] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini. Predictive Modeling for Job Power Consumption in HPC Systems. In High Performance Computing: 31st International Conference, ISC High Performance 2016, Frankfurt, Germany, June 19-23, 2016.
- [5] Whitepaper: "Energy and Power Aware Job Scheduling and Power Management", Energy Efficient HPC Working Group, Working draft, [https://eehpcwg.llnl.gov/documents/conference/sc17/sc17\\_bof\\_epa\\_jsrm\\_whitepaper\\_110917](https://eehpcwg.llnl.gov/documents/conference/sc17/sc17_bof_epa_jsrm_whitepaper_110917)
- [6] H. Endo, H. Kodama, H. Fukuda, T. Sugimoto, T. Horie and M. Kondo, "Effect of climatic conditions on energy consumption in direct fresh-air container data centers," IEEE 4th International Green Computing Conference (IGCC), June 27-29, 2013
- [7] Y. Tarutani, K. Hashimoto, G. Hasegawa, Y. Nakamura, T. Tamura, K. Matsuda, and M. Matsuoka, "Temperature distribution prediction in data centers for decreasing power consumption by machine learning," December 2015.
- [8] G. Serale, M. Fiorentini, A. Capozzoli, D. Bernardini, A. Bemporad Model predictive control (MPC) for enhancing building and HVAC system energy efficiency: problem formulation, applications and opportunities Energies, 11 (2018), p. 631
- [9] Y. Chai, A. Wu, N. Dong, Y. Wang and Y. Li: "Dynamic Operation and Control Strategy of Absorption Chiller under different working Conditions" In proceedings of the 13th World Congress on Intelligent control and Automation, 2018
- [10] S. Wallace, X. Yang, V. Vishwanath, W. E. Allcock, S. Coghlan, M. E. Papka, and Z. Lan, "A data driven scheduling approach for power management on hpc systems," SC16: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 656-666, Nov 2016.
- [11] M. Zasadziński, V. Muntés-Mulero, M Solé and T. Ludwig: Mistral Supercomputer Job History Analysis(2018), <https://arxiv.org/abs/1801.07624>.
- [12] E. Meneses, X. Ni, T. Jones and D. Maxwell: "Analyzing the Interplay of failures and Workload on a Leadership-Class Supercomputer," In Cray User Group Conference, 2015.
- [13] W. Joubert and S. Su: "Application workloads on the jaguar cray xt5 system," In Cray User Group Conference, 2012.
- [14] S Suzuki, M Hiraoka, T Shiraishi, H Fukuda, T Yamamoto, S Matsui, A Uno, "Power prediction with probabilistic topic modeling for HPC", ISC2019 HPC RESEARCH POSTER, Frankfurt, Germany ,2019
- [15] Papadimitriou, Christos; Raghavan, Prabhakar; Tamaki, Hisao; Vempala, Santosh "Latent Semantic Indexing: A probabilistic analysis" (Postscript). Proceedings of ACM PODS: 159-168. doi:10.1145/275487.275505. ISBN 978-0897919968., 1998.
- [16] Netti, A., Galleguillos, C., Kiziltan, Z., Sirbu, A., Babaoglu, O.: Heterogeneity-aware resource allocation in HPC systems. In: Proc. of ISC'18, vol. 10876 of Lecture Notes in Computer Science, pp. 3-21. Springer, 2018.
- [17] Q. Le, M. Ranzato, R. Monga, M. Devin, G. Corrado, K. Chen, J. Dean, and A. Ng. "Building high-level features using large scale unsupervised learning." In Proceedings of ICML, pages 81-88, 2012. [research.google.com/archive/unsupervised-icml2012.pdf](https://research.google.com/archive/unsupervised-icml2012.pdf)
- [18] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems, pages 3104-3112, 2014.
- [19] Graves, Alex, Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pp. 6645-6649. IEEE, 2013.
- [20] Guoqiang Zhang, B. Eddy Patuwo, and Michael Y. Hu. Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting, 14(1):35-62, 1998.
- [21] Flunkert, V., Salinas, D., & Gasthaus, J., DeepAR: Probabilistic forecasting with autoregressive recurrent networks. arXiv preprint arXiv:1704.04110, 2017.
- [22] Steinhaus, H., "Sur la division des corps materiels en parties" (French). Bull. Acad. Polon. Sci. 4 (12): 801-804. MR 0090073. Zbl 0079.16403, 1957
- [23] Keiji Yamamoto, Yuichi Tsujita, Uno, "Classifying Jobs and Predicting Applications in HPC Systems", ISC High Performance 2018: High Performance Computing pp 81-99, 2018.
- [24] A. Uno, H Hida, F Inoue, N Ikeda, T Tsukamoto, F Sueyasu, S Matsushita, F Shoji. "Operation of the K computer Focusing on System Power Consumption", IPSJ Transactions on Advanced Computing Systems Vol.8, No.4, pp.13-25, Nov 2015.
- [25] Robert L. Thorndike, "Who Belongs in the Family?". Psychometrika. 18 (4): 267-276. doi:10.1007/BF02289263, December 1953.
- [26] Storlie, C., Sexton, J., Pakin, S., et al. "Modeling and predicting power consumption of high performance computing jobs", arXiv preprint arXiv:1412.5247, 2014.
- [27] Y. Li, H. Hu, Y. Wen, and J. Zhang, "Learning-based power prediction for data centre operations via deep neural networks," in Proceedings of the 5th International Workshop on Energy Efficient Data Centres, no. 6. ACM, 2016.
- [28] J. T. Connor, L. E. Atlas, and R. D. Martin, "Recurrent networks and NARMA modeling," Advances in Neural Information Processing Systems 4. J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. San Matteo, CA: Morgan Kaufman, pp. 301-308, 1992.
- [29] Giles, C.L., Lawrence, S., Tsoi, A.C.: Noisy time series prediction using recurrent neural networks and grammatical inference. Machine learning 44(1-2), 161-183, 2001
- [30] Jaeger, H., Haas, H.: Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. science 304(5667), 78-80, 2004.
- [31] Hsieh, T.J., Hsiao, H.F., Yeh, W.C.: Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. Applied soft computing 11(2), 2510-2525, 2011.
- [32] Langkvist, M., Karlsson, L., Loutfi, A.: A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognition Letters 42, 11-24, 2014
- [33] X. Shao, D Ma., Y. Liu, Q. Yin, "Short-term forecast of stock price of multi-branch LSTM based on K-means", 4th International Conference on Systems and Informatics (ICSAI), pp. 1546-1551, 2017.

## 付録

### CTPC 再選択アルゴリズム

CTPC の再選択フローをアルゴリズム 1 に示す。ジョブ電力は 5 分ごとに取得され、タイムステップ  $i$  は開始からの現在までのタイムステップである。  $P(X)_0$  は、ジョブ開始時刻での候補  $X$  の予測電力である。  $P(X)_i$  は、タイムステップ  $i$  でのジョブ  $X$  の予測電力である。  $X$  は、トピックモデルから 10 の候補を示す 1~10 の数字である。  $PA_i$  はタイムステップ  $i$  における実際のジョブ電力である。 予測対象ジョブの過去 30 分～現在の電力履歴と、10 候補それぞれの同期間の電力履歴を単純平均(Simple Average: SA)で 5 分毎に比較した。 比較期間が 30 分に限定されているのは計算オーバーヘッドを考慮したためである。 予測対象ジョブが開始から 30 分 (6 ステップ) 以上経過していれば、10 個の候補すべてについて 30 分前～現在の SA を計算する。 開始から 30 未満の場合は、実行開始～現在の電力で SA を計算し評価する。 アルゴリズムは SA が最小となるジョブ  $i$  を、予測対象ジョブに最も電力に近いジョブと判断し、予測結果として選択する。

---

#### アルゴリズム 1: CTPC

---

**Run every 5 minutes (time step of the job power monitored)**

**Input:** Predicted power of candidate X:  $P(X)_0, P(X)_1, \dots, P(X)_i$

Actual power of the target:  $PA_0, PA_1, \dots, PA_i$

Current time step is  $i$

**Output:** Most similar candidate and future prediction power

**for** X=1 to 10 **do**

**where**  $i < 6$

**then**

**return**  $k=i$

**else**

**return**  $k=6$

**calculate**  $SA_x = \left| \frac{1}{k} \sum_{j=1}^k (P(X)_{i-j} - PA_{i-j}) \right|$

**return**  $SA_x$

**find** the job  $X$  with minimum  $SA_x$  and

**return** future power of the job  $X$  as prediction result

---

### VRNN アルゴリズム

アルゴリズム 2 は、VRNN の訓練アルゴリズムである。最初に訓練用のデータセットが作成される。ここで、 $P(X)_i$  はタイムステップ  $i$  でのジョブ  $X$  の電力である。このアルゴリズムでは、多くのサブモデルを訓練するために、各サブモデル毎に各ジョブの電力履歴を加工して学習データセットを作成する必要がある。ジョブ実行開始から 10~40 分後を予測するサブモデルでは、 $P(X)_0$  が訓練入力  $X$  に使用され、 $\{P(X)_1, P(X)_2, P(X)_3, P(X)_4, P(X)_5, P(X)_6\}$  は、訓練出力  $Y$  に使用される。入力期間に完了したジョブは訓練データセットから削除される。15~45 分と 20~50 分を予測するサブモデルの作成も同様に実行される。30~60 分を予測するサブモデルでは、 $\{P(X)_0, P(X)_1, P(X)_2, P(X)_3, P(X)_4, P(X)_5\}$  が訓練入力  $X$  に使用され、 $\{P(X)_6, P(X)_7, P(X)_8, P(X)_9, P(X)_{10}, P(X)_{11}\}$  は訓練出力  $Y$  として使用される。その後、サブモデルは 30 分ごとに作成され、サブモデルのネットワークのサイズは、入力データのサイズに応じて大きくなる。

次に、アルゴリズム 3 に示されている予測手順を説明する。予測は、5, 10, 20, 30 分、およびその後 30 分ごとに各ジョブに対して実行される。上記の予測処理は別々のサブ RNN モデルを使用して実行される。10, 20, 30 分の場合、予測結果に重複が起こるが、長い電力履歴から訓練されたサブモデルの方が高精度であるため、新しい予測結果で上書きすることで予測値を得る。

---

**アルゴリズム 2: VRNN 訓練**

---

**Run only once for training**

**Input :** All job power for training

Power of job X at each time step:  $P(X)_0, P(X)_1, \dots, P(X)_i$

**Output :** Trained model

/\*Create Dataset\*/

**for** n=1 **to** training job number **do**

**get** maximum length of JOB X as iMax (X)

**if** iMax (X) > 1

    x1 =  $P(X)_0$

    y1 =  $P(X)_1 \dots P(X)_6$

**store** x1, y1 to DataSet<sub>1</sub>

**end if**

**if** iMax (X) > 2

    x2 =  $P(X)_0, P(X)_1$

    y2 =  $P(X)_2 \dots P(X)_7$

**store** x2, y2 to DataSet<sub>2</sub>

**end if**

**if** iMax (X) > 4

    x4 =  $P(X)_0, P(X)_1, P(X)_2, P(X)_3$

    y4 =  $P(X)_4 \dots P(X)_9$

**store** x4, y4 to DataSet<sub>4</sub>

**end if**

**if** iMax (X) > 6

**for** Z=1 **to** int (iMax (X) /6) **do**

      x<sub>6Z</sub> =  $P(X)_0 \dots P(X)_{6Z-1}$

      y<sub>6Z</sub> =  $P(X)_{6Z} \dots P(X)_{6(Z+1)-1}$

**store** x<sub>6Z</sub>, y<sub>6Z</sub> to DataSet<sub>6Z</sub>

**end for**

**end if**

**end for**

/\*Training\*/

**for** N=1 **to** max number of Dataset<sub>N</sub> **do**

**create** sub-RNN Model<sub>N</sub> and **train** using Dataset<sub>N</sub>

**store** sub-RNN Model<sub>N</sub>

**end for**

---



---

**アルゴリズム 3: VRNN 予測**

---

**Run at i = 1, 2, 4, 6 and every 6 steps after that (12, 18, 24...)**

**Input:** prediction target job power from start to current time step *i*

**Output:** predicted power from i+1 to i+7

**search** sub-RNN Model<sub>i</sub>

**return** sub-RNN Model<sub>i</sub>

**calculate** prediction results using sub-RNN Model<sub>i</sub>

**return** prediction results from i+1 to i+7

**if** exist previous prediction power from i+1 to i+7

**overwrite** that power

**end if**

---