Vol.2018-HPC-167 No.21
2018/12/18

自動電源制御方式 JSCAPS の実運用システム"JSS2"への適用と評価

今出広明[†1]　加賀美崇紘[†1]　三鴨利彰[†1]　井口裕次[†1]　坂口吉生[†1]　藤田直行[†2]

概要：近年，社会全体で節電，省電力化の意識が高まるなか，PC クラスタにおいても省電力化の要請が強くなっている．JSCAPS は，ジョブが実行されていない待機状態にある計算ノードの電源を停止する自動電源制御方式の 1 つである．JSCAPS では，バッチジョブのスケジューリング結果を元に，電源停止，再投入処理に要する時間的，電力的なコストを考慮することで，既存のバッチジョブスケジューラで用いられる自動電源制御方式より高い省電力効果を実現している．我々は，JSCAPS を JAXA スーパーコンピュータ"JSS2"の実行環境"PP"に適用し，実運用上での省電力効果を評価した．その結果，計算ノードの待機中に消費する電力が 80%削減されたことを確認した．本稿では，JSCAPS を実運用に適用した際の効果を評価するとともに，運用上検出された影響などについて考察する．

キーワード：PC クラスタ，省電力化，自動電源制御，バッチジョブスケジューラ

1. はじめに

近年，社会全体で節電，省電力化の意識が高まるなか，PC クラスタにおける省電力化の要請が高くなっている．省電力化のための対策としては，省電力 CPU やメモリの採用により PC サーバ自身の消費電力を抑える方式や，その上でジョブを実行していない待機状態となっている計算ノードの電源を停止する方式などが挙げられる．

待機中の計算ノード（待機ノード）の電源を停止する方式は，ジョブの実行性能に影響を与えないことから有効な省電力方式の 1 つとして注目されている．バッチジョブスケジューラである Slurm[1]や LSF[2]では一定時間待機状態が継続した計算ノードに対し電源を停止し，次のジョブの実行開始予定時刻になると電源を再投入する自動電源制御方式が採用されている．

これに対し，我々はバッチジョブスケジューラによるスケジューリング結果を元に，電源停止，再投入処理に要する時間的，電力的なコストを考慮し，計算ノードの電源停止，再投入を判断する自動電源制御方式 JSCAPS (Job Scheduling Aware Power Save) を提案し，シミュレータ上で既存のバッチジョブスケジューラで用いられる自動電源制御方式より高い省電力効果が得られることを確認した[3]．JSCAPS は現在，富士通の HPC ミドルウェア である Fujitsu Software Technical Computing Suite に実装し，提供を開始している．

本稿では，JSCAPS を JAXA スーパーコンピュータ”JSS2”[4]のノード群”SORA-PP”内にある実行環境”PP”に適用し，シミュレータ上での評価結果と比較することで JSCAPS における省電力効果を評価する．また，JSS2 の省電力化において運用上検出された影響について考察する．

2. 自動電源制御方式 JSCAPS

Slurm，LSF で用いられる自動電源制御方式では，図 1 のように待機状態が一定時間以上経過した場合，電源停止処理が開始される．

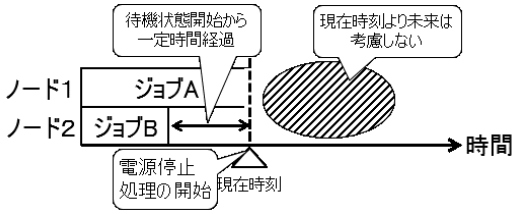

図 1 既存方式における電源停止処理

現在時刻より未来のスケジューリング結果は考慮されないため，次のジョブの実行開始予定時刻が直近であった場合，電源停止処理の完了直後に再投入処理を開始しなければならず，この間の消費電力は余分に消費される[1][2]．また，電源再投入処理は次のジョブの実行開始予定時刻となってから開始されるため，再投入処理の完了まで次のジョブの実行開始は遅くなってしまう．

JSCAPS はこれらの課題を解決するために，図 2 のようにバッチジョブスケジューラのスケジューリング結果から待機時間を予測し，あらかじめ決めた時間（電源停止基準待機時間）以上であれば電源を停止する．

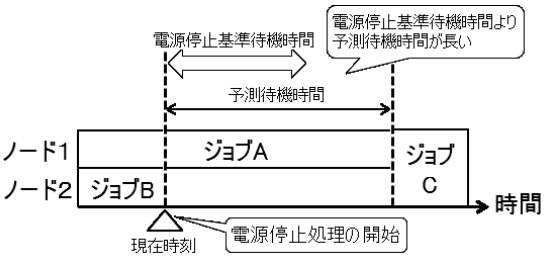

図 2 JSCAPS における電源停止処理

電源停止基準待機時間は予測した待機時間の間だけ電源を停止した場合の消費電力量と，停止しなかった場合の消費電力量を計算することで求めることができる．電源停止基準待

†1 富士通株式会社
Fujitsu Ltd.
†2 国立研究開発法人 宇宙航空研究開発機構
Japan Aerospace Exploration Agency

機時間以上電源を停止した場合，電源停止した場合の消費電力量が電源を停止しなかった場合の消費電力量よりも小さくなるように，電源停止基準待機時間を決定する[3].

また JSCAPS における電源再投入処理は，図 3 のように電源再投入処理に要する時間だけ次のジョブの実行開始予定時刻より前に開始する．次のジョブの実行開始予定時刻直前に電源再投入処理を完了させることで，次のジョブの実行が遅くならない．

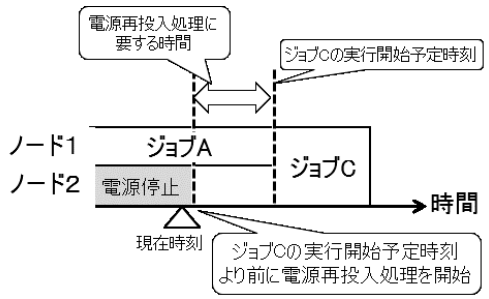

図 3 JSCAPS における電源再投入処理

JSCAPS におけるパラメータを以下に示す．

- 電源停止基準待機時間
- 最大停止ノード数

電源停止処理に要する時間，電源再投入処理に要する時間は対象の計算ノードの処理時間を計測することで決定することができる．最大停止ノード数は JSCAPS が停止する計算ノードの上限値を決めるパラメータである．大規模な実環境では，全ノードを一度に電源停止した場合急激な電力変動が発生し，電源設備に負荷がかかる可能性がある．最大停止ノード数を設定することで，大量の電源停止による電源設備への負荷を抑えることができる．

3. JSS2 における省電力運用の目標と課題

JSS2 では，投入されるジョブをバッチジョブと会話型ジョブの 2 種類に分類している． バッチジョブではユーザが指定したプログラムはバッチ処理により実行される．JSS2 では，バッチジョブの実行開始予定時刻はノードの利用状況に応じてバッチジョブスケジューラが決定している．会話型ジョブではユーザは対話処理により，プログラムのデバッグ処理やインタラクティブ処理等のプログラムを実行することができる．JSS2 では，会話型ジョブの実行開始予定時刻，つまり対話処理が可能となる時刻もノードの利用状況に応じてバッチジョブスケジューラが決定している．JSS2 においては，これら両方の運用において省電力化を実現することが目標となる．

JSCAPS では，新規に投入されたジョブがバックフィルにより電源停止しているノードに割り当てることができる場合，そのノードの電源を再投入し，再投入処理の完了を待ってジョブの実行を開始する．会話型ジョブが電源停止しているノードに割り当てられた場合，ユーザは対話処理のために再投入処理が完了するまでその場で待ち続けることになる．待ち時間が長くなった場合，ユーザが会話型ジョブの実行開始まで待ちきれずに諦めてしまうことが考えられ，会話型ジョブのキャンセルが増加する可能性がある．

4. JSS2 実行環境 PP と運用特性

JSCAPS の適用は JSS2 の実行環境 PP で行った．実行環境 PP は JAXA で運用される計算機システム JSS2 内のノード群 SORA-PP に属する実行環境である．実行環境 PP は PRIMERGY RX350 S8(RX350)が 138 ノードと PRIMERGY CX2550 M2(CX2550)が 40 ノードから構成される．実行環境 PP における各ノードの構成と消費電力について表 1 に示す．

表 1 実行環境 PP のノード構成

	PRIMERGY RX350 S8	PRIMERGY CX2550 M2
ノード数	138 ノード	40 ノード
CPU	Intel Xeon E5-2643 v2 3.5GHz×2	Intel Xeon E5-2643 v4 3.4GHz×2
メモリ	64GB	64GB
ジョブ実行中の平均消費電力	250W	170W
待機中の平均消費電力	180W	80W

ジョブ実行中の平均消費電力と待機中の平均消費電力は 2018 年 9 月 1 日～30 日に IPMI を用いて計測された消費電力を元に計算した．

2018 年 9 月 1 日～30 日までに実行環境 PP に投入されたジョブの内訳を表 2 に示す．

表 2 実行環境 PP へのジョブ投入状況

	ジョブ数	全ジョブに占める割合
バッチジョブ	59489	96.5%
会話型ジョブ	2180	3.5%
合計	61669	100%

計測期間中に投入された全ジョブにおける会話型ジョブの占める割合は小さい．

会話型ジョブにはパラメータとして，投入されてから実行開始までの最大待ち時間 wait-time を設定することができる．会話型ジョブの投入から実行開始までの待ち時間が wait-time を超過した場合，そのジョブは自動的にキャンセルされる．実行環境 PP では wait-time は 0 秒としていた．

また，実行環境 PP では会話型ジョブを投入したユーザの待ち時間をなくすために，5:00～20:00 の間は会話型ジョブを優

先的にスケジューリングしている.

5. 実環境での JSCAPS による省電力効果の評価方針

実環境における JSCAPS による省電力効果を評価するために, まず SWoPP2017 において評価したシミュレーション結果[3]と同様の省電力効果が実環境においても得られるのかを評価する. 次に, JSS2 において長期間運用した場合の省電力効果を評価する.

1. バッチジョブ中心の実環境における省電力効果の評価
   SWoPP2017 ではバッチジョブが投入順にスケジューリングされるシミュレータ上において JSCAPS の省電力効果を評価した. JSS2 では会話型ジョブを優先するスケジューリングが行われているため, バッチジョブの実行が中心となっている期間に電力情報を計測し, 省電力効果をシミュレーション結果と比較する.
2. 会話型ジョブを考慮した JSS2 における省電力効果の評価
   JSS2 において会話型ジョブが優先される期間も踏まえて長期的に計測し, 省電力効果を確認する. JSS2 では運用ポリシーとして会話型ジョブを優先しているため, JSCAPS の適用による会話型ジョブへの影響を確認する.

ノード状態の確認や消費電力の計測は IPMI を用いて 1 分おきに実施する.

JSCAPS による省電力効果を確認するための評価指標として,「待機電力の削減割合」を用いる. 待機電力の削減割合は以下の式で求める.

$$(\text{待機電力の削減割合}) = \frac{(\text{JSCAPS による削減電力})}{(\text{JSCAPS 未適用時の待機電力})}$$

$$(\text{JSCAPS による削減電力}) = (\text{RX350 の停止ノード数} \times 180\text{W}) + (\text{CX2550 の停止ノード数} \times 80\text{W})$$

$$(\text{JSCAPS 未適用時の待機電力}) = (\text{計測した全待機ノードの総消費電力}) + (\text{JSCAPS による削減電力})$$

計測期間中の各計測タイミングにおいて計算した待機電力の削減割合の平均値を「待機電力の平均削減割合」とする.

会話型ジョブへの影響を確認するための評価指標として, 計測期間中に投入された会話型ジョブが実行開始までにキャンセルされた割合「会話型ジョブ実行前キャンセル率」を用いる. 会話型ジョブ実行前キャンセル率は以下の式で求める.

$$(\text{会話型ジョブ実行前キャンセル率}) = \frac{(\text{実行前にキャンセルされた会話型ジョブ数})}{(\text{投入された全会話型ジョブ数})}$$

また, ある時刻にジョブを実行しているノードの割合を示すために, 以下の式で求める「ノード稼働率」を用いる.

$$(\text{ノード稼働率}) = \frac{(\text{ジョブ実行中のノード数})}{(\text{実行環境 PP の全ノード数})}$$

計測期間中の各計測タイミングにおいて計算したノード稼働率の平均値を「平均ノード稼働率」とする.

6. バッチジョブ中心の実環境における省電力効果の評価

SWoPP2017 で評価したシミュレータでの省電力効果と実環境での省電力効果を比較するために, 実行環境 PP においてバッチジョブの実行が中心となっている期間(2018 年 8 月 25,26 日)の消費電力とノード稼働率を計測した. 実行環境 PP とシミュレータの比較を表 3 に, JSCAPS におけるパラメータ値を表 4 に示す. 最大停止ノード数はシミュレータと同様に, 計算ノード数と同じとした.

表 3 実行環境 PP とシミュレータの比較

	パラメータ値	
	実行環境 PP	SWoPP2017 のシミュレータ[3]
電源停止処理に要する時間	50 秒	33 秒
電源再投入処理に要する時間	450 秒	301 秒
計算ノード数	RX350: 138 ノード CX2550: 40 ノード	160 ノード
待機中の消費電力	RX350: 180W CX2550: 80W	180W

表 4 JSCAPS におけるパラメータ値

パラメータ名	パラメータ値	
	実行環境 PP	SWoPP2017 のシミュレーション[3]
電源停止基準待機時間	500 秒	335 秒
最大停止ノード数	178 ノード	160 ノード

計測結果を表 5 に示す. 比較として SWoPP2017 において評価したシミュレーション結果[3]も示す.

表 5 バッチジョブ中心の実環境における省電力効果

	実行環境 PP 8/25,26	SWoPP2017 のシミュレーション[3]
平均ノード稼働率	54.3%	50.3%
全待機ノードの総消費電力の平均値	0.7kW	1.6kW
JSCAPS 未適用時の待機電力の平均値	13.1kW	14.3kW
待機電力の平均削減割合	待機電力を 95%削減	待機電力を 89%削減

SWoPP2017 におけるシミュレーション結果と同様にバッチジ

ョブ中心の実環境においても JSCAPS による待機電力の平均削減割合が高いことが確認できる.

実行環境 PP における待機電力の平均削減割合が SWoPP2017 におけるシミュレーション結果より高くなった原因として, JSCAPS を適用した実行環境 PP 上の環境が待機電力の異なる RX350 と CX2550 から構成されているためと考えられる.

7. 会話型ジョブを考慮した JSS2 における省電力効果の評価

JSS2 における省電力効果を確認するために, 実行環境 PP における消費電力とノード稼働率を 1 カ月間計測した.

JSCAPS のパラメータについて, 最大停止ノード数を 30 とし, その他のパラメータは表 4 とした. JSS2 では会話型ジョブを優先した運用を行っているが, JSCAPS による電源停止により即時実行可能な待機ノード数が減少した場合, 会話型ジョブに影響が出る可能性が考えられる. そこで JSCAPS の最大停止ノード数を実行環境 PP のノード数より小さくすることで, 会話型ジョブが投入されても即時割り当て可能な待機ノードを確保する.

2018 年 4 月 3 日～30 日の計測結果を表 6 に示す.

表 6 実行環境 PP における省電力効果

	実行環境 PP 4/3～4/30
平均ノード稼働率	45.0%
システム全体の総消費電力の平均値	28.2kW
実行中ノードの総消費電力の平均値	16.4kW
全待機ノードの総消費電力の平均値	11.8kW
JSCAPS 未適用時に想定される 待機電力の平均値	17.0kW
待機電力の平均削減割合	待機電力を 31%削減

表 5 と比較し待機電力の削減割合が 95%から 31%に低下したことから, 1 カ月の計測期間では省電力効果が低くなる結果となった.

JSCAPS の適用による会話型ジョブへの影響について, JSCAPS 適用前(2018 年 3 月 1 日～31 日), 適用後(2018 年 4 月 3 日～30 日)の会話型ジョブの実行前キャンセル率を表 7 に示す. なお, JSCAPS 適用後の 4/10 は 1 ユーザが他の期間に比べ 10 倍近くの会話型ジョブを実行前にキャンセルしていたことから除外した.

表 7 会話型ジョブの実行前キャンセル率

	JSCAPS 適用前 3/1～3/31	JSCAPS 適用後 4/3～4/30(4/10 除く)
会話型ジョブの 実行前キャンセル率	0.8%	5.8%

会話型ジョブの wait-time は 0 秒としているため, 投入後に即時実行されない会話型ジョブは自動的にキャンセルされる. この結果, 会話型ジョブの実行前キャンセル率は増加した.

8. JSS2 の運用状況に合わせた JSCAPS のパラメータチューニング

8.1 JSS2 での省電力化の解析とパラメータチューニング

会話型ジョブの実行前キャンセル率が高かった 2018 年 4 月 23 日～29 日までのノード稼働率と消費電力の推移を図 4 に示す.

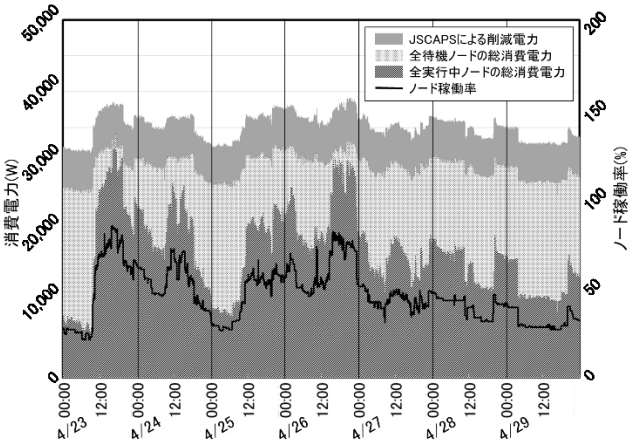

図 4 ノード稼働率と消費電力の推移(4/23～4/29)

実行環境 PP 内の全実行中ノードの総消費電力と全待機ノードの総消費電力を積み重ねた値が実行環境 PP で計測された全ノードの総消費電力となる. 実行環境 PP 内の全実行中ノードの総消費電力と全待機ノードの総消費電力, 削減電力を積み重ねた値が JSCAPS を適用しなかった場合に想定される実行環境 PP 全体の総消費電力となる.

図 4から, ノード稼働率が高い時間帯では待機ノード数が最大停止ノード数より少ない傾向にあり, 待機電力が十分削減されていることがわかる. また, ノード稼働率が低い時間帯では待機ノード数が最大停止ノード数より多い傾向にあり, 待機電力が十分削減されていないことから, 待機電力の削減余地がある.

図 5 は 2018 年 4 月 23 日～29 日までのバッチジョブが実行されているノード数と会話型ジョブが実行されているノード数の推移を示している.

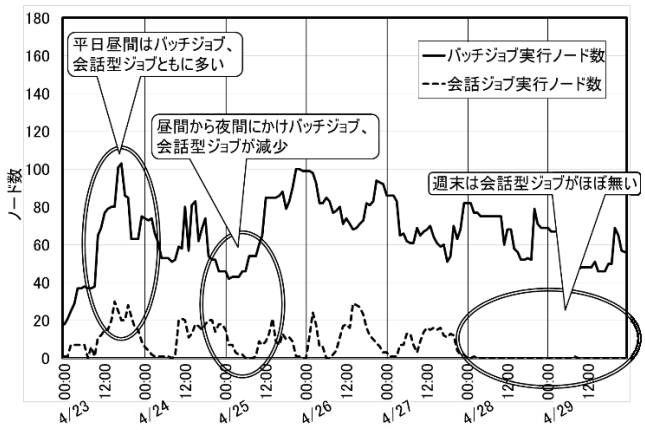

図 5 バッチジョブと会話型ジョブの実行状況

図 5 から, 平日昼間はバッチジョブ・会話型ジョブが共に多くノード稼働率が多い傾向にあるが, 夜間・週末は会話型ジョブ, バッチジョブが平日昼間に比べて少ない傾向にあることがわかる. したがって, 即時割り当て可能なノードを必要とする会話型ジョブが少なく, かつバッチジョブの要求ノード数も少ない夜間・週末に最大停止ノード数を増加することで, より待機電力を削減することができると考えられる. 最大停止ノード数は平日昼間と夜間・週末で, 表 8 のように切り替えることとする.

表 8 最大停止ノード数の切り替え

時間帯	最大停止ノード数
5:00～20:00	30
20:00～5:00	178

また, 表 7 のように会話型ジョブが実行を開始するまでの待ち時間に自動的にキャンセルされることを防ぐために, 会話型ジョブの wait-time を 1,200 秒とする.

8.2 JSCAPS パラメータチューニングの評価

8.1 節で述べたパラメータチューニングによる省電力効果と会話型ジョブへの影響の変化を確認する. 最大停止ノード数以外のパラメータは表 4 と同じとした.

表 9 に 2018 年 8 月 1 日～30 日における待機電力の削減割合と会話型ジョブの実行開始前キャンセル率を示す.

表 9 パラメータチューニング後の省電力効果と
会話型ジョブへの影響

	実行環境 PP 8/1～8/30
平均ノード稼働率	71.9%
システム全体の総消費電力の平均値	32.1kW
全実行中ノードの総消費電力の平均値	30.5kW
全待機ノードの総消費電力の平均値	1.6kW
JSCAPS 未適用時の 待機電力の平均値	8.2kW
待機電力の平均削減割合	待機電力を 80%削減
会話型ジョブの実行前キャンセル率	14.7%

表 6 と表 7 と比較すると, 待機電力の削減割合は 31%から 80%に増加し, 省電力効果は大幅に改善されたが, 会話型ジョブの実行前キャンセル率も 5.8%から 14.7%に増加し, 会話型ジョブへの影響はさらに大きくなった.

省電力効果が大きかった 2018 年 8 月 20 日～26 日までのノード稼働率と消費電力の推移を図 6 に示す.

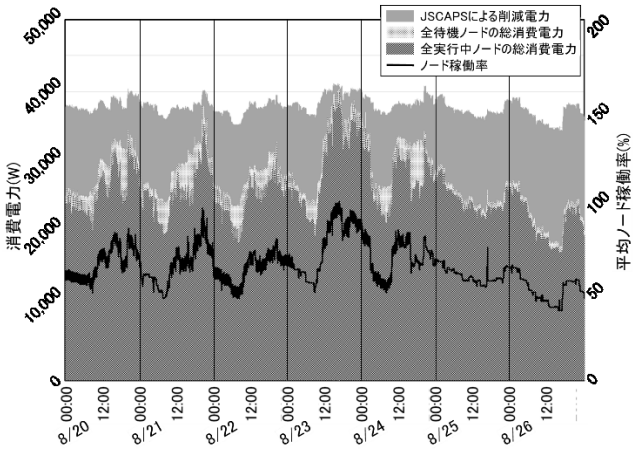

図 6 ノード稼働率と消費電力の推移(8/20～8/26)

図 4 と比較し, 全待機ノードの総消費電力がすべての期間において大幅に減少していることが確認できる.

早朝(5:00～8:00 ごろ)にかけて全待機ノードの総消費電力が他の時間帯よりも多くなっている. これは, 5:00 以降に最大停止ノード数が 30 になったが, ノード稼働率がまだ低いために電源が停止されなかった待機ノードが多くなっていることが原因と考えられる. また, 8/21 の午後(12:00～16:00 ごろ)や 8/24 の夕方(16:00～20:00 ごろ)にかけても全待機ノードの総消費電力が他の時間帯よりも多くなっている. この時間帯にノード稼働率が一時的に低下しているが, 最大停止ノード数に制限があるために待機ノードが停止されず, 全待機ノードの総消費電力が増加したと考えられる.

最大停止ノード数を切り替える時刻を調整することで, 早朝に増加する待機電力が削減されることが期待される. しかし一時的なノード稼働率の低下による待機電力の増加は毎日同じ時間帯におきるとは限らないために, ノード稼働率の状況に応じた最大停止ノード数をあらかじめ決めておくことは難しい.

2018 年 8 月 20 日～26 日までに投入された会話型ジョブが要求したノード数と, 会話型ジョブの実行前キャンセル率, 待機ノード数の時間帯ごとの変化を図 7 に示す.

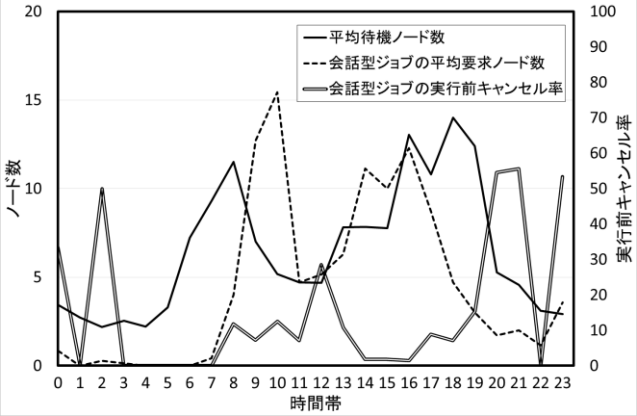

図 7 会話型ジョブと待機ノード数の時間帯ごとの状況

図 7 の横軸は 0 時～23 時までの時間帯を示している. 左の縦軸は 2018 年 8 月 20 日～26 日までに投入された会話型ジョブが要求したノード数と待機ノード数について, 時間帯ごとの平均値を示している. 右の縦軸は同様に, 2018 年 8 月 20 日～26 日までの会話型ジョブの実行前キャンセル率について時間帯ごとの平均値を示している.

会話型ジョブの要求ノード数は夜間より昼間が多くなる傾向が確認できる. 会話型ジョブの要求ノード数に対し, 即時割り当て可能な待機ノード数が十分でない場合, 会話型ジョブの実行開始までの待ち時間が発生すると考えられる. 最大停止ノード数を昼間に少なくすることで割り当て可能な待機ノード数が増加し, 会話型ジョブの実行前キャンセル率も低下したと考えられる. ただし 10:00～12:00 の時間帯では待機ノード数が一時的に減少していることから, この時間帯は会話型ジョブの実行前キャンセル率も増加した.

会話型ジョブの wait-time を 1,200 秒に設定していたことから, 会話型ジョブが実行前にキャンセルされた原因として, 実行開始前の待ち時間が 1,200 秒以上経過したか, 待ち時間が長すぎてユーザ自身の手でキャンセルされたかのいずれかが考えられる. 会話型ジョブのために待機ノードを十分な数に保つことで実行前キャンセル率が低下すると考えられるが, その分省電力効果が低下するため, 運用状況にあわせて適切なパラメータを設定する必要がある.

9. 今後の取り組み

JSS2 への JSCAPS の適用を通じて, 待機電力の削減割合と会話型ジョブの実行前キャンセル率を最適化するための課題を以下に挙げる.

- 多くのユーザが利用中の実環境下でパラメータチューニングを行うために, 慎重な試行錯誤と長い時間が必要となる
- 時間帯によるパラメータの変更だけでは, 一時的なノード稼働率の低下などの運用状況の急な変化に対応できない

これらの問題を解決する施策として, 以下について今後検討を行う.

- 効率的な試行錯誤を行うための, より実環境に近いシミュレータの導入
- 運用状況の変化に迅速に対応するために, JSCAPS のパラメータ変更をリアルタイムにフィードバックできる仕組みの追加や, 会話型ジョブの投入履歴をもとにした投入予測機構の追加

これらの施策をもとに, より実環境に適した使いやすい JSCAPS の提供を行う予定である.

10. おわりに

本稿では JSCAPS を JAXA スーパーコンピュータ JSS2 の実行環境 PP に適用し, 実環境における省電力効果を評価した. バッチジョブを中心とした実環境において JSCAPS を適用した結果, 待機電力の平均削減割合は 95%となった. この結果, シミュレーションと同様に, JSCAPS は実環境においても高い省電力効果を得られることが確認された. また実行環境 PP において省電力効果と会話型ジョブへの影響を長期的に計測した結果, JSCAPS のパラメータである最大停止ノード数をノード稼働率や会話型ジョブの投入状況に合わせて調整することで, 待機電力の削減割合が 80%となることを確認した. ただし会話型ジョブの実行前キャンセル率は 14.7%に増加したことから, 待機電力の削減割合と会話型ジョブの実行前キャンセル率はトレードオフとなり, 運用状況に合わせて適切なパラメータを設定することが必要であることが確認された.

今後は JSCAPS が多様な実環境下で容易に最適な省電力効果が得られるように, パラメータチューニングのためのシミュレータやリアルタイムに運用状況をフィードバックできる仕組みなどの導入を進める予定である.

謝辞 本研究における省電力の検証結果は, JAXA スーパーコンピュータ"JSS2"を利用して得られたものである.

参考文献

[1] "Slurm Power Saving Guide", https://slurm.schedmd.com/power_save.html.
[2] "IBM Spectrum LSF", https://www.ibm.com/support/knowledgecenter/en/SSWRJV_10.1.0/lsf_welcome/lsf_kc_eas.html.
[3] 今出広明,加賀美崇紘,三浦健一,井口裕次,坂口吉生,藤田直行: "PC クラスタにおける省電力化のための自動電源制御", 研究報告ハイパフォーマンスコンピューティング（HPC）, 2017-HPC-160,(2017).
[4] "JSS2@JAXA", https://www.jss.jaxa.jp.