

高並列処理におけるスケーラビリティ評価方法

折居 茂夫

富士通株式会社

逐次処理時間なしで記述した並列効率と、この並列効率の等高線を直線で可視化できる時空間を導入する。この時空間ではこの並列効率のIsoefficiencyラインが直線となるため、Grama等が定義したスケーラブルパラレルシステムを容易に同定できる。この同定方法は、高並列処理を含む全ての並列処理に適用可能である。プロセッサ数と問題の規模を変えた処理時間をこの空間にプロットすることのみで、並列効率と処理時間を考慮したスケーラビリティ評価ができることを示す。

Method of Scalability Evaluation for Highly Parallel Processing

Shigeo Orii

Fujitsu Limited

I introduce a parallel efficiency that is described without a serial processing time and, a time space which visualizes contour lines of the parallel efficiency as a straight line. Because the isoefficiency of parallel processing is also visualized as a straight line in the time space, one can easily identify a scalable parallel system as defined by Grama et al. This identification method can be applied to the whole parallel processing, especially including highly parallel processing. It is shown that scalability based on parallel efficiency and parallel processing time can be evaluated by simply plotting timing data for varying numbers of processors and problem sizes in the time space.

1. はじめに

ペタフロプスの計算能力を持つ数万台規模のプロセッサを持つ計算機が稼動する現在、高並列処理のスケーラビリティ評価の研究は、巨大な資源の有効利用という観点から益々重要になると考える。

そこでGrama等によって提唱された「スケーラブルパラレルシステム」[1]を高並列で同定する方法を研究した。この同定では並列効率を用いるので、スケーラブルであることを同定すると同時に、資源の有効利用をも計量することができる。しかし並列効率が逐次処理時間を用いて定義されているゆえ、高並列

処理においてこの同定を行うことは難しい。

そこで本論文では第一に高並列処理でも決定可能な並列効率を提案する。第二にスケーラブルパラレルシステム同定が容易となる時空間を導入し、その時空間を用いた同定方法を提案する。

この時空間におけるスケーラブルパラレルシステム同定、並列効率及び処理時間の振る舞いを分析することにより、高並列処理におけるスケーラビリティ評価ができる。

2章ではスケーラブルパラレルシステムについて述べ、3章でこの同定方法を提案する。4章では3章で提案した時空間を用いた処理時

間評価方法を示す。5章で測定値によるスケラビリティ評価例を示し、6章でまとめと議論を行う。

2. スケラブルパラレルシステム

並列処理において、問題の規模を固定してプロセッサ数を増すと、並列効率は減少する。一方プロセッサ数を増しても問題を大きくすると、並列効率が等しくなる場合がある。この状態はIsoefficiencyと呼ばれる。Grama等は、計算機とプログラムからなるシステムがIsoefficiencyとなった状態を「スケラブルパラレルシステム」と定義し、スケラビリティを評価することを提案した[1]。

Isoefficiencyの確認には2つのアプローチがある。1つは、プログラムの処理時間を問題の大きさとプロセッサ数でモデル化し、Isoefficiencyとなる問題の大きさを表す関数を導出するアプローチである[1, 2]。このアプローチは問題の規模を外挿したスケラビリティ評価ができるメリットがある。

2つ目は、測定した処理時間からIsoefficiencyを確認するアプローチである[2]。このアプローチは処理時間モデルを必要とせず、値のみでIsoefficiencyの確認ができるメリットがある。処理時間モデルがないため問題の規模を外挿した評価はできないが、高並列処理時間を予測する研究[3]から値が与えられるとすると、値のみでスケラブルパラレルシステムを同定できることに大きなメリットが見出せる。そこでこのアプローチによるスケラビリティ評価方法を研究した。

3. スケラブルパラレルシステムの同定

3.1 逐次処理時間なしで記述した並列効率

高並列でスケラブルパラレルシステムを同定するため、逐次処理時間なしで記述した

並列効率を導出する。

式(1)に従来の並列効率を示す。

$$E_p(p, n) = \frac{\tau(1, n)}{p \cdot \tau(p, n)} \quad (1)$$

ここに $\tau(p, n)$ は並列処理時間、 p はプロセッサ数、 n は問題の大きさである。

逐次処理時間 $\tau(1, n)$ を使わない並列効率を導くため、Grama等によって示された並列処理時間モデル式(2)[1]を用いる。

$$p \cdot \tau(p, n) = \tau(1, n) + T_0(p, n) \quad (2)$$

ここに $T_0(p, n)$ は並列時に発生するオーバーヘッドである。式(2)を式(1)に代入し整理すると式(3)となる。

$$\begin{aligned} E_p(p, n) &= \frac{p \cdot \tau(p, n) - T_0(p, n)}{p \cdot \tau(p, n)} \\ &= 1 - \frac{T_0(p, n)}{p \cdot \tau(p, n)} \end{aligned} \quad (3)$$

今 $T_0(p, n) = p \cdot X(p, n)$ とすると並列効率は逐次処理時間 $\tau(1, n)$ を含まない式(4)になる。

$$E_p(p, n) = 1 - \frac{X(p, n)}{\tau(p, n)} \quad (4)$$

ここで測定値より並列効率を決定することを考え、 $\tau(p, n) = \gamma(p, n) + \chi_s(n) + \chi_p(p, n)$ 、 $\chi_p(1, n) = 0$ 、 $\gamma(p, n) - \gamma(1, n)/p = 0$ とすると、 $X(p, n) = (p-1)/p \cdot \chi_s(n) + \chi_p(p, n)$ と表せる。 $\gamma(p, n)$ は並列計算部、 $\chi_s(n)$ は全てのプロセッサで同じ処理を行う冗長計算部、 $\chi_p(p, n)$ はその他の並列オーバーヘッド部の処理時間である。 $\chi_s(n)$ と $\chi_p(p, n)$ を別々に測定することは困難を伴うことを考え、 $\chi(p, n) (\equiv \chi_s(n) + \chi_p(p, n))$ と置き、式(4)の $X(p, n)$ と換えて式(5)を定義し、これをスケラブルパラレルシステム同定に使う並列効率とした。

$$\varepsilon_p(p, n) \equiv 1 - \frac{\chi(p, n)}{\tau(p, n)} = \frac{\gamma(p, n)}{\tau(p, n)} \quad (5)$$

式(5)は、導かれた並列効率が $\tau(1,n)/p$ から $\chi(p,n)$ に基準が変わった効率であることを示す。

3.2 Isoefficiencyラインが直線となる空間

ここでは3.1節で導出した並列効率 ε_p を評価する空間を示す。式(5)を変形すると式(6)を得る。そこで τ を横軸、 χ を縦軸にした座標系（以後 τ - χ 時空間と呼ぶ）を設ければ、 ε_p のIsoefficiencyラインは原点を通る直線となる。

$$\chi(p,n) = (1 - \varepsilon_p(p,n)) \cdot \tau(p,n) \quad (6)$$

τ - χ 時空間のスケールを決定し、1点をプロットするとそのIsoefficiencyラインは一義的に決まる。座標軸をLogスケールにした τ - χ 時空間を図1に示す。LogスケールにおいてもIsoefficiencyラインは直線である。またLogスケールでは問題規模が異なり処理時間の桁が異なる現象を一つの図に収めて比較することが可能となる。5章にこの例を示す。

表-1 τ - χ 空間における並列効率の状態とスケラブルパラレルシステム(SPS)の関係

ステータス	効率の状態	図1中の場所
SPS	$\varepsilon_p(p,n) = \varepsilon_p(p',n')$	破線と(□→△)線の交点
SPS候補	$\varepsilon_p(p,n) < \varepsilon_p(p',n')$	領域(I)
否SPS	$\varepsilon_p(p,n) > \varepsilon_p(p',n')$	領域(II)

3.3 スケラブルパラレルシステム同定方法

スケラブルパラレルシステム（以後、**SPS**と記す）は、Isoefficiency即ち $\varepsilon_p(p,n) = \varepsilon_p(p',n')$ を確認して同定する。ここに $p < p'$ 、 $n < n'$ である。表-1の3つの状態を次の手順で確認する。

- 1) 着目した点($\tau(p,n)$, $\chi(p,n)$)、今後(p,n)と記す、を τ - χ 空間にプロットする（図1）。
- 2) プロットした点と原点を結びIsoefficiencyラインを描く（図1の破線）。

- 3) 問題の規模を n から n' に変化させた測定データ(p,n')をプロットする。

➤ もしプロットした位置が図1の破線の下領域(I)にある、即ち○から□を結ぶ矢印のように遷移するなら、ステップ4に進む。

✓ もしプロットした位置が図1の破線の上領域(II)にある、即ち○から△を結ぶ矢印のように遷移するなら、

Isoefficiencyは存在せず**SPS**ではない。

- 4) 問題の規模を n' に固定し、プロセッサ数を p から p' に増加する。

➤ もし $\varepsilon_p(p,n) = \varepsilon_p(p',n')$ となれば、**SPS**と同定できる。

➤ あるいは(p,n')と(p',n')を結ぶ線が(p,n)のIsoefficiencyラインを横切れば**SPS**と同定できる。

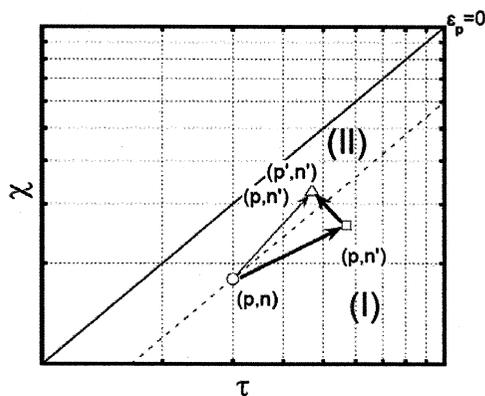


図1 τ - χ 時空間におけるIsoefficiencyを用いたスケラブルパラレルシステム(SPS)の同定。破線は(p,n)のIsoefficiencyライン。

4. τ - χ 時空間における処理時間の評価

τ - χ 時空間では、並列効率と同時に処理時間を評価できる。図2に**SPS**と同定されたIsoefficiencyライン上の1点から、問題の大きさを固定してプロセッサ数を増したときの処

理時間の変化を示す。□に着目してプロセッサ数を増すと、 τ が減少する領域(C1)、変化しない場合(C2)及び、増加する領域(C3)が存在する。もし領域(C3)なら、この点はSPSであるがプロセッサが冗長に使われている或いは、プロセッサ数を増すとき冗長に使われる可能性があることを意味する。従ってスケーラビリティ評価では、この点が領域(C1)にあることの確認が重要となる。

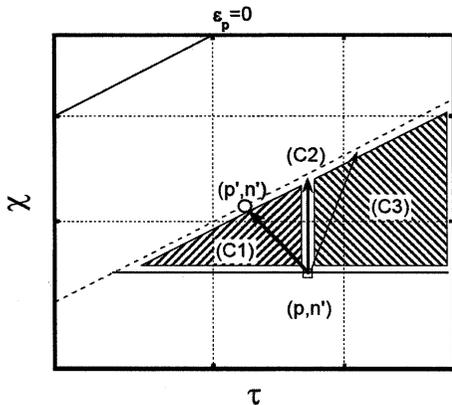


図2 τ - χ 空間の処理時間変化の分類(n' =一定). 領域(C1): τ 減少, 領域(C2): τ 無変化, 領域(C3): τ 増加.

5. スケーラビリティ評価例

τ - χ 空間に測定値をプロットし、3章で示したSPSの同定と、4章で示した処理時間の領域(C1)を確認し、このSPSがプロセッサを有効に使用しているかを確認したスケーラビリティ評価例を、並列計算機SP2で分子動力学プログラム[4]を実行した測定結果[5]を用いて示す。ここに測定値は τ と χ , χ は $\chi = \tau - \gamma$ で求めた。

(a) 3点の測定値によるSPS同定

SPSは3点の測定値で同定できる。以下に3.3節の方法に沿った評価を示す。

- 1) τ - χ 空間に点○の($p=10, n=39200$)をプロットする(図3-a).

- 2) ($p=10, n=39200$)と原点を結ぶIsoefficiencyラインを描く. 図3-aにおいてこれは $\epsilon_p=0.552$ のラインである. このラインより下側が領域(I), このラインと $\epsilon_p=0$ に挟まれた部分が領域(II)となる.
- 3) プロセッサ数が同じで問題の規模 $n=80000$ の処理時間は□となる. この□は領域(I)に在る故, SPS候補である.
- 4) 問題の規模 $n=80000$ でプロセッサを増加すると, $p=16$ で ϵ_p が0.552を横切り小さくなる. したがって($p=10, n=39200$), ($p=10, n=80000$)と($p=16, n=80000$)の3点によりSPSが同定された.

このSPSでは $\epsilon_p=0.552$. Isoefficiencyライン上の $n=39200$ の点で、このプログラムが約55%の効率で処理されている。これをプロセッサ数に $p \cdot \epsilon_p$ で換算すると約5.5台分に相当する。

次に処理時間を評価する。 $n=80000$ でプロセッサ数増加した場合、 $\epsilon_p=0.552$ のラインを横切る $p=10$ から $p=16$ への矢印は負の傾きを持っており、並列処理は図2の領域(C1)にあるので、プロセッサを冗長に使用していないこと、プロセッサが増すと処理時間が減る場合があることがわかる。

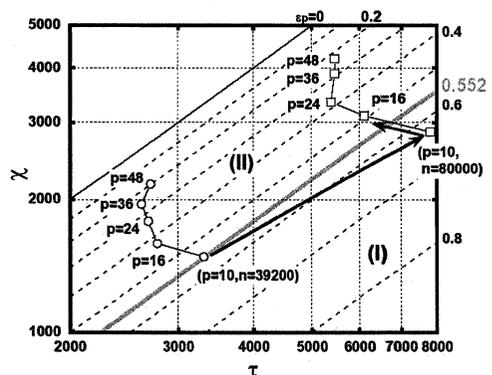


図3-a 3点の測定値によるスケーラブルパラレルシステム同定

(b) SPS であるがプロセッサ数を増すと処理時間が増える場合

今度は図3-aの $p=24$, $n=39200$ に着目し, このSPS同定を図3-bに示す. 問題の規模 $n=80000$ で効率が向上し, この点で p を24から36に増せば $n=39200$ のIsoefficiencyラインを横切る. したがってSPSと同定できる. 一方この横切る矢印の傾きは正なので, 並列処理は図2の領域(C3)に当たる. したがってこのシステムはSPSであるが, $\epsilon_p=0.334$ のIsoefficiencyラインでプロセッサを増すと $n=80000$ では処理時間が増え, プロセッサが冗長に使われる.

(c) システムのスケラビリティ評価

τ - χ 時空間に多数の測定値をプロットすることにより, 計算機とプログラムからなるシステムのスケラビリティ評価ができる.

図3-cに n と p を変えた測定値を示す. 図中の矢印は $p=10$ に対して問題の大きさを変えた点をつないだものである. Isoefficiencyラインとこの矢印を比較すると, 並列効率は問題が大きくなると $\epsilon_p=0.5$ をまたいで向上する. そこで各々の問題の規模において p を増加すると

各々のIsoefficiencyラインを横切るので, このプログラムはSP2に対して, $n=7200$ から80000においてSPSと同定できる.

同様に図3-cの破線で示したIsoefficiencyの等高線とプロセッサ数 $p=24, 36$ における測定値を比較すると, これら全ての測定点からもSPSであると同定できる.

このようにシステムのSPSは, τ - χ 時空間に多数の測定値をプロットするだけで同定できる.

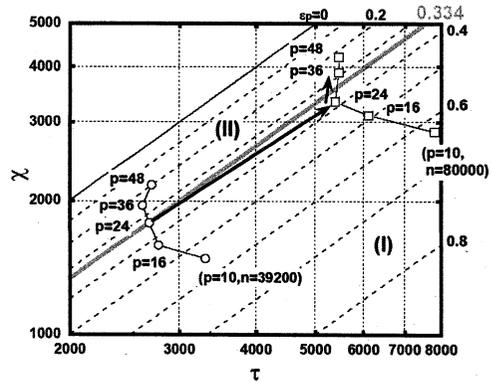


図3-b $n=80000$ でプロセッサを増すと処理時間が増える Isoefficiencyライン($\epsilon_p=0.334$)

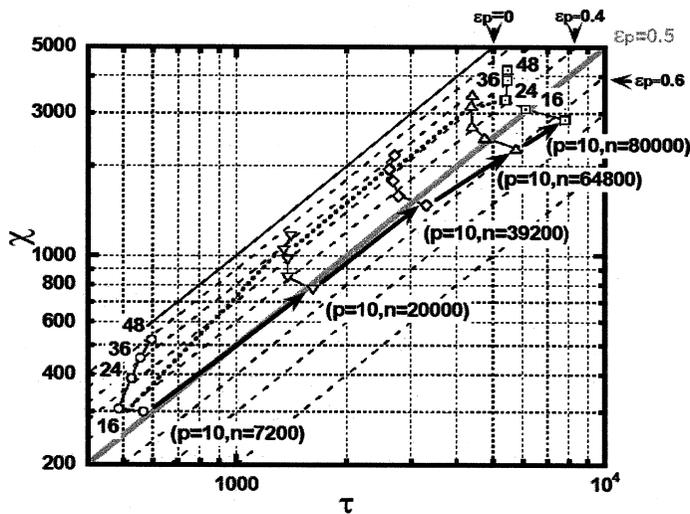


図 3-c SP2 における分子動力学プログラム[4]のスケラビリティ評価

図 3-c に処理時間の増減に関する領域(C1)と(C3)の境界を太点線でプロットした。(C1)はこの点線の下側で、この領域は問題の規模が $n=7200$ から 80000 と 1 桁変わっても $p=24$ が上限となる。また $n=80000$ において $\varepsilon_p=0.5$ を実現できるプロセッサは $p=16$ である。従って、SP2 と分子動力学プログラムから成るこのシステムは SPS であるが、低並列処理向きのシステムであると評価できる。

6. まとめと議論

逐次処理時間なしで記述した並列効率を導出し、その等高線図が直線となる τ - χ 時空間を用いたスケーラビリティ評価方法を示した。

τ - χ 時空間の特徴の1つは、図1に見られるように、並列処理による処理時間の下限が、 $\varepsilon_p=0$ のライン上の時間として可視化されることである。また τ - χ 時空間では、並列効率と処理時間の評価を同時に行うことができる。従って図4のように並列効率を計算して再プロットすること無しに、スケーラビリティ評価ができる。さらに最小処理時間の判別において、図4に示した通常用いられる p - τ プロットでは不鮮明な場合でも、図3-cの τ - χ 時空間では大きな変化となり容易に判別することができる。

このような τ - χ 時空間に並列計算機とプログラムを組み合わせ、多数の測定値をプロットすることは、性能評価のカルテを作成することに相当すると考える。本論文では実測値を用いたが、予測処理時間値をプロットしたスケーラビリティ評価もできる。並列性能情報を τ - χ 時空間に記述して蓄積・比較することで、アルゴリズム、アーキテクチャの研究開発に役立つものと期待する。

謝辞 富士通株式会社テクニカルコンピューティング・ソリューション事業本部の奥田基

エグゼクティブアーキテクトの援助に感謝します。

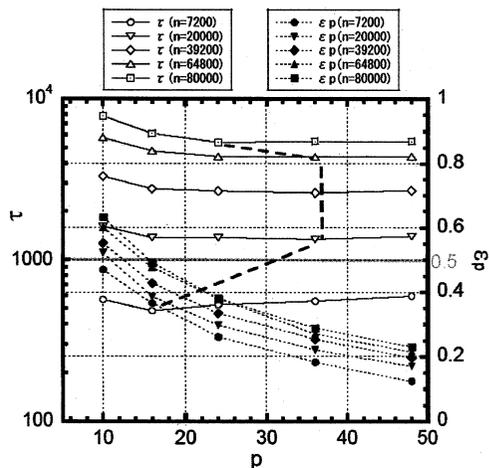


図4 図3-cのデータを p - ε_p と p - τ でプロット。太点線は領域(C1)と(C3)の境界。

参考文献

- [1] A. Y. Grama, et. al., Isoefficiency: Measuring the scalability of parallel algorithms and architectures, IEEE Parallel & Distributed Technology Vol. 1, No. 3, (1993), 12-21.
- [2] V. Kumar, et. al., Parallel Depth First Search, Part II: Analysis, International Journal of Parallel Programming Vol. 16, No. 6, (1987), 501-519.
- [3] 例えば, R. Susukita, et. al, Performance Prediction of Large-scale Parallel System and Application using Macro-level Simulation, In proceedings of SC2008.
- [4] T. Watanabe, et. al., Increase in Chaotic Motions of Atoms in a Large-Scale Self-Organized Motion, Phys. Rev. E, 54 (1996), 1504-1509.
- [5] 折居茂夫, レベル1・2並列ベンチマーク仕様及びそれに基づくスカラ並列計算機SP2のベンチマークテスト, JAERI-Data/Code 98-020 (1998).