情報処理学会研究報告
IPSJ SIG Technical Report

Vol.2011-ARC-195 No.29
Vol.2011-OS-117 No.29
2011/4/14

継続概念に基づくCMPにおける複数サービスの制御

森山英明[†1]　山内利宏[†1]　谷口秀夫[†1]

継続概念に基づくCMPでは，スレッドをハードウェアスケジューラで制御することで，処理の高速な実行を実現している．一方で，ハードウェアスケジューラはFIFOの規則でスケジュールをすることから，スレッドの優先度を考慮した実行ができない．そこで，ハードウェアスケジューラを支援するソフトウェアスケジューラとして，一つのサービスを対象とし，同時使用TEU数を制御する制御法を提案した．しかし，複数のサービス同時実行時に，各サービスの優先度にしたがった制御を行うことができない．このため，この制御法を拡張する必要がある．

本論文では，同一のサービスを複数同時に実行したときに，特定のサービスの優先実行や各サービスの均等実行を可能とする制御法を提案する．また，提案した制御法をシミュレータによって評価した結果を示す．

Control Method of Multiple Services for CMP Based on Continuation Model

Hideaki Moriyama [†1], Toshihiro Yamauchi [†1] and Hideo Taniguchi [†1]

In chip multi-processor based on the continuation concept, the hardware scheduler controls threads and achieves the high performance on thread-scheduling. However, the execution of each thread can not consider priority to threads because the hardware thread scheduler schedules threads in FIFO manner. Therefore, when multiple services execute simultaneously, the execution of each service can not consider priority to service. It needs the software support to control the execution of each service.

This paper describes the software scheduler of multiple services to support the hardware scheduler. In addition, this paper reports the evaluation of the software scheduler, which targets multiple services.

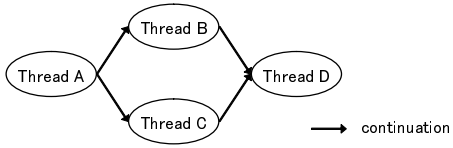


図 1　継続概念

1. はじめに

マルチコアプロセッサ環境の普及に伴い，複数のコアを持つ計算機上での並列処理の効率的な実行が重要である．しかし，従来の汎用プロセッサとオペレーティングシステムを用いた並列処理では，システムコール発行やコンテキストスイッチといったオペレーティングシステムのオーバヘッドが問題になる[1)]．また，並列処理の研究として，データフローマシンを用いた並列実行方式[2)]がある．データフローマシンでは，実行が可能となったスレッドから順に処理を開始する．このため，スレッド管理によるオーバヘッドを削減し，高速な並列処理を実現できる．しかし，スレッドの優先度による制御が行えない，特別なアーキテクチャによる命令セットを用いるため実装が困難といった問題がある．

これらの問題への対処として，細粒度マルチスレッドの手法[3)]が研究されている．細粒度マルチスレッド方式は，データフローモデルを基盤としたスレッド並列実行を追及する手法である．Fuce(FUsion of Communication and Execution) プロセッサは，細粒度マルチスレッド方式を採用しており，複数のスレッド実行ユニット (Thread Execution Unit，以降，TEUと略す) を持つCMPとして開発[4)5)]されている．Fuceプロセッサにおいて，サービスは複数の細粒度スレッド (以降，スレッドと略す) で構成される．スレッドは走りきりであり，処理を開始すると処理終了までTEUを横取りされない．また，ハードウェアのスケジューラは，継続と呼ぶ同期手法によりスレッドの実行制御を行う．これらにより，高速な並列処理を実現する．

図1に継続の概念を示す．図1は，4つのスレッドA，B，C，Dの依存関係を示している．BとCはAの計算結果を必要とし，DはBとCの計算結果を必要としている．これら4つのスレッドを実行するには，Aは計算結果をBとCに通知し，BとCは計算結果を

†1 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University

Vol.2011-ARC-195 No.29
Vol.2011-OS-117 No.29
2011/4/14

D に通知する．Fuce プロセッサでは，この結果の通知を継続と呼ぶ．A は B と C に継続し，B と C は D に継続するという．ハードウェアスケジューラのスケジュール規則は，継続が解決する順に FIFO であり，サービスを構成するスレッドの実行において，優先度の概念はない．このため，Fuce プロセッサ上で複数のサービスが同時に実行する場合に，特定のサービスの優先実行や各サービスの均等実行といった制御ができない．

そこで，サービスを構成するスレッド間の継続に着目し，スレッド間の継続をソフトウェアスケジューラによって制御する手法が提案されている[6]．この手法では，一つのサービスを対象として同時使用 TEU 数を制御するスケジュール法を実現している．また，評価より，ソフトウェアスケジューラで設定する制御パラメータの性質を明らかにし，制御法を確立した．しかし，複数のサービス同時実行時に，各サービスの優先度にしたがった制御を行うことができない．このため，このスケジュール法を拡張する必要がある．

本論文では，同一のサービスを複数同時に実行したときに，特定のサービスの優先実行や各サービスの均等実行を可能とする制御法を提案する．まず，文献 6) で示したスケジュール法について，複数のサービスを制御できるようソフトウェアスケジューラを拡張する．次に，同一のサービスを複数同時実行したときの優先実行制御と均等実行制御を実現する実行制御法について述べる．最後に，提案した制御法をシミュレータによって評価した結果を示す．

2. スケジュール法の機構

2.1 基 本 方 式

スケジュール法は，以下の 3 つを前提条件とする．

(条件 1) PU のみによる処理

(条件 2) サービスのスレッド並列度 N は既知

(条件 3) TEU の総数 E(Execution Unit) は既知

スケジュール法は，スレッド間の継続にソフトウェアスケジューラが介入して継続を制御する．スケジュール法の基本方式[6] を図 2 に示し，以下に説明する．

(1) スレッド x_i からスレッド y_i への継続を，制御確率 P でスケジューラに継続する．

(2) スケジューラは，制御継続発行周期 ω で動作する．

(3) スケジューラは，1 回の動作につき，制御量 M 個のスレッドへ継続を発行する．

以降では，制御確率 P，制御継続発行周期 ω，および制御量 M を制御パラメータと呼ぶ．

次に，制御パラメータの特徴について記述する．制御確率 P は，スレッド制御への関与

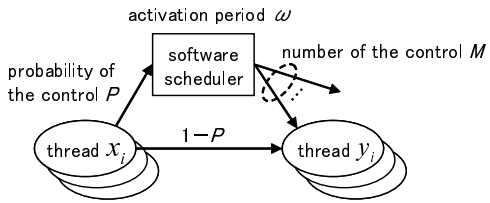


図 2 基本方式

度を調定するものである．制御確率 P の値が大きいと，サービス全体で継続元スレッドからスケジューラへ継続する数が増加し，スケジューラによる制御の頻度が高くなる．このため，より細かい制御が可能となる．一方，スケジューラが継続を発行する頻度が増加するため，スケジューラの処理時間が増加する．制御継続発行周期 ω と制御量 M について，文献 6) における分析より，サービスのスレッド平均実行時間 T とスレッド平均並列度 N を考慮してスケジューラのオーバヘッドが小さくなる条件式は，

$$\frac{M}{PN} \leq \frac{\omega}{T} \tag{1}$$

である．特に，スレッドの実行が比較的同期している場合は，$M = PN$ かつ $\omega = T$ のときにオーバヘッドは最小となる．

2.2 複数サービスの拡張

図 2 で示した基本方式に従い，一つのサービスを制御するスケジュール法の実装は，文献 6) に示した．ここでは，複数サービスを制御できるように拡張したスケジュール法について述べる．スケジュール法では，継続の発行元のスレッド (以降，継続元スレッドと略す) から継続の発行先のスレッド (以降，継続先スレッドと略す) への継続に対して，ソフトウェアスケジューラが介入して継続を制御する．また，複数サービスの制御において，ソフトウェアスケジューラは，制御確率 P，制御継続発行周期 ω，および制御量 M を各サービスで個別に管理する．

例として，2 個のサービスを制御する場合のスケジューラの処理構造を図 3 に示し，以下で説明する．

(1) i 番目のサービスの継続元スレッド src は，制御確率 P_i でソフトウェアスケジューラのスレッドである $thRegister$ へ継続する．$thRegister$ スレッドはサービスごとに用意されている．

Vol.2011-ARC-195 No.29
Vol.2011-OS-117 No.29
2011/4/14

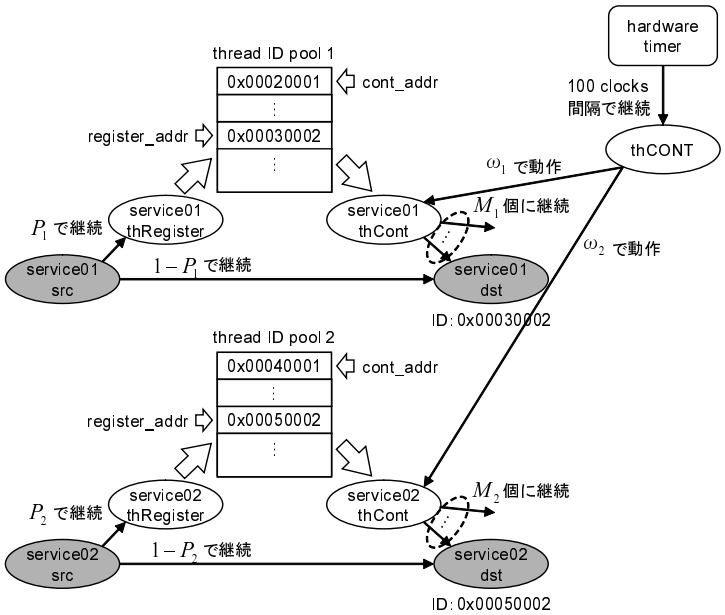


図 3 2 個のサービスを制御する場合のスケジューラの処理構造

(2) *thRegister* スレッドは継続先スレッド *dst* のスレッド ID を継続元スレッド *src* から受け取り，スレッド ID の格納領域 (以降，スレッド ID プールと呼ぶ) に格納する．ただし，スレッド ID プールの更新に伴い，スレッド ID プールの排他制御を行う．スレッド ID プールはサービスごとに用意されている．

(3) *thCONT* スレッドは，ハードウェアタイマから 100*clocks* 周期で継続を受けることで走行を開始する．各サービスの継続発行周期 ω を管理しており，i 番目のサービスの *thCont* スレッドに対して $\omega_i \div 100$ 回目の走行時に継続を発行する．

(4) i 番目のサービスの *thCont* スレッドは，ω_i 周期で継続を受けることで，ω 周期で走行を開始する．*thCont* スレッドは走行を開始するとスレッド ID プールから M_i 個のスレッド ID を取得する．

(5) *thCont* スレッドは，取得した M_i 個のスレッド ID のスレッドに対して継続を発行する．

3. 複数サービスを対象とした制御法

3.1 優先実行制御と均等実行制御

複数サービスの同時実行において，各サービスには優先度が設定されている．i 番目の優先度を $p_i(1, 2, \cdots, i, \cdots)$ で表し，優先度 p_i のサービスの個数を m_i と定義する．ここで，$p_i > p_{i+1}$ であり，優先度の値が小さいほど優先度は高い．

優先実行制御では，優先度が高いサービスに対して，サービスのスレッド並列度 N に等しい値の TEU を割り当てる．このとき，利用可能な TEU 数がスレッド並列度 N よりも小さい場合，利用可能な TEU 数をすべてこのサービスに割り当てる．具体的には，優先度 p_i のサービスが利用可能な TEU 数を x とした場合，$x \geq N$ ならば N 個の TEU を割り当て，$x < N$ ならば x 個の TEU を割り当てる．

均等実行制御では，優先度の同じサービスに対して，利用可能な TEU 数を均等に分割して割り当てる．具体的には，優先度 p_i のサービス m_i 個について，利用可能な TEU 数を x とした場合，各サービスに $x \div m_i$ 個の TEU を割り当てる．

3.2 実行制御法

実行制御法は，複数サービス実行時に，各サービスの優先度に応じた制御パラメータの設定が必要となる．以下に，制御パラメータの設定を示す．

(1) 制御継続発行周期 ω はスレッド平均実行時間 T と等しい値とする．これは，文献 6) の評価より，$\omega = T$ とすることでソフトウェアスケジューラのオーバヘッドを低減できるからである．

(2) 優先度 p_1 である m_1 個のサービスについて，制御確率 $P = 0$ として制御する．$P = 0$ とすることで，サービスのスレッド間の継続に対してソフトウェアスケジューラが介入しないため，サービスの同時使用 TEU 数が抑制されることはない．

(3) 優先度 p_1 のサービス m_1 個が使用していない TEU について，優先度 p_2 以降のサービスで同時使用 TEU 数を抑制する必要のないものについて，$P = 0$ で制御する．すなわち，TEU の総数 E に対して，

$$E \geq \sum_{i=1}^{j} m_i \times N > E - m_k \times N \tag{2}$$

(k は, 優先度 p_{j+1} 以下で最高優先度のサービスを指す)

を満足する優先度 p_1 から p_j までのサービスは，同時使用 TEU 数を抑制する必要が

Vol.2011-ARC-195 No.29
Vol.2011-OS-117 No.29
2011/4/14

ないため，$P = 0$ とする．

(4) 上記 (1)，(2) でサービスに割り当てた TEU を除き，次に優先度の高いサービス群に残りの TEU を分割して均等に割り当てる．すなわち，

$$E_j = \sum_{i=1}^{j} m_i \times N \tag{3}$$

とすると，優先度 p_k のサービス m_k 個について，同時使用 TEU 数が，

$$\frac{E - E_j}{m_{j+1}} \tag{4}$$

となるように制御する．この値は，同時使用 TEU 数の制御の目標値 (以降，制御目標値と略す) である．サービスの同時使用 TEU 数を制御目標値に近い値で制御するために，文献 6) で示した一つのサービスを対象とした制御の測定結果を用いる．ここで，サービスの開始から終了までの時間 ET に対して，時刻 $t(1 \leq t \leq ET)$ におけるサービスの同時使用 TEU 数を N_t として，同時使用 TEU 数の平均 N_{av} を，

$$N_{av} = \frac{1}{ET} \sum_{t=1}^{ET} N_t \tag{5}$$

で定義する．サービスの同時使用 TEU 数を (制御目標値)$\geq N_{av}$ となる最大の N_{av} に制御することで，このサービスを制御目標値に近い値で制御することができる．このとき，制御確率 $P = 1.0$ とし，制御量 M によって制御する．$P = 1.0$ とすることで，M を適切な値にすることによる制御効果を明らかにできる．

(5) 優先度 p_{k+1} 以降のサービスは $P = 1.0, M = 0$ として制御する．このとき，サービスのスレッドはスレッド ID プールに登録され，継続を発行されない．つまり，サービスのスレッドは，実行を開始できない状態となる．

また，$P = 0$ として制御するサービスの実行終了を契機として，優先度に応じて (1)〜(4) を行い，制御パラメータの再設定を行う．

4. 評　　価

4.1 評価対象サービス

評価対象サービスとして，スレッド並列度 $N = 10$，スレッド平均実行時間 $T = 9700$ の 10-Queens 問題解法プログラム (以降，10-Queens プログラムと略す) を用いた．この理由は，単数の 10-Queens プログラム実行制御時の同時使用 TEU 数 N_{av} と制御パラメータの関係を文献 6) で明らかにしており，この関係を実行制御法で用いるためである．

4.2 評価の観点

評価では，複数の 10-Queens プログラムの同時実行時に，各サービスの優先度に応じて，優先実行制御と均等実行制御を行ったときの測定結果と考察を示す．

例えば，同時に実行するサービス数が 2 個，3 個，5 個の場合について，サービスの優先度の組み合わせを表 1 に示す．表 1 において，複数のサービスを $AP1, AP2, \cdots$ と略し，「優先度」の項目は，各サービスの優先度の大小関係を表す．例えば，通番 3 の $AP1 > AP2 = AP3$ の場合，$AP1$ は優先度 p_1，$AP2$ と $AP3$ は優先度 p_2，$p_1 > p_2$ である．「制御内容」の項目は，3.2 節で示した実行制御法に従い，各サービスの優先度に基づいた制御パラメータの設定や変更を示している．また，「同じ制御内容の優先度」の項目では，各サービスの優先度にしたがって制御した場合に制御パラメータの設定が同じになる優先度の組み合わせを示している．

ここでは，以下の観点で評価を行う．

(1) スレッド並列度の合計が TEU の総数 E を超過しない場合

(2) スレッド並列度の合計が TEU の総数 E を超過する場合

- (a) 各サービスを均等実行する場合
- (b) 1 サービスを優先実行し，他サービスを均等実行する場合
- (c) 2 サービスを優先実行し，他サービスを均等実行する場合
- (d) 3 サービス以上を優先実行する場合

4.3 考　　察

4.3.1 評 価 内 容

評価のために，24 個の TEU を搭載し，1 命令を 1 クロックで実行する Fuce シミュレータを作成した．以降では，複数サービスの実行制御について同時使用 TEU 数の理想の図と測定結果の図を示し，考察を示す．また，測定結果の図中の点は，同時使用 TEU 数の 10000 *clock* 間の平均値である．

評価尺度として，サービスの同時使用 TEU 数の分散を定義する．N_{av} に対する N_t の割合を，

$$s_t = \frac{N_t}{N_{av}} \tag{6}$$

で定義する．また，s_t の平均を $\bar{s}$ として，同時使用 TEU 数の分散 $V(S)$ を

Vol.2011-ARC-195 No.29
Vol.2011-OS-117 No.29
2011/4/14

表 1 複数サービスの優先度の組み合わせ

通番	サービス数	優先度	制御内容	同じ制御内容の優先度
1	2	$AP1 = AP2$	(初期) $AP1, AP2 : P = 0$	
2	3	$AP1 = AP2 = AP3$	(初期) $AP1$〜$AP3 : P = 0$	
3		$AP1 > AP2 = AP3$	(初期) $AP1 : P = 0$, $AP2, AP3 : M = 10$(制御目標値 7 に対し, $N_{av} = 5.80$) ($AP1$ 終了時) $AP2, AP3 : P = 0$	$AP1 > AP2 > AP3$
4		$AP1 = AP2 > AP3$	(初期) $AP1, AP2 : P = 0$, $AP3 : M = 6$(制御目標値 4 に対し, $N_{av} = 3.58$) ($AP1, AP2$ 終了時) $AP3 : P = 0$	
5	5	$AP1 = AP2 = AP3 = AP4 = AP5$	(初期) $AP1$〜$AP5 : P = 0$	
6		$AP1 > AP2 = AP3 = AP4 = AP5$	(初期) $AP1$〜$AP5 : P = 0$, $AP2$〜$AP5 : M = 5$(制御目標値 3.5 に対し, $N_{av} = 2.98$) ($AP1$ 終了時) $AP2$〜$AP5 : P = 0$	
7		$AP1 = AP2 > AP3 = AP4 = AP5$	(初期) $AP1, AP2 : P = 0$, $AP3$〜$AP5 : M = 2$(制御目標値 2 に対し, $N_{av} = 1.23$) ($AP1, AP2$ 終了時) $AP3$〜$AP5 : P = 0$	$AP1 > AP2 > AP3 = AP4 = AP5$
8		$AP1 = AP2 = AP3 > AP4 = AP5$	(初期) $AP1$〜$AP3 : P = 0$, $AP4, AP5 : M = 0$ ($AP1$〜$AP3$ 終了時) $AP4, AP5 : P = 0$	$AP1 = AP2 = AP3 = AP4 > AP5$, $AP1 = AP2 = AP3 > AP4 > AP5$
9		$AP1 > AP2 = AP3 > AP4 = AP5$	(初期) $AP1 : P = 0$, $AP2, AP3 : M = 10$(制御目標値 7 に対し, $N_{av} = 5.80$), $AP4, AP5 : M = 0$ ($AP1$ 終了時) $AP2, AP3 : P = 0$, $AP4, AP5 : M = 3$(制御目標値 2 に対し, $N_{av} = 1.82$) ($AP2, AP3$ 終了時) $AP4, AP5 : P = 0$	$AP1 > AP2 = AP3 > AP4 > AP5$
10		$AP1 > AP2 = AP3 = AP4 > AP5$	(初期) $AP1 : P = 0$, $AP2$〜$AP4 : M = 7$(制御目標値 4.67 に対し, $N_{av} = 4.12$), $AP5 : M = 0$ ($AP1$ 終了時) $AP2$〜$AP4 : P = 0$, $AP5 : M = 0$ ($AP2$〜$AP4$ 終了時) $AP5 : P = 0$	
11		$AP1 = AP2 > AP3 > AP4 = AP5$	(初期) $AP1, AP2 : P = 0$, $AP3 : M = 6$(制御目標値 4 に対し, $N_{av} = 3.58$), $AP4, AP5 : M = 0$ ($AP1$,$AP2$ 終了時) $AP3 : P = 0$, $AP4, AP5 : M = 10$(制御目標値 7 に対し, $N_{av} = 5.80$) ($AP3$ 終了時) $AP4, AP5 : P = 0$	$AP1 > AP2 > AP3 > AP4 = AP5$
12		$AP1 = AP2 > AP3 = AP4 > AP5$	(初期) $AP1, AP2 : P = 0$, $AP3, AP4 : M = 3$(制御目標値 2 に対し, $N_{av} = 1.82$), $AP5 : M = 0$ ($AP1$,$AP2$ 終了時) $AP3, AP4 : P = 0$, $AP5 : M = 6$(制御目標値 4 に対し, $N_{av} = 3.58$) ($AP3$,$AP4$ 終了時) $AP5 : P = 0$	$AP1 > AP2 > AP3 = AP4 > AP5$
13		$AP1 > AP2 > AP3 > AP4 > AP5$	(初期) $AP1, AP2 : P = 0$, $AP3 : M = 6$(制御目標値 4 に対し, $N_{av} = 3.58$), $AP4, AP5 : M = 0$ ($AP1$,$AP2$ 終了時) $AP3, AP4 : P = 0$, $AP5 : M = 6$(制御目標値 4 に対し, $N_{av} = 3.58$) ($AP3$ 終了時) $AP4, AP5 : P = 0$ ($AP4$ 終了時) $AP5 : P = 0$	$AP1 = AP2 > AP3 > AP4 > AP5$

$$V(S) = \frac{1}{ET}\sum_{t=1}^{ET}(s_t - \bar{s})^2 \tag{7}$$

で定義する．複数のサービスを同時実行することによりサービスの $V(S)$ が増加する場合，同時使用 TEU 数に関して他サービスから影響を受けているといえる．

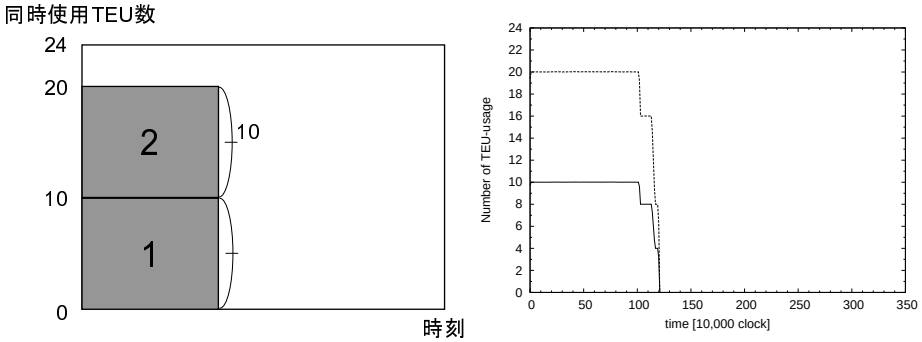
4.3.2 スレッド並列度の合計が TEU の総数を超過しない場合

最初に，複数サービスのスレッド並列度 N の合計値が TEU の総数 E を超過していない場合について，表 1 の通番 1($AP1 = AP2$) の測定結果を図 4 に示す． 各サービスのスレッド並列度の合計は $2 \times 10 = 20$ であり，TEU の総数 $E = 24$ を下回っている．このため，各サービスは同時使用 TEU 数 $= 10$ で実行する．このとき，各 AP の同時使用 TEU 数の平均 $N_{av} = 9.52$ であり，スレッド並列度 $N = 10$ に非常に近い値である．また，同時使用 TEU 数の分散 $V(S) = 0.021$ である．

4.3.3 スレッド並列度の合計が TEU の総数を超過する場合

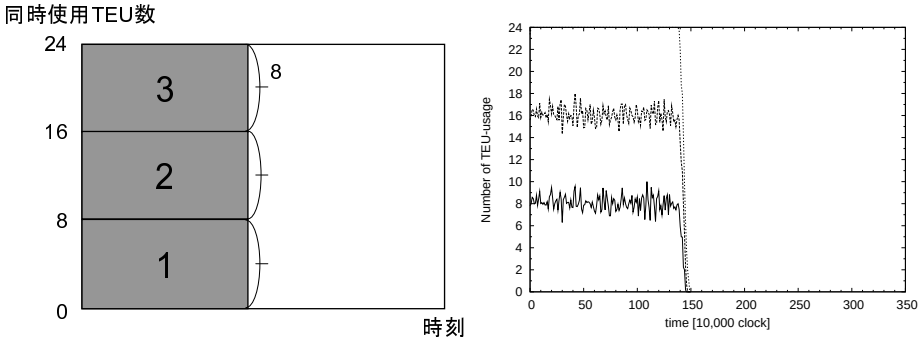
以降では，サービスのスレッド並列度の合計値が TEU の総数 E を超過する場合について考察する．各サービスを均等実行制御する場合について，表 1 の通番 2($AP1 = AP2 = AP3$) の測定結果を図 5 に示す． また，表 1 の通番 5($AP1 = AP2 = AP3 = AP4 = AP5$) の

Vol.2011-ARC-195 No.29
Vol.2011-OS-117 No.29
2011/4/14



(A) 理想　(B) 測定結果

図 4　2 サービス ($AP1 = AP2$) の同時使用 TEU 数

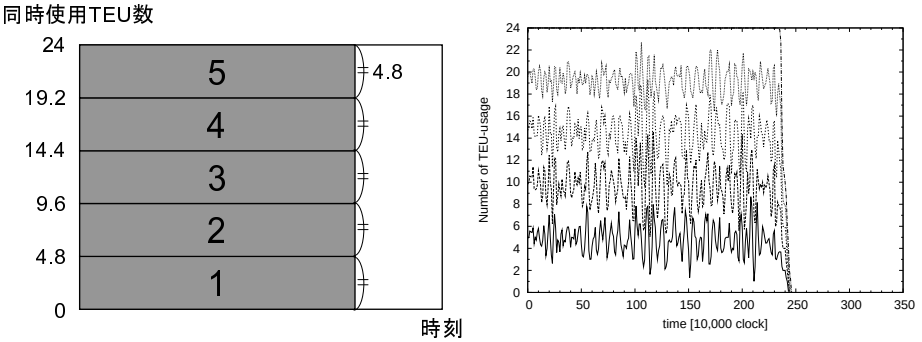


(A) 理想　(B) 測定結果

図 5　3 サービス ($AP1 = AP2 = AP3$) の同時使用 TEU 数

測定結果を図 6 に示す．図 5 と図 6 より，以下のことがわかる．

(1)　優先度 p_i のサービス m_i 個について，制御確率 $P = 0$ とする制御により，各サービスの均等実行が可能である．図 5 では $AP1$ から $AP3$ までの各サービスは TEU を 8 個ずつ使用し，図 6 では $AP1$ から $AP5$ までの各サービスは TEU を 4.8 個ずつ使用して実行する．この理由は，各サービスのスレッド並列度 N とスレッド実行時間 T がそれぞれ等しく，さらに，スレッド実行の契機が比較的同期しているからであると考えられる．



(A) 理想　(B) 測定結果

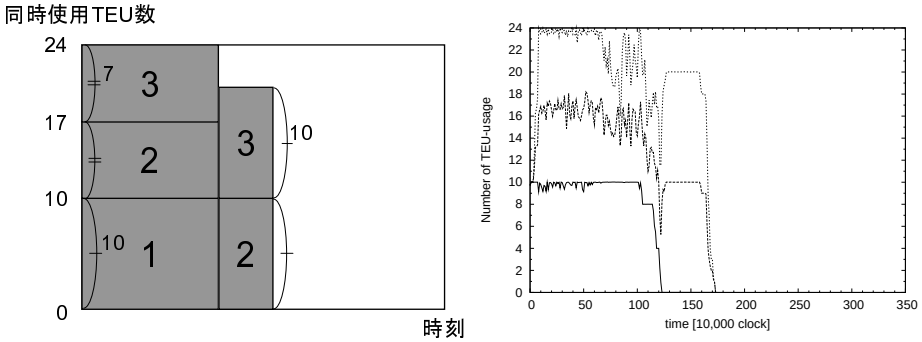
図 6　5 サービス ($AP1 = AP2 = AP3 = AP4 = AP5$) の同時使用 TEU 数

(2)　制御確率 $P = 0$ とする優先度 p_i のサービス m_i 個の均等実行制御において，各サービスのスレッド並列度の合計が TEU の総数を上回る場合，すなわち，$E < m_i \times N$ を満たすとき，各サービスの同時使用 TEU 数の分散 $V(S)$ は大きくなる．特に，サービス数が増えるにしたがい，$V(S)$ は大きくなる．図 5 において，$AP1$ の同時使用 TEU 数の分散 $V(S) = 0.025$ であり，図 4 における $AP1$ の $V(S) = 0.021$ と比較して大きい．さらに，図 6 における $AP1$ は $V(S) = 0.112$ であり，図 5 と比較してさらに大きい．

次に，1 サービスを優先実行し他サービスを均等実行する場合について，表 1 の通番 3($AP1 > AP2 = AP3$) の測定結果を図 7 に示す． また，表 1 の通番 6($AP1 > AP2 = AP3 = AP4 = AP5$) の測定結果を図 8 に示す． 図 7 と図 8 より，以下のことがわかる．

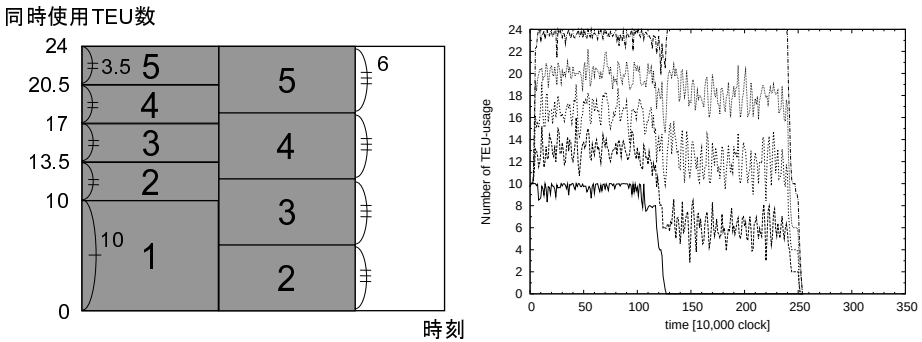
(3)　制御確率 $P = 1.0$ とし制御量 M を適切な値とすることで，サービスの同時使用 TEU 数を制御目標値に近い値に制御することができる．図 7 において，$AP1$ の実行時に $AP2$ と $AP3$ の同時使用 TEU 数は制御目標値 7 に近い値で制御する必要がある．文献 6) の評価より，(制御目標値)$\geq N_{av}$ となる最大の N_{av} は 5.80 であり，このとき $M = 10$ である．したがって，$AP2$ と $AP3$ について制御パラメータの初期値は $M = 10$ であり，図 7 の測定結果において $AP1$ 実行時に $N_{av} = 5.80$ で実行してい

Vol.2011-ARC-195 No.29
Vol.2011-OS-117 No.29
2011/4/14



(A) 理想　(B) 測定結果

図 7 3 サービス ($AP1 > AP2 = AP3$) の同時使用 TEU 数

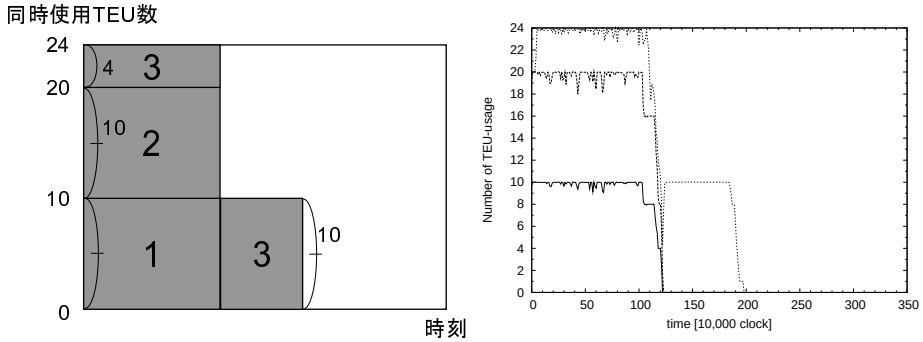


(A) 理想　(B) 測定結果

図 8 5 サービス ($AP1 > AP2 = AP3 = AP4 = AP5$) の同時使用 TEU 数

る．図 8 も同様に，$AP1$ 実行時の $AP2$ から $AP5$ の制御目標値は 3.5 で，(制御目標値)$\geq N_{av}$ となる最大の N_{av} は 2.98 である．このとき，$M = 5$ である．したがって，測定結果において，$AP2$ から $AP5$ は $AP1$ 実行時に $N_{av} = 2.98$ で実行している．

(4) 制御確率 $P = 1.0$ で制御量 M を適切な値とすることで，優先度の高いサービスの同時使用 TEU 数への影響を小さくすることができる．図 6 と図 8 で $AP1$ の同時使用



(A) 理想　(B) 測定結果

図 9 3 サービス ($AP1 = AP2 > AP3$) の同時使用 TEU 数

TEU 数の分散 $V(S)$ を比較すると，前者は $V(S) = 0.112$ で後者は $V(S) = 0.036$ であり，後者の方が小さい．

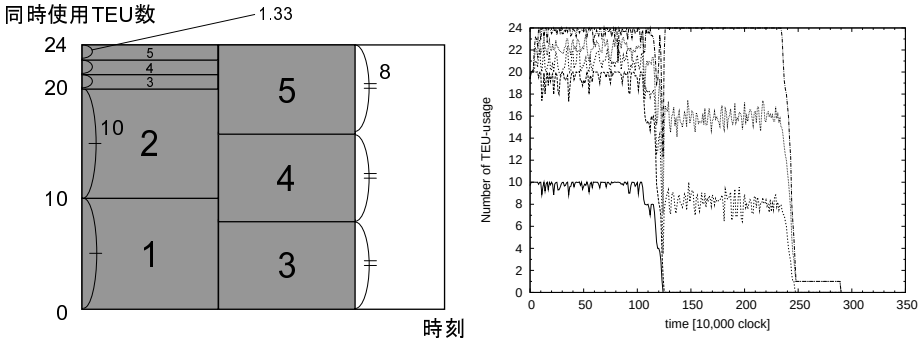
次に，2 サービスを優先実行し他サービスを均等実行する場合について，表 1 の通番 4($AP1 = AP2 > AP3$) の測定結果を図 9 に示す． また，表 1 の通番 4($AP1 = AP2 > AP3 = AP4 = AP5$) の測定結果を図 10 に示す． 図 9 と図 10 より，以下のことがわかる．

(5) 優先度の高いサービスが 2 個の場合においても，制御確率 $P = 1.0$ で制御量 M を適切な値とすることで，優先度の高いサービス群の同時使用 TEU 数への影響を小さくすることができる．図 9 では，優先度が最も高い $AP1$，$AP2$ について，それぞれ $N_{av} = 9.44$，9.37 で実行しており，スレッド並列度の 10 に近い同時使用 TEU 数で実行している．このとき，$V(S)$ はそれぞれ 0.020，0.023 で，小さい値である．また，図 10 も同様に，$AP1$ と $AP2$ について，N_{av} は 9.31，9.20 でスレッド並列度 10 に近い値であり，$V(S)$ は 0.025，0.030 で，小さい値である．

最後に，3 サービス以上を優先実行する場合について，表 1 の通番 13($AP1 > AP2 > AP3 > AP4 > AP5$) の測定結果を図 11 に示す． 図 11 より，以下のことがわかる．

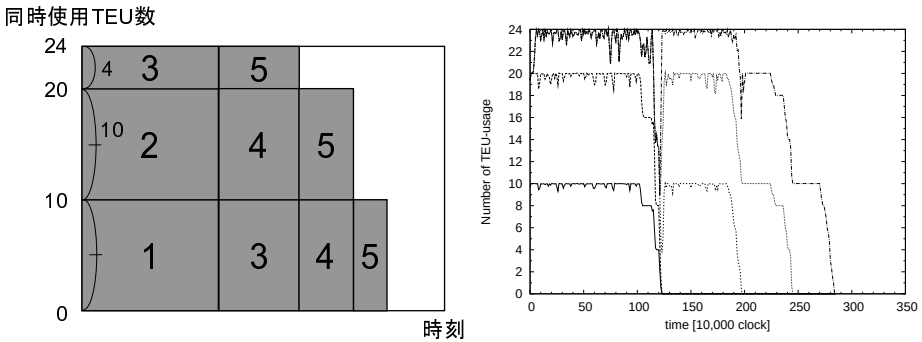
(6) 各サービスの優先度に応じて，$AP1$ から順に $AP5$ まで優先実行制御を行えている．$AP1$ は最も優先度が高く，次に $AP2$ の優先度が高い．このため，$AP1$ と $AP2$ は $P = 0$ とし，同時使用 TEU 数を抑制しない．$AP3$ は $AP2$ の次に優先度が高く，制御目標値 4 で実行する．(制御目標値)$\geq N_{av}$ となる最大の N_{av} は $M = 5$ のとき 3.58

Vol.2011-ARC-195 No.29
Vol.2011-OS-117 No.29
2011/4/14



(A) 理想　(B) 測定結果

図 10 5 サービス ($AP1 = AP2 > AP3 = AP4 = AP5$) の同時使用 TEU 数



(A) 理想　(B) 測定結果

図 11 5 サービス ($AP1 > AP2 > AP3 > AP4 > AP5$) の同時使用 TEU 数

であるため，$AP3$ は $M = 5$ とする．$AP4$ と $AP5$ は，$M = 0$ として実行しない．$AP1$ と $AP2$ の実行終了後，$AP3$ と $AP4$ は同時使用 TEU 数を抑制せず，$AP5$ は制御目標値 4 で実行する．$AP3$ の実行終了後，$AP5$ の同時使用 TEU 数を抑制せずに実行する．

5. おわりに

継続概念に基づく CMP におけるソフトウェアスケジューラについて，複数サービスの実効制御法と評価を述べた．実効制御法は，複数サービス同時実行時に，サービスの優先度にしたがって特定のサービスの優先実行や各サービスの均等実行を行うことを目的とする．各サービスの優先度に従い，各サービスの制御確率 P，制御継続発行周期 ω，および制御量 M を設定する．

評価では，複数の 10-Queens プログラムの同時実行時に，各サービスの優先度に応じて，優先実行制御と均等実行制御を行ったときの測定結果と考察を示した．同時に実行するサービス数が 2 個，3 個，5 個の場合について，サービスの優先度の組み合わせを示し，以下の観点で評価を行った．スレッド並列度の合計が TEU の総数を超過しない場合，各サービスを均等実行する場合，特定のサービスを優先実行して他サービスを均等実行する場合について，それぞれ評価を行った．測定結果の考察より，実効制御法によりサービスの優先度にしたがった制御を行うことができ，また，優先度が高いサービスの使用する TEU 数は優先度の低いサービスによって制限されることなく実行することができることを示した．

残された課題として，スレッドの実行が同期していないサービスを対象とした評価がある．

参 考 文 献

1) 谷口秀夫，日下部茂，中山大士，乃村能成，雨宮真人，“OS の処理を多く含む並列処理の効率化を指向した一括システムコール機能，” 情報処理学会論文誌，vol.44，no.SIG01，pp.81-92，2003.
2) Ben Lee, A.R.Hurson, “Dataflow Architectures and Multithreading, ” IEEE COMPUTER, vol.27, no.8, pp.27-39, 1994.
3) 日下部茂，富安洋史，村上和彰，谷口秀夫，雨宮真人，“並列分散オペレーティングシステム CEFOS(Communication-Execution FusionOS)，” 電子情報通信学会技術研究報告，vol.99，no.251，pp.25-32，1999.
4) M. Amamiya, H. Taniguchi, and T. Matsuzaki, “An architecture of fusing communication and execution for global distributed processing,” Parallel Processing Letters, vol.11, no.1, pp.7-24, 2001.
5) 雨宮聡史，松崎隆哲，雨宮真人，“排他実行マルチスレッド実行モデルに基づくオンチップ・マルチプロセッサの設計，” 情報処理学会研究報告，vol.2003，no.119，pp.51-56，2003.
6) 森山英明，乃村能成，谷口秀夫，“継続概念に基づく CMP 向け S/W スケジューラの評価，” 電子情報通信学会技術研究報告，vol.110，no.278，pp.11-16，2010.