Vol.2016-HPC-156 No.9
2016/9/16

配分資源量に基づくフェアシェア機能の提案

中島 真[1]　岩田 章孝[1]　関澤 龍一[1]　宇野 篤也[2]　山本 啓二[2]

概要：多数の利用者が共用利用するスーパーコンピュータにおける計算資源の配分方法として，利用者ごとに決められた資源量を配分する方式がある．一般的なスケジューリング機能であるフェアシェア機能は，計算資源の使用量に基づいてジョブ優先度制御を行い利用者間の公平な資源使用を実現する機能であるが，配分資源量は考慮していない．そのため，一定のペースで計画的に計算資源を使用している利用者のジョブの実行が，それまで計算資源を使用していない利用者が集中的に投入したジョブにより阻害されてしまうといった問題が起こりうる．この問題を解決するためには，すべての利用者が配分資源量に応じた計算資源を一定期間内に使用できるようにする制御が有効と考える．本稿では，配分資源量に基づいてジョブ優先度制御を行うフェアシェアの方式である Planned Use を提案し，その有効性について検証した結果を報告する．

1. はじめに

幅広い実アプリケーションでの高い実行性能の実現を目指して開発されたスーパーコンピュータ「京」(以下,「京」) は 2012 年 9 月から共用が開始され,「生命科学・医療」,「エネルギー」,「防災・減災」,「次世代ものづくり」,「物質と宇宙」の戦略 5 分野を中心とした幅広い分野において質の高い研究成果を生み出している [1], [2].「京」に限らず，スーパーコンピュータを用いたシミュレーション技術は学術利用から産業活用まで幅広い分野に広がり，高い計算能力を有するスーパーコンピュータの需要は高まり続けている．

高い計算能力を有するスーパーコンピュータの設置・維持には多額の設備投資・運用コストの負担が必要であり，大学，研究機関，企業等が運営するスーパーコンピュータを多数の利用者が共用利用する形態が一般的である．一部のスーパーコンピュータ運用機関では，スーパーコンピュータにより得られる成果の最大化を目的として，計算資源の利用目的やその社会的意義などに基づいた審査を行い，個々の利用者に対して年間・半期など一定の期間（以下，資源配分期間）中に使用可能な計算資源量（以下，配分資源量）を割り当てる運用を行っている [3], [4].

共用利用されるスーパーコンピュータではバッチスケジューラを利用することが多く，個々の利用者はアプリケーションが行う一連の処理をジョブと呼ばれる単位にまとめ，当該ジョブの実行のために必要となる計算資源[*1]の量（以下，要求資源量）を指定してバッチスケジューラへと投入する．ジョブの要求資源量は，スーパーコンピュータの提供するすべての計算資源を長時間専有するものから，ごく少数の計算資源を短時間専有するものまで多岐にわたる．

ジョブの要求を満たすだけの計算資源が空いている場合，ジョブは投入後すぐに実行されるが，当該ジョブ以外に多数のジョブが投入・実行されている場合などすぐに利用可能な計算資源が不足している場合は他のジョブの実行完了まで実行が待たされる．このとき，実行が待たされているジョブの実行順序はジョブの優先度と呼ばれる指標で決定され，高優先度のジョブほど早く実行される．利用者にとっては自身が投入したジョブの優先度が常に高い状態となり可能な限り早く実行されるのが望ましいが，共用利用されるスーパーコンピュータではすべての利用者のすべてのジョブを即時に実行することはできない．このため，利用者の納得できる何らかの方法により，個々のジョブの実行順序（優先度）を適切に決定しなければならない．例えば，年間の計算資源量を個々の利用者に配分する運用においては，個々の利用者は配分された計算資源量に応じた計算資源の使用権利を常に有すると考えることができ，この場合は過去の計算資源使用量に関係なく常にある計算資源量までは適切な優先度でジョブが実行されることになる．

多くのバッチスケジューラでは個々の利用者の計算資

[1] 富士通株式会社
FUJITSU LIMITED
[2] 理化学研究所計算科学研究機構
RIKEN Advanced Institute for Computational Science

[*1] バッチスケジューラの種別や，運用によっても異なるが，ノード数，CPU 数，予測実行時間などが代表的

源使用の公平化を実現するためにフェアシェアと呼ばれる機能が実装されている [5], [6], [7]．従来のフェアシェアは，個々の利用者の過去の計算資源使用量を比較し，使用量が少ない利用者のジョブの優先度を高くする制御により計算資源使用の公平化を実現している．しかし，個々の利用者の使用可能な計算資源量に上限（配分資源量）が決められている運用においては，過去に使用した計算資源量に基づく制御では計画的に計算資源を使用する利用者にとって不利になる場合がある．そこで，本稿では配分資源量に基づいたジョブの優先度制御を実現するフェアシェア機能 (Planned Use) を提案する．

2. 配分資源量を割り当てる運用

本章では利用者にとっての使い勝手の良さという観点および，個々の利用者の計算資源使用量が配分資源量に基づいて適切に調整されるかという観点から，配分資源量を割り当てる運用における，バッチスケジューラのジョブ優先度制御のあり方について述べる．

本章以降では，バッチスケジューラが行うジョブ実行開始処理・終了処理等にかかる時間はジョブの実行時間と比較すると十分小さく無視できるものとみなす．また，ジョブの要求資源量の単位はノード時間積とし，特別な注記がない限り「資源」は「計算資源」を意味するものとする．ノード時間積とは，ジョブが要求するノード数とジョブの実行時間の積で表される量であり，単位は [ノード · 時] または [ノード · 日] である．例えば，5 ノードで 3 日間にわたり実行されるジョブの要求資源量は 15[ノード · 日] である．ジョブの要求資源量の単位として要求 CPU 数と実行時間の積である CPU 時間積を用いる場合でも，ノードを CPU と置き換えることで本稿と同様の議論が成立する．

2.1 常用資源量

1 章で述べた通り，多数の利用者で共用利用されるスーパーコンピュータの運用においては，常にすべての利用者のすべてのジョブを優先的に実行することはできない．しかし，資源を利用者間で配分する運用では，個々の利用者が一定量の資源を常に使用する権利を持つと考えることができ，権利の範囲内ならいつでもジョブが実行できる（資源を使用できる）ような制御が行えることが理想である．例えば，利用者自身がコントロールできない要因（他の利用者による大量のジョブ投入など）により投入したジョブの実行が大きく遅れてしまうこと，特に権利の範囲内で毎日ジョブ投入している利用者のジョブは毎日実行されるべきであるにも関わらず，全くジョブが実行されない日が生じることは望ましくない．また，ある利用者が過去に権利の範囲を越えて資源を使用していたとしても，直近（例えば過去数日間）で権利の範囲を越えていないのであれば，当該利用者は権利の範囲内の資源は使用できることが望ましい．

以下では，個々の利用者が常に使用する権利を持つ資源量を常用資源量と呼ぶ．常用資源量は，個々の利用者が時間あたりに使用可能な資源量を意味しており，単位は資源量（ノード時間積）の単位（[ノード · 日] または [ノード · 時]）を時間（[日] または [時]）で割った [ノード] である．ある利用者の常用資源量が 100[ノード] の場合，1 日あたり 100[ノード · 日] だけの資源量がいつでも使用可能であり，要求資源量が 200[ノード · 日] のジョブを投入するのであれば，最低限 2 日に 1 回は実行可能であることを意味する．

すべての利用者が常用資源量分の資源を使用できるようにするためには，個々の利用者の常用資源量の総和がスーパーコンピュータの時間あたりに提供する資源量以下でなければならない．配分資源量を割り当てる運用では，個々の利用者の常用資源量 R_i の総和がスーパーコンピュータの時間あたりに提供する資源量と等しくなり，また R_i の利用者間での比率が，配分資源量の利用者間での比率と等しくなるように，利用者 i の 常用資源量 R_i を以下のように設定する．

$$R_i = \frac{S_i}{\sum_{j \in I} S_j} N \tag{1}$$

ここで I はすべての利用者の集合，S_i は利用者 i の配分資源量， N はスーパーコンピュータの提供する資源の数である．配分資源量の総和 $\sum_{j \in I} S_j$ が NT （T は資源配分期間）と等しい場合，すなわち資源配分期間で利用可能なすべての資源量を全利用者で重複なく分配する運用の場合，$R_i = \frac{S_i}{T}$ となる．

2.2 遊休資源

ある利用者が常用資源量未満の資源を使用し，その他の利用者が常用資源量分の資源を使用する場合には，どのジョブも実行されない資源（遊休資源）が生じることになる．遊休資源が存在するのであれば，他の利用者（常用資源量分の資源を使用する利用者）に当該資源を割り当て，遊休資源を有効活用するべきである．

複数の利用者が遊休資源を使用する場合，各利用者への遊休資源の割り当て方式は，各利用者のジョブのうち投入時刻が早いジョブへと優先的に割り当てる方式，個々の利用者が時間あたりに使用する遊休資源の比率を常用資源量の比率と等しくなるように割り当てる方式，配分資源量の残量が多い利用者を優先する方式などが考えられる．それぞれの方式にはメリット・デメリットが存在するため，スーパーコンピュータの運用者が適したものを選択できることが望ましい．

本稿では，遊休資源が特定の利用者に偏って割り当てられないよう，個々の利用者が時間あたりに使用する遊休資源の比率が常用資源量の比率と等しくなるように割り当てる方式を考える．例えば，利用者 a, b, c が存在し，常用資

源量がそれぞれ R_a, R_b, R_c であるとする．すべての利用者が十分多くのジョブを投入した場合，利用者 a, b, c が時間あたりに使用する資源量はそれぞれ R_a，R_b，R_c となる．また，利用者 a, b が十分多くのジョブを投入し，利用者 c がジョブを投入しなかった場合，利用者 a, b が時間あたりに使用する資源量はそれぞれ $R_a + \frac{R_a}{R_a+R_b}R_c = \frac{R_a}{R_a+R_b}N$ および $R_b + \frac{R_b}{R_a+R_b}R_c = \frac{R_b}{R_a+R_b}N$ となる．

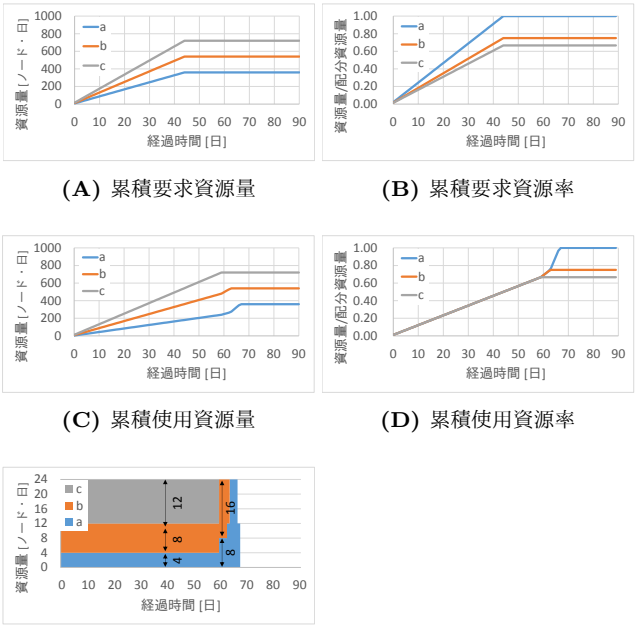
2.3 運用方針の実施例

2.1 節 および 2.2 節 で述べた運用方針について，具体的な例を用いて説明する．以下のような運用形態・投入ジョブを考える．

- 計算ノード数: 24[ノード]
- 資源配分期間: 90[日]
- 利用者: a, b, c の 3 名
- 配分資源量（括弧内は常用資源量）：
  - 利用者 a: 4×90[ノード · 日]（4[ノード]）
  - 利用者 b: 8×90[ノード · 日]（8[ノード]）
  - 利用者 c: 12×90[ノード · 日]（12[ノード]）
- 投入ジョブ（括弧内はジョブ投入開始から投入完了までの時間あたりの要求資源量）：
  - 利用者 a: 資源配分期間の初めから 45 日間，要求ノード数 4, 実行時間 1 日のジョブを 1 日に 2 回投入（8[ノード · 日]）
  - 利用者 b: 資源配分期間の初めから 45 日間，要求ノード数 4, 実行時間 1 日のジョブを 1 日に 3 回投入（12[ノード · 日]）
  - 利用者 c: 資源配分期間の初めから 45 日間，要求ノード数 4, 実行時間 1 日のジョブを 1 日に 4 回投入（16[ノード · 日]）

ジョブ投入の様子を示しているのが**図 1(A)** および**図 1(B)** である．図 1(A) は各利用者が投入したジョブの要求資源量の累積値（累積要求資源量）の時間変化である．図 1(B) は累積要求資源量を各利用者の配分資源量で割ることで正規化したもの（累積要求資源率）である．利用者 a が投入するジョブの総要求資源量は配分資源量と同じであり累積要求資源率は 1.0 に到達しているが，利用者 b, c については投入するジョブの総要求資源量は配分資源量より少なく，累積要求資源率は 1.0 より小さい値になる．

上記ジョブ投入に対し，2.1 節，2.2 節で述べた理想的な状況となるような順序でジョブ実行された場合の各利用者が使用する資源の状況を示しているのが**図 1(C)**，**図 1(D)** および**図 1(E)** である．図 1(C) は各利用者の実行されたジョブの使用した資源量の累積値（累積使用資源量）の時間変化である．図 1(D) は累積使用資源量を各利用者の配分資源量で割ることで正規化したもの（累積使用資源率）である．図 1(E) は期間中の各日に使用された資源量（日別使用資源量）の推移である．



(A) 累積要求資源量　**(B)** 累積要求資源率

(C) 累積使用資源量　**(D)** 累積使用資源率

(E) 日別使用資源量

図 1: 配分資源量に基づく使用資源量制御

資源配分期間の初めから 59 日目までは，利用者 a, b, c のいずれも実行待ちのジョブが存在し，個々の利用者の時間あたりの使用資源量は常用資源量（a: 4[ノード], b: 8[ノード], c: 12[ノード]）と一致しており，配分資源量に基づいてフェアに使用されている（図 1(E)）．時間あたりの使用資源量は累積使用資源量の傾きとしてグラフ中に表現されており，利用者 a, b, c の傾きの比率は時間あたりの使用資源量の比率と同じく 1:2:3 となる（図 1(C)）．配分資源量に基づいてフェアに使用されている状態は，すべての利用者の累積使用資源率の傾きが同じになることでも表現される（図 1(D)）．59 日目までの期間は，すべての利用者が常用資源量分の資源を使い切っているため遊休資源は存在せず，常用資源量より多くの資源は使用されない．

利用者 c のジョブは 59 日目までにすべて終了するため，60 日目から 62 日目までは利用者 a, b のジョブのみが実行される．このため，利用者 a, b が常用資源量分の資源を使い切っても利用者 c の常用資源量分の資源が遊休資源として存在する．利用者 a, b の常用資源量の比率に応じて遊休資源を配分した結果，両利用者の時間あたりの使用資源量は，両利用者の常用資源量の比率と同じく 1:2 となり，配分資源量に基づいてフェアに利用されている（図 1(E)）．59 日目までと同様，配分資源量に基づいてフェアに利用されている状態は， 利用者 a, b の累積使用資源率の傾きが同じになることでも表現される（図 1(D)）．

利用者 b のジョブは 62 日目までにすべて終了するため，63 日目以降は利用者 a のジョブのみ実行され，すべての資源を利用者 a が使用する（図 1(E)）．

3. 従来のフェアシェアの概要と課題

一般的なフェアシェアでは，個々の利用者に対して，過去の使用資源量より算出される指標（以下，資源使用指標）に基づいて優先度が算出される．ジョブの実行順序は利用者の優先度に基づいて，優先度の高い利用者のジョブがより早く実行されるように，また同一利用者により投入されたジョブ同士では投入時刻が早いものがより早く実行されるように決定される．

フェアシェアには資源使用指標の算出方法として，例えば Technical Computing Suite[*2] のバッチスケジューラが採用する方式（以下，線形減衰型のフェアシェア）や Slurm[5], Platform LSF[6], PBS Professional[7] が採用する方式（以下，指数減衰型のフェアシェア）などが存在する．本章では，これらのアルゴリズムにおいて，配分資源量を割り当てる運用を行った場合に生じる課題について述べる．

3.1 線形減衰型のフェアシェア

線形減衰型のフェアシェアは，資源使用指標が時間経過に比例した量だけ減衰するアルゴリズムである．本アルゴリズムの概要，性質および配分資源量を割り当てる運用における課題について述べる．

3.1.1 線形減衰型のフェアシェアのアルゴリズム

線形減衰型のフェアシェアにおいては，利用者 i の優先度 $p_{1,i}$ は，資源使用指標 $u_{1,i}$ により以下のように決定される．

$$p_{1,i} = -u_{1,i} \tag{2}$$

$u_{1,i}$ は利用者 i が資源を使用するのに従い大きくなるため，資源を多く使用する利用者の優先度は低くなる（資源使用指標に逆比例する）．

$u_{1,i}$ は以下のルールにより算出される．

- 初期値 $u_{1,i} = 0$
- 利用者 i の投入したジョブが実行開始される時点で，$u_{1,i}$ に $r/W_{1,i}$ を加算する．ここで，r は当該ジョブの要求資源量，$\frac{1}{W_{1,i}}$ はあらかじめ決められた利用者 i が投入したジョブの使用資源量に対する重みである．
- 単位時間経過ごとに利用者ごとにあらかじめ決められた減衰量 $D_{1,i}$ （$D_{1,i} > 0$）だけ $u_{1,i}$ から減算する．減算した結果 $u_{1,i}$ が負になった場合 $u_{1,i}$ は初期値 0 を設定する．

システム運用者が設定するパラメータは，$W_{1,i}$ および $D_{1,i}$ である．時間あたりの使用資源量が常用資源量 R_i 以下となる利用者のジョブを優先するためには，最高優先度 $p_{1,i} = 0$ である利用者がある時点で $R_i t$ だけの資源を使用した場合，時間 t 経過時点で再び最高優先度になればよいため，

$$\frac{R_i t}{W_{1,i}} - D_{1,i} t = 0 \tag{3}$$

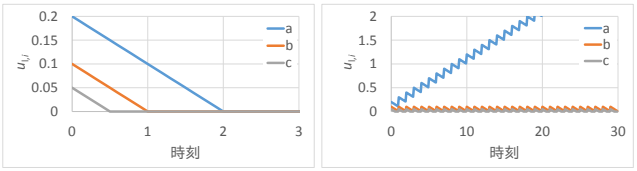
$$\Leftrightarrow W_{1,i} D_{1,i} = R_i \tag{4}$$

を満たすよう $W_{1,i}$, $D_{1,i}$ を設定すれば良い．また，個々の利用者の時間あたりの使用資源量の比率が常用資源量の比率と同じになるようにするためには，すべての利用者が常用資源量の比率に応じた資源量 $R_i s$ （s は任意の正の数）を使用した場合の $u_{1,i}$ の増加量が全利用者で同じになれば良いため，

$$\frac{R_i s}{W_{1,i}} = \frac{R_j s}{W_{1,j}} \quad \forall i, j \in I \tag{5}$$

$$\Leftrightarrow W_{1,i} = k R_i \tag{6}$$

を満たすよう $W_{1,i}$ を設定すれば良い．ここで，k は適当な正の数である．



(A) 単一ジョブ実行時　　**(B)** 複数ジョブ実行時

図 2: 線形減衰型のフェアシェアでの資源使用指標の推移

3.1.2 線形減衰型のフェアシェアの性質

ジョブの実行により $u_{1,i}$ の値がどのように変化するかについて述べる．

$u_{1,i}$ はジョブが実行されると増加し，時間とともに一定のペースで 0 へ近づいていく．時間あたりの減少量は $D_{1,i}$ であり，その後ジョブが実行されなければ $u_{1,i}$ に比例した時間（$u_{1,i}/D_{1,i}$）が経過した時点で 0 に戻る．単一のジョブが実行された場合の各利用者の $u_{1,i}$ の時間遷移を **図 2(A)** に示す．この図は利用者 a, b, c が投入したジョブが時刻 0 で 1 つだけ実行された場合を示している．各利用者のジョブの要求資源量はそれぞれ $2W_{1,\mathrm{a}}D_{1,\mathrm{a}}$, $W_{1,\mathrm{b}}D_{1,\mathrm{a}}$, $0.5W_{1,\mathrm{c}}D_{1,\mathrm{a}}$ であり，$D_{1,i}$ は全利用者で 0.1 と設定したため，実行開始時（時刻 0）に $u_{1,i}$ はそれぞれ 0.2, 0.1, 0.05 だけ増加する．その後は単位時間経過ごとに $u_{1,i}$ は 0.1 ずつ減少する．

要求資源量の同じジョブが一定間隔で繰り返し実行される場合，時間あたりに実行されるジョブの要求資源量の大小により $u_{1,i}$ のふるまいが変化する．時間あたりに実行されるジョブの要求資源量が $W_{1,i}D_{1,i}$ 以下の場合，$u_{1,i}$ はジョブ実行開始時に一時的に増加するが時間経過とともに時間あたりの減衰量 $D_{1,i}$ だけ減少していくことで 0 に戻り，これを繰り返すことで 0 付近を振動する．時間あたりの使用資源量が $W_{1,i}D_{1,i}$ を超過する場合も $u_{1,i}$ はジョブ実行開始時に増加し時間とともに減少するが，時間あたりの増加量が減少量よりも大きくなるため 0 までは戻らず，

[*2] 富士通の開発した HPC 向けミドルウェア

細かな振動を繰り返しながら一定のペースで増加していく．要求資源量が同一のジョブが一定の間隔で実行された場合の各利用者の $u_{\mathrm{l},i}$ の時間遷移を 図 **2(B)** に示す．利用者 a, b, c はそれぞれ要求資源量 $2W_{\mathrm{l,a}}D_{\mathrm{l,a}}$，$W_{\mathrm{l,b}}D_{\mathrm{l,b}}$，$0.5W_{\mathrm{l,c}}D_{\mathrm{l,c}}$ のジョブを時刻 0 から単位時間間隔で実行している．$u_{\mathrm{l,a}}$ は時間経過とともに増加し続けるが，$u_{\mathrm{l,b}}$ および $u_{\mathrm{l,c}}$ は 0 付近で振動し続ける．

3.1.3 線形減衰型のフェアシェアの課題

線形減衰型のフェアシェアでは，2 章で述べた運用方針を実現することはできない．過去に遊休資源を多く使用した場合（時間あたりの使用資源量が常用資源量を超過し続けた場合），新たに資源を使用することができず常用資源量分の資源を使用することができないという課題が存在するためである．$u_{\mathrm{l},i}$ が 0 以上の状態から 0 に戻る，すなわち優先度が低い状態から再び優先度が高い状態に戻るまでには $u_{\mathrm{l},i}$ に比例した時間がかかるため，過去に遊休資源を多く使用した利用者は $u_{\mathrm{l},i}$ の値は長時間大きい状態が続く．例えば，過去の 1 日間の時間あたりの使用資源量が常用資源量の 100 倍であった場合，優先度が 0 に戻るまで 100 日を要する．このような場合，当該利用者の優先度は相対的に低い状態が続くため，常用資源量分の資源を使うことができない．これでは直近（過去数日間）の資源使用量が常用資源量以下の場合なら少なくとも常用資源量分の資源が使用できることが望ましいという 2.1 節で述べた運用方針を実現することができない．

3.2 指数減衰型のフェアシェア

指数減衰型のフェアシェアは，資源使用指標が指数的に減衰するアルゴリズムである．本アルゴリズムの概要，性質および配分資源量を割り当てる運用における課題について述べる．

3.2.1 指数減衰型のフェアシェアのアルゴリズム

指数減衰型のフェアシェアにおいては，利用者 i の優先度 $p_{\mathrm{e},i}$ は，資源使用指標 $u_{\mathrm{e},i}$ により以下のように決定される．

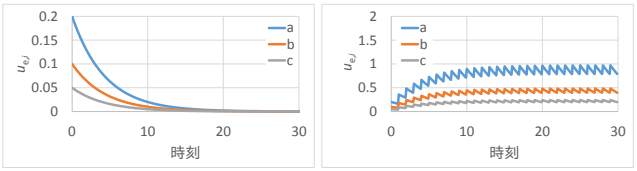
$$p_{\mathrm{e},i} = -u_{\mathrm{e},i} \tag{7}$$

優先度 $p_{\mathrm{e},i}$ と資源使用指標 $u_{\mathrm{e},i}$ は線形減衰型のフェアシェアと同じく逆比例の関係にあり，計算資源を多く使用する利用者の優先度は低くなる．

$u_{\mathrm{e},i}$ は以下の式で定義される．

$$\begin{aligned} u_{\mathrm{e},i} &= \frac{1}{W_{\mathrm{e},i}}\left\{s_i(n) + D_{\mathrm{e}}s_i(n-1) + D_{\mathrm{e}}^2 s_i(n-2) + \dots\right\} \\ &= \frac{1}{W_{\mathrm{e},i}}\sum_{m=0}^{n} D_{\mathrm{e}}^m s_i(n-m) \end{aligned} \tag{8}$$

この式では，時刻を 0 から Δt 間隔で複数の区間 $T_m = [m\Delta t, (m+1)\Delta t)$ に分割し，個々の区間の使用資源量に対して重みと減衰係数を乗算することで資源使用指標を定義している．$\frac{1}{W_{\mathrm{e},i}}$ は利用者 i の使用資源量に対する重み，n は現在時刻を含む区間（T_n）の番号，$s_i(m)$ は区間 T_m に実行開始された利用者 i のジョブの要求資源量の総和，D_{e} は区間あたりの減衰係数（$0 < D_{\mathrm{e}} < 1$）である．システム運用者が設定するパラメータは，$W_{\mathrm{e},i}$, D_{e} および Δt である．



(A) 単一ジョブ投入時　　**(B)** 複数ジョブ投入時

図 3: 指数減衰型のフェアシェアでの資源使用指標の推移

個々の利用者の時間あたりの使用資源量の比率が常用資源量の比率と同じになるようにするためには，すべての利用者が常用資源量の比率に応じた資源量 $R_i s$（s は任意の正の数）を使用した場合の $u_{\mathrm{e},i}$ の増加量が全利用者で同じになれば良いため，

$$\frac{R_i s}{W_{\mathrm{l},i}} = \frac{R_j s}{W_{\mathrm{l},j}} \quad \forall i, j \in I \tag{9}$$

$$\Leftrightarrow W_{\mathrm{e},i} = kR_i \tag{10}$$

を満たすよう $W_{\mathrm{e},i}$ を設定すれば良い．ここで，k は適当な正の数である．

3.2.2 指数減衰型のフェアシェアの性質

ジョブの実行により $u_{\mathrm{e},i}$ の値がどのように変化するかについて述べる．

線形減衰型のフェアシェアの $u_{\mathrm{l},i}$ と同じく，$u_{\mathrm{e},i}$ はジョブが実行されると増加し，その後時間経過とともに 0 へと近づいていく．単一のジョブが実行された場合の各利用者の $u_{\mathrm{e},i}$ の時間遷移を 図 **3(A)** に示す．この図は利用者 a, b, c が投入したジョブが時刻 0 で 1 つだけ実行された場合を示している．各利用者のジョブの要求資源量はそれぞれ $0.2W_{\mathrm{e,a}}$, $0.1W_{\mathrm{e,b}}$, $0.05W_{\mathrm{e,c}}$ であり，実行開始時（時刻 0）に $u_{\mathrm{e},i}$ はそれぞれ 0.2, 0.1, 0.05 だけ増加する．D_{e} は $10^{-1/10}$ と設定したため，時刻が 10 経過するごとに $u_{\mathrm{e},i}$ は 1/10 の値になる．

要求資源量の同じジョブが一定間隔で繰り返し実行される場合，$u_{\mathrm{e},i}$ はジョブ実行開始時の増加とその後の減衰により細かな振動を繰り返しながら，時間あたりの使用資源量に比例する一定値の付近で振動するようになる．各区間ごとに $xW_{\mathrm{e},i}$ だけの資源量を使用する場合（$xW_{\mathrm{e},i} = s_i(0) = s_i(1) = s_i(2) = \dots$），$u_{\mathrm{e},i} = \frac{x}{1-D_{\mathrm{e}}}$ となる．要求資源量の同じジョブが一定間隔で実行された場合の $u_{\mathrm{e},i}$ の時間遷移を図 **3(B)** に示す．利用者 a, b, c はそれぞれ要求資源量 $0.2W_{\mathrm{e,a}}$, $0.1W_{\mathrm{e,b}}$, $0.05W_{\mathrm{e,b}}$ のジョブを時刻 0 から Δt 間隔で実行している．それぞれ振動しながら $\frac{0.2}{1-D_{\mathrm{e}}} \simeq 0.97$, $\frac{0.1}{1-D_{\mathrm{e}}} \simeq 0.49$, $\frac{0.05}{1-D_{\mathrm{e}}} \simeq 0.24$ へと漸近して

いる．

資源使用指標 $u_{\mathrm{e},i}$ は指数的に減衰するため，線形減衰型のフェアシェアにあったような，過去に遊休資源を多く使用した利用者の優先度が再び高優先度に戻るまでに時間がかかる課題は生じない．これは，線形減衰型のフェアシェアでは $u_{\mathrm{l},i}$ の増加後，元の値に戻るまでに増加量に比例した時間（1 から x へ増加した場合 $\frac{x-1}{D_{\mathrm{l},i}}$）がかかるのに対し，指数減衰型のフェアシェアでは増加量の対数に比例した時間（1 から x へ増加した場合 $-\frac{\log x}{\log D_{\mathrm{e}}}$）しかかからないためである．

3.2.3 指数減衰型のフェアシェアの課題

指数減衰型のフェアシェアでは，2 章で述べた運用方針を実現することはできない．ある利用者の過去の使用資源量が少ない場合に，他の利用者が資源を使用することができず，常用資源量分の資源を使用することができないという課題が存在するためである．例えば，時間あたりの使用資源量が常用資源量と等しい利用者 a と，常用資源量よりも少ない利用者 b が存在したとする．利用者 a の $u_{\mathrm{e,a}}$ はある正の数に収束しており利用者 b よりも優先度が低い状態にあり，この状態で利用者 b がジョブを多く投入した場合，利用者 b の優先度が利用者 a の優先度を下回るまでは利用者 b のジョブが実行され続ける．特に，過去一度も資源を使用しなかった利用者は最も高い優先度になるため，当該利用者がジョブを大量投入すると他の利用者のジョブ実行が妨げられる．これでは常用資源量の範囲内で毎日ジョブ投入している利用者のジョブは毎日実行されることが望ましいという 2.1 節で述べた運用方針を実現することができない．

4. Planned Use

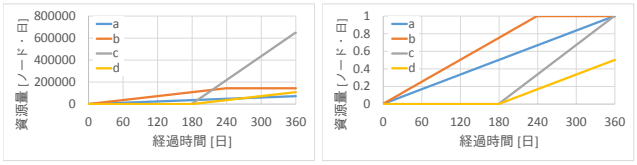
2 章で述べた運用方針を実現するための方式として Planned Use を提案する．Planned Use では，利用者 i の優先度 $p_{\mathrm{p},i}$ は，資源使用指標 $u_{\mathrm{p},i}$ により以下のように決定する．

$$p_{\mathrm{p},i} = \begin{cases} 0 & \text{if } u_{\mathrm{p},i} \leq 1 \\ 1 - u_{\mathrm{p},i} & \text{otherwise} \end{cases} \tag{11}$$

$u_{\mathrm{p},i}$ が 1 以下の場合，優先度は最高 (0) になり，1 より大きくなるに従い優先度が低下していく．複数の利用者の優先度が同じ場合，FCFS (first-come first-served) と同じく当該利用者のジョブの中から投入時刻が最も早いものを選択する．

$u_{\mathrm{p},i}$ は以下の式で定義する．この式は，指数減衰型フェアシェアの $u_{\mathrm{e},i}$ に対し，時間あたりの使用資源量が常用資源量 R_i と一致し続けた場合，すなわち区間あたりの使用資源量が $R_i\Delta t$ となる場合の収束値が 1 になるよう，$W_{\mathrm{e},i} = \frac{R_i\Delta t}{1-D_{\mathrm{p}}}$ を代入したものである．

$$u_{\mathrm{p},i} = \frac{1-D_{\mathrm{p}}}{R_i\Delta t} \sum_{m=0}^{n} D_{\mathrm{p}}^m s_i(n-m) \tag{12}$$



(A) 累積要求資源量　　**(B)** 累積要求資源率

図 4: シミュレーション 1: 累積要求資源量・率の推移

D_{p} は区間あたりの減衰係数（$0 < D_{\mathrm{p}} < 1$）である．システム運用者が設定するパラメータは，D_{p} および Δt である．

本方式では，時間あたりの使用資源量が常用資源量以下である利用者の資源使用指標は $u_{\mathrm{p},i} \leq 1$ となり，優先度は常に最高優先度（$p_{\mathrm{p},i} = 0$）となる．線形減衰型のフェアシェアで生じる，過去に多くの資源を使用した利用者の優先度が再び高優先度に戻るまでに時間がかかる課題は，$u_{\mathrm{p},i}$ を指数減衰型のフェアシェアと同じ方法で算出するため 3.2.2 項で述べたのと同じ理由により生じない．指数減衰型のフェアシェアで生じる，使用資源量の少ない利用者が優先されてしまう課題についても，時間あたりの使用資源量が常用資源量以下であるすべての利用者が最高優先度となり使用資源量の少ない利用者が過度に優遇されることはないため，生じない．

常用資源量を越えて遊休資源を使用する場合，すなわち $u_{\mathrm{p},i} > 1$ の場合は，指数減衰型のフェアシェアと同じふるまいになる．$W_{\mathrm{e},i} = \frac{\Delta t}{1-D_{\mathrm{p}}} \cdot R_i$ であることから (10) の条件に合致するため，利用者ごとの総使用資源量の比率は R_i の比率と一致する．

5. 評価

本章では，2 章で述べた運用方針の実現性について，従来の線形減衰型および指数減衰型のフェアシェアと，本稿で提案した Planned Use を用いてシミュレーションにより評価した．

今回のシミュレーションでは，各方式のパラメータには 3.1.1 項，3.2.1 項，4 章で述べた条件を満たすよう以下の値を設定した．単位時間は 1 日である．

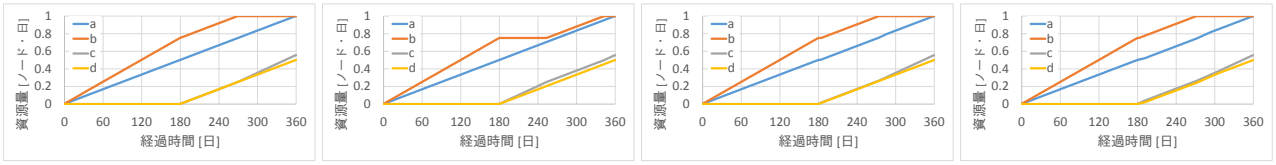
- 線形減衰型のフェアシェア: $W_{\mathrm{l},i} = R_i$, $D_{\mathrm{l},i} = 1$
- 指数減衰型のフェアシェア: $W_{\mathrm{e},i} = R_i$, $D_{\mathrm{e}} = 10^{-1/15}$, $\Delta t = 1$
- Planned Use: $D_{\mathrm{p}} = 10^{-1/15}$, $\Delta t = 1$

5.1 シミュレーション 1

まず，それぞれ異なる特徴を持つ複数の利用者が存在する環境でシミュレーションを行い各方式を評価した．

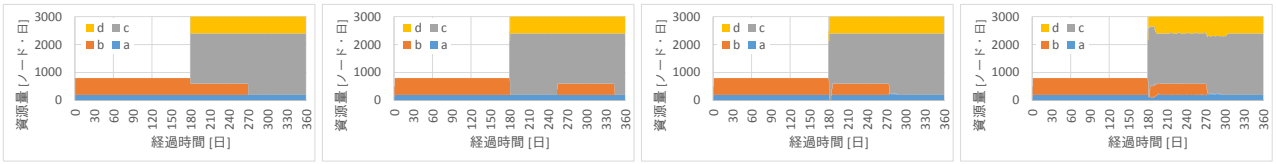
本シミュレーションでは，a, b, c, d の 4 名の利用者が存在する環境を想定する．各利用者の特徴は以下の通り．

- 利用者 a: 一定ペース型．資源配分期間の初めからジョ



(A) 理想的な結果　**(B)** 線形減衰型のフェアシェア　**(C)** 指数減衰型のフェアシェア　**(D)** Planned Use

図 5: シミュレーション 1: 累積使用資源率の推移



(A) 理想的な結果　**(B)** 線形減衰型のフェアシェア　**(C)** 指数減衰型のフェアシェア　**(D)** Planned Use

図 6: シミュレーション 1: 日別使用資源量の推移

ブ投入開始．時間あたりの要求資源量が常用資源量と一致．資源配分期間の終わりまでに配分資源量を使い切る

- 利用者 b: 先行型．資源配分期間の初めからジョブ投入開始．時間あたりの要求資源量が常用資源量を超過．資源配分期間の終わりまでに配分資源量を使い切る
- 利用者 c: 追い込み型．資源配分期間の途中からジョブ投入開始．時間あたりの要求資源量が常用資源量を超過．資源配分期間の終わりまでに配分資源量を使い切る
- 利用者 d: 遅刻型．資源配分期間の途中からジョブ投入開始．時間あたりの要求資源量が常用資源量と一致．資源配分期間の終わりまでに配分資源量を使い切らない

システムの運用形態，利用者への配分資源量，個々の投入ジョブの要求資源量の詳細は以下の通りである．

- 計算ノード数: 3000[ノード]
- 資源配分期間: 360[日]
- 配分資源量（括弧内は常用資源量）:
  - 利用者 a: 200×360[ノード・日] (200[ノード])
  - 利用者 b: 400×360[ノード・日] (400[ノード])
  - 利用者 c: 1800×360[ノード・日] (1800[ノード])
  - 利用者 d: 600×360[ノード・日] (600[ノード])
- 投入ジョブ（括弧内はジョブ投入開始から投入完了までの 1 日あたりの要求資源量）:
  - 利用者 a: 資源配分期間の初めから終わりまで，要求ノード数 100, 実行時間 1 日のジョブを 12 時間ごとに 1 回投入（200[ノード・日]）
  - 利用者 b: 資源配分期間の初めから 240 日間，要求ノード数 100, 実行時間 1 日のジョブを 4 時間ごとに 1 回投入（600[ノード・日]）
  - 利用者 c: 180 日目から 180 日間（資源配分期間の終わりまで），要求ノード数 100, 実行時間 1 日のジョブを 40 分ごとに 1 回投入（3600[ノード・日]）
  - 利用者 d: 180 日目から 180 日間（資源配分期間の終わりまで），要求ノード数 100, 実行時間 1 日のジョブを 4 時間ごとに 1 回投入（600[ノード・日]）

ジョブ投入の様子を示しているのが**図 4(A)** および**図 4(B)** である．図 4(A) は各利用者の累積要求資源量の時間変化，図 4(B) は累積要求資源率の時間変化である．利用者 a, b, c は配分資源量をすべて使い切るようにジョブを投入するが，利用者 d は配分資源量の 1/2 のジョブしか投入しない．

図 5 は，2 章の運用方針に基づいた理想的な累積使用資源率の推移，および，各方式による累積使用資源率の推移をシミュレーションした結果である．**図 6** は，図 5 の日別使用資源量の推移を表した図である．

2 章の運用方針に基づいた理想的な結果 (**図 5(A)**, **図 6(A)**) は，以下のようになる．

- 一定ペース型の利用者 a は，時間あたりの要求資源量が常用資源量（200[ノード]）と同じになるようなジョブ投入をしているため，常用資源量分の資源を使用する
- 先行型の利用者 b は，時間あたりの要求資源量が常用資源量（400[ノード]）を超過するようなジョブ投入をしているため，遊休資源が存在する資源配分期間の初めから 179 日目までは常用資源量分の資源 に加えて遊休資源を使用し，180 日目以降は遊休資源が存在しないため常用資源量分の資源のみを使用する
- 追い込み型の利用者 c は，時間あたりの要求資源量が常用資源量（1800[ノード]）を超過するようなジョブ投入をしているため，遊休資源が存在しない 180 日目から 270 日目までは常用資源量分の資源を使用し，271 日目以降は遊休資源が存在するため，常用資源量分の資源に加えて遊休資源を使用する
- 遅刻型の利用者 d は，時間あたりの要求資源量が常用

表 1: シミュレーション 1 の結果

	線形減衰型	指数減衰型	Planned Use
利用者 a	◎	△	○
利用者 b	×	△	△
利用者 c	○	○	○
利用者 d	◎	◎	○

資源量（600[ノード]）と同じになるようなジョブ投入をしているため，常用資源量分の資源を使用する

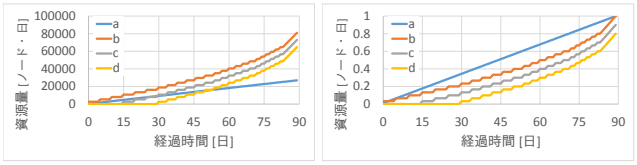
線形減衰型のフェアシェアによるシミュレーション（**図 5(B)**，**図 6(B)**）では，利用者 b の使用資源量が理想と異なる結果となった．利用者 a, d は概ね理想的な結果を得られているが，利用者 b は 180 日目以降の 73 日間ひとつもジョブが実行されていない．本来使用可能な資源（常用資源量分の資源 ）が長期にわたり全く使用できていないのは問題である．これは 3.1.3 項で述べた通り，過去に時間あたりに常用資源量以上の資源を多く使用した場合に他の利用者より優先度が低い状態が長期間（73 日間）続くためである．

指数減衰型のフェアシェアによるシミュレーション（**図 5(C)**，**図 6(C)**）では，利用者 a, b の使用資源量が理想と異なる結果となった．利用者 d は概ね理想的な結果が得られているが，利用者 c の影響により利用者 a は 180 日目以降の 2 日間，利用者 b は 3 日間，ジョブがひとつも実行されていない．順調に常用資源量分の資源を使用できていた利用者 a, b が，追い込み型の利用者 c によって，突然全く資源を使用できなくなるのは問題である．これは 3.2.3 項で述べた通り，過去にジョブを実行していなかった利用者の優先度が高くなることが原因である．

Planned Use のフェアシェアによるシミュレーション（**図 5(D)**，**図 6(D)**）では，線形減衰型および指数減衰型のフェアシェアでみられた問題は生じず，より理想に近い結果が得られた．利用者 a, d については 180 日目以降，利用者 c のジョブが投入されたため，一時的に使用資源量が減少している日が存在するが，ジョブが全く実行されない期間はなく概ね理想通りに制御されている．特に，資源配分期間の初めから一定ペースで計画的にジョブ投入してきた利用者 a のジョブが順調に実行されており，指数減衰型のフェアシェアの問題が解決されている．利用者 b は 180 日目から 1 日ジョブが実行されていないが，線形減衰型および指数減衰型のフェアシェアと比較するとジョブが実行されない日数は少なく良好な結果となっている．

シミュレーション 1 の結果を**表 1** にまとめる．表中の記号は個々の利用者の使用資源量が理想的な結果と比較してどの程度一致しているかを評価したものであり，それぞれ以下の意味である．

- ◎: 概ね理想的な結果と一致
- ○: 一部を除き理想的な結果と一致．理想的な結果となっていない期間でも毎日ジョブが実行されている
- △: ジョブが実行されない日が数日存在
- ×: ジョブが実行されない日が数十日以上存在



(A) 累積要求資源量　　**(B)** 累積要求資源率

図 7: シミュレーション 2: 累積要求資源量・率の推移

利用者にとっての利便性に影響する “ジョブが実行されない日数” に着目し，個々の利用者の使用資源量について理想的な結果と比較して総評すると， Planned Use の結果が最も良く，次に指数減衰型のフェアシェアが良い結果となった．

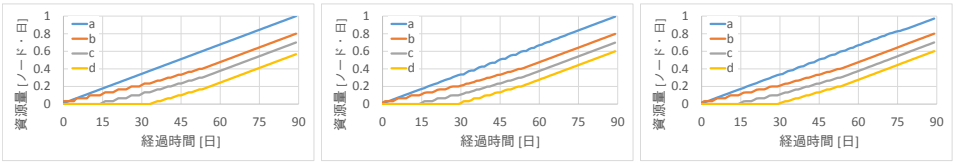
シミュレーション 1 の結果から，資源配分期間の途中から多くの資源を要求するジョブを投入し始める利用者 c による他の利用者への影響の度合いが各方式の善し悪しを決定づける要素であることが分かった．実際の運用では，利用者 c のような資源配分期間の終わりに近づくとジョブ投入の頻度が増す追い込み型の利用者が多く存在する．このことから，シミュレーション 1 で良い結果が得られた指数減衰型のフェアシェアと Planned Use について，現実の運用を想定したモデルを用いてシミュレーション 2 を行った．

5.2 シミュレーション 2

配分資源量を割り当てる運用においては，配分資源量を使い切るために資源配分期間の終わりに多くの利用者が集中してジョブ投入する傾向がみられる [8]．この時，資源配分期間の初めから時間あたりの要求資源量が常用資源量の範囲に収まるよう一定のペースでジョブを投入していた利用者のジョブ実行が著しく妨げられるのは望ましくない．シミュレーション 1 で良い結果が得られた 指数減衰型のフェアシェアと Planned Use について，資源配分期間の終わりにジョブ投入が集中する状況を再現するシミュレーションを行った．

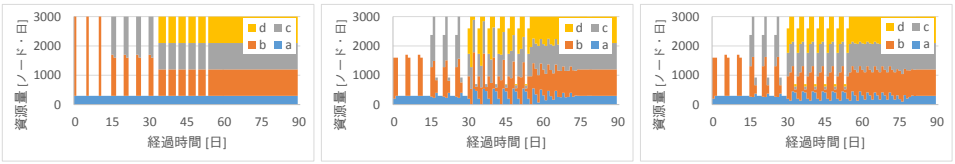
本シミュレーションでは，a, b, c, d の 4 名の利用者が存在する環境を想定する．各利用者の特徴は以下の通り．

- 利用者 a: 一定ペース型．資源配分期間の初めからジョブ投入開始．時間あたりの要求資源量が常用資源量と一致．資源配分期間の終わりまでに配分資源量を使い切る
- 利用者 b: 追い込み型．資源配分期間の途中からジョブ投入開始．時間あたりの要求資源量が常用資源量を超過．資源配分期間の終わりまでに配分資源量を使い切る
- 利用者 c: 追い込み型．資源配分期間の途中からジョブ



(A) 理想的な結果　**(B)** 指数減衰型のフェアシェア　**(C)** Planned Use

図 8: シミュレーション 2: 累積使用資源率の推移



(A) 理想的な結果　**(B)** 指数減衰型のフェアシェア　**(C)** Planned Use

図 9: シミュレーション 2: 日別使用資源量の推移

投入開始．時間あたりの要求資源量が常用資源量を超過．資源配分期間の終わりまでに配分資源量を 90.0% 使い切る

- 利用者 d: 追い込み型．資源配分期間の途中からジョブ投入開始．時間あたりの要求資源量が常用資源量を超過．資源配分期間の終わりまでに配分資源量を 80.0% 使い切る

システムの運用形態，利用者への配分資源量，個々の投入ジョブの要求資源量は以下の通りである．

- 計算ノード数: 3000[ノード]
- 資源配分期間: 90[日]
- 配分資源量（括弧内は常用資源量）:
  - 利用者 a: 300 × 90[ノード · 日] (300[ノード])
  - 利用者 b: 900 × 90[ノード · 日] (900[ノード])
  - 利用者 c: 900 × 90[ノード · 日] (900[ノード])
  - 利用者 d: 900 × 90[ノード · 日] (900[ノード])
- 投入ジョブ（括弧内は区間ごとの 1 日あたりの要求資源量）:
  - 利用者 a: 資源配分期間の初めから終わりまで，要求ノード数 100, 実行時間 1 日のジョブを 8 時間ごとに 1 回投入（資源配分期間の初めから終わりまで 300[ノード · 日]）
  - 利用者 b: 資源配分期間の初めから 30 日間は 5 日おきに，30 日目から 24 日間は 4 日おきに，54 日目から 18 日間は 3 日おきに，72 日目から 12 日間は 2 日おきに，84 日目から資源配分期間の終わりまでは毎日，要求ノード数 100, 実行時間 1 日のジョブを 3200 秒ごとに 1 回投入することを 1 日の間繰り返す（資源配分期間の初めから 30 日間は 540[ノード · 日]，30 日目から 24 日間は 675[ノード · 日]，54 日目から 18 日間は 900[ノード · 日]，72 日目から 12 日間は 1350[ノード · 日]，84 日目から資源配分期間の終わりまでは 2700[ノード · 日]）
  - 利用者 c: 資源配分期間の初めから 15 日間はジョブを投入しないが，それ以降は利用者 b と同じだけのジョブを投入する（15 日目から 15 日間は 540[ノード · 日]，30 日目から 24 日間は 675[ノード · 日]，54 日目から 18 日間は 900[ノード · 日]，72 日目から 12 日間は 1350[ノード · 日]，84 日目から資源配分期間の終わりまでは 2700[ノード · 日]）
  - 利用者 d: 資源配分期間の初めから 30 日間はジョブを投入しないが，それ以降は利用者 b と同じだけのジョブを投入する（30 日目から 24 日間は 675[ノード · 日]，54 日目から 18 日間は 900[ノード · 日]，72 日目から 12 日間は 1350[ノード · 日]，84 日目から資源配分期間の終わりまでは 2700[ノード · 日]）

ジョブ投入の様子を示しているのが**図 7(A)** および**図 7(B)** である．図 7(A) は各利用者の累積要求資源量の時間変化，図 7(B) は累積要求資源率の時間変化である．

図 8 は，2 章の運用方針に基づいた理想的な累積使用資源率の推移，および，指数減衰型のフェアシェアと Planned Use による累積使用資源率の推移をシミュレーションした結果である．**図 9** は，図 8 の日別使用資源量の推移を表した図である．

2 章の運用方針に基づいた理想的な結果（**図 8(A)**, **図 9(A)**）は，以下のようになる．

- 一定ペース型の利用者 a は，時間あたりの要求資源量が常用資源量（300[ノード]）と同じになるようなジョブ投入をしているため，常用資源量分の資源を使用する
- 追い込み型の利用者 b, c, d は，時間あたりの要求資源量が常用資源量（900[ノード]）を超過するようなジョブ投入をしているため，遊休資源が存在する 42 日目までは常用資源量分の資源に加えて遊休資源を使用し，遊休資源が存在しない 43 日目以降は常用資源量分の資源を使用する

指数減衰型のフェアシェアによるシミュレーション(**図 8(B)**，**図 9(B)**）では，計画的にジョブ投入している一定ペース型の利用者 a が，追い込み型の利用者 b, c, d のジョブの影響によりジョブがひとつも実行されない日が何度も存在する．資源配分期間の終わりが近づき追い込み型の利用者が多くなるほど，計画的にジョブ投入していた利用者への影響が顕著になるのは問題である．

Planned Use のフェアシェアによるシミュレーション（**図 8(C)**，**図 9(C)**）では，利用者 b, c, d の影響により利用者 a の使用資源量が変動するものの，ジョブが全く実行されない日はなく，指数減衰型のフェアシェアの問題が解消されている．

資源配分期間の終わりにジョブ投入が集中する運用想定モデルを用いたシミュレーション 2 の結果から，2 章の運用方針では，指数減衰型のフェアシェアより Planned Use の方が望ましい制御が実現できていることが示された．

6. おわりに

本稿では配分資源量を割り当てる運用における，ジョブの優先度制御の一案を示し，当該運用下における従来のフェアシェアの問題点を解決する手段として Planned Use を提案した．Planned Use は個々の利用者の配分資源量に基づいて常に一定の資源を各利用者が使用できるようにするアルゴリズムである．従来のフェアシェアと比較して，ある利用者のジョブ実行が他の利用者のジョブ実行に及ぼす悪影響を軽減することができ，一定のペースで計画的に資源を使用することが可能であることをシミュレーションにより確認した．

本稿では個々の利用者にあらかじめ割り当てられた配分資源量に基づいて常用資源量を決定しジョブの優先度付けを行っているが，実際のスーパーコンピュータでの運用では追加料金を支払った利用者のジョブを優先実行するようにするなど，配分資源量以外の要因で優先度を変更したい場合が存在する．今後は，本稿で提案した方式を拡張し配分資源量以外の要因により利用者の優先度を高める仕組みを検討したい．また，今回のシミュレーションでは，利用者数・配分資源量・ジョブの要求資源量等をアルゴリズム評価のためのシンプルなモデルにして検証したが，今後は実際の運用データを用いた実践的な検証を行う予定である．

謝辞 本論文の一部は，文部科学省「特定先端大型研究施設運営費等補助金（次世代超高速電子計算機システムの開発・整備等）」で実施された内容に基づくものである．

参考文献

[1] 山本啓二，宇野篤也，塚本俊之，菅田勝文，庄司文由：（続）スーパーコンピュータ「京」の利用：1．スーパーコンピュータ「京」の運用状況，情報処理，Vol. 55, No. 8, pp. 786–793 (2014).
[2] 理化学研究所計算科学研究機構 (AICS)：HPCI 戦略プログラム（戦略 5 分野）について，理化学研究所（オンライン），入手先 ⟨http://www.aics.riken.jp/jp/science/spire⟩（参照 2016-08-01）.
[3] High Performance Computing Infrastructure: 課題選定の結果, High Performance Computing Infrastructure（オンライン），入手先 ⟨http://www.hpci-office.jp/pages/adoption⟩（参照 2016-08-07）.
[4] 理化学研究所情報基盤センター：利用申請（課題及びアカウント作成の申請），理化学研究所（オンライン），入手先 ⟨http://accc.riken.jp/supercom/application/⟩（参照 2016-07-23）.
[5] SchedMD LLC: Multifactor Priority Plugin, SchedMD LLC (online), available from ⟨http://slurm.schedmd.com/priority_multifactor.html⟩ (accessed 2016-08-07).
[6] International Business Machines Corporation: *Administering Platform LSF* (2014).
[7] Altair Engineering, Inc: *PBS Professional 13.0 Administrator's Guide* (2015).
[8] Yamamoto, K., Uno, A., Murai, H., Tsukamoto, T., Shoji, F., Matsui, S., Sekizawa, R., Sueyasu, F., Uchiyama, H., Okamoto, M., Ohgushi, N., Takashina, K., Wakabayashi, D., Taguchi, Y. and Yokokawa, M.: The K computer Operations: Experiences and Statistics, *Procedia Computer Science*, Vol. 29, No. Complete, pp. 576–585 (online), DOI: 10.1016/j.procs.2014.05.052 (2014).