3K-6

Dynamic resource planning for grid computing system based on capacity planning

野々田 峰寛† 小野寺 直歌† 福田 裕大† 中村 太一†
東京工科大学†

1. はじめに

異なる性能の多数のパソコン（PC）で構成される Grid computing(Grid)で実行される業務処理（以下 Grid アプリケーション）の実行時間は並列化率と分散処理に使用する PC 台数に関係する。また、PC を追加するとオーバヘッド処理時間が増加するため、PC の追加が Grid アプリケーションの高速化に繋がるとは言えない。本研究では、マスタ PC がジョブを一括または順にスレーブ PC に送りバッチ処理をするモデルとスレーブ PC に対してマスタがオンライントランザクション（OLTP）処理をするモデル[2]を検討対象とし、Grid アプリケーション実行時間を MPI 関数の初期化時間、PC に分散したジョブの処理時間および処理結果の結合時間の合計とする。MPI_Init 関数処理時間をスレーブ PC 台数の関数で表し、PC に分散したジョブの処理時間はキューに滞留するジョブ数を利用して求める。MPI_Init 関数処理時間と、PC に分散したジョブの処理時間を比較し Grid アプリケーション実行時間が最短となるスレーブ PC 台数を見積もる。最後に、Globus Toolkit と MPICH-G2 を用いて Grid システムを構築してモデルを評価する。

2. オーバヘッドの把握

オーバヘッドは Grid システムを構築するミドルウェアが分散処理を実現するための処理である。Grid アプリケーションが MPI 実行環境を初期化するための MPI_Init 関数処理時間の測定結果[2]から、スレーブ PC のクロック数を c_i (MHz)とし、MPI 初期化時間 *I(n)*を実験式(1)で表す。

$$I(n)=\sum_{i=1}^{n}\frac{917}{c_i}. \quad (400 \leqq c \leqq 3000) \qquad (1)$$

実験式(1)による計算値と、実測値を図 1 に示す。PC が全て同じ性能の場合、誤差は約 5%で一定、性能が異なる PC の場合、最大誤差は 30%である。

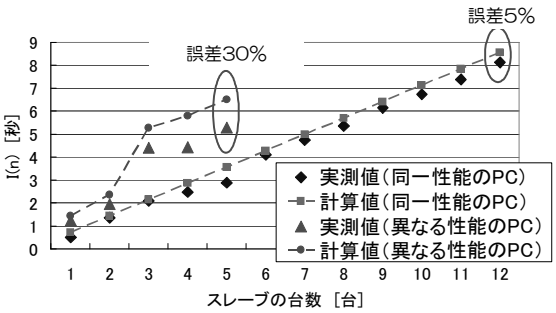

図 1 MPI_Init 関数処理時間

3. Grid アプリケーション実行時間見積もり

Grid アプリケーション処理時間を MPI_Init 関数処理時間、分散したジョブの処理時間、処理結果結合時間の合計とする。結合時間は 10 マイクロ秒と極めて小さいので無視する。図 2 はバッチ処理型のモデルで測定したスレーブ PC のキューに格納されるジョブ数の測定結果である。スレーブ PC 台数が 1 台から 4 台の時、到着率とサービス率の大小関係は $\lambda > \mu$ となり[2]、キューに格納されているジョブの減少を表す直線を延長すると、グラフの縦軸でスレーブ PC1 台当たりが処理をするジョブ数と交わることから、時間 t とスレーブ PC 全体のサービス率 μ を用いて、$-\mu t$ と表す。スレーブ PC 台数が 5 台以上の場合、到着率とサービス率は $\lambda < \mu$ となり[2]、到着率は PC に分散した処理時間に影響する。

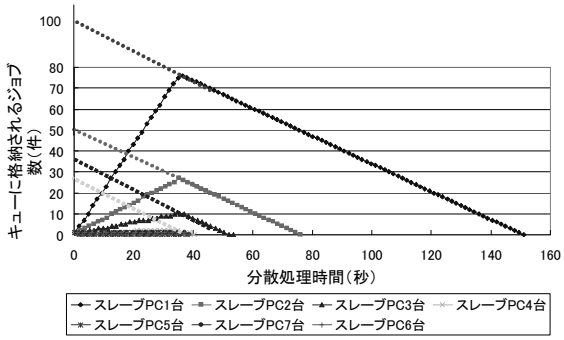

図 2 バッチ処理モデルのキューの状態

このモデルの実装では、ジョブの送信に要する時間を表す到着率の値は一定のため、分散されたジョブの処理時間はジョブ送信時間に収束する。この

†Takahiro NONODA, †Naoka ONODERA,
†Yuta FUKUDA, †Taichi NAKAMURA
Tokyo University of Technology

結果からスレーブへの到着率λ、スレーブのサービス率μ、Grid アプリケーションの分割数をジョブ数 M とすると、PC に分散したジョブの処理時間は式(2)、(3)で見積もることができる。

$$D(\lambda,\mu,M)=\frac{M}{\mu} \quad (\lambda>\mu) \qquad (2)$$

$$D(\lambda,\mu,M)=\frac{M}{\lambda} \quad (\lambda<\mu) \qquad (3)$$

スレーブ PC のサービス率から、スレーブのサービス率μを求める。スレーブが n 台の PC で構成される時、サービス率を固有かつ降順に並び替えた $\vec{\mu}'$ と、μ_i に対応するPC数 $\vec{\varpi}$ 、$\vec{\mu}'$ 要素の逆数 $\vec{S}$ を式(4)に代入して、サービス率を導出する。

$$\mu=\frac{S_1}{S_m}\vec{\mu}\bullet\vec{\varpi}+\sum_{i=2}^{m}\left(\frac{S_i-S_{i-1}}{S_m}\sum_{j=i}^{m}\mu'_j\varpi_j\right) \qquad (4)$$

式(4)は、バッチ処理型モデルで $\mu_1=\mu_2=\cdots=\mu_n$ の時とOLTP 型モデルの時式(5)と表せる。

$$\mu=\sum_{i=1}^{n}\mu_i \qquad (5)$$

図 3 に、バッチ処理型モデルおよび、OLTP 型モデルの PC に分散したジョブの処理時間の見積もりと実測値を示す。バッチ処理型モデルでスレーブ PC が同一性能の場合、スレーブ PC1 台から 4 台では式(2)を、5 台以降は式(3)を利用した。スレーブ PC が異なる性能の場合、今回の測定では式(4)で求めたサービス率を到着率と比較した結果、式(2)を利用した。OLTP 型モデルは式(3)を利用した。どちらも見積もりの誤差は約 5%程度であった。

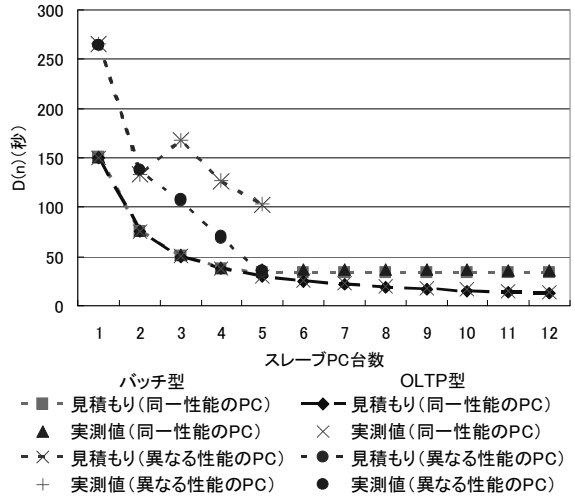

図 3 PC に分散したジョブの処理時間の評価

4. Grid アプリケーション実行時間を最短とするスレーブ台数の見積もり

Grid アプリケーション実行時間の短縮量が 0 になる時を最短 Grid アプリケーション実行時間とし式(6)と表す。Grid アプリーション実行時間 *T(n)*は初期化時間の関数 *I(n)*と、PC に分散したジョブの処理時間の関数 *D(n)*の合計 *T(n)=I(n)+D(n)*を式(6)に代入し式(7)を得る。

$$T(n)-T(n+1)=0 \qquad (6)$$

$$D(n)+I(n)-D(n+1)-I(n+1)=0 \qquad (7)$$

式(7)から求めたバッチ処理型モデル、OLTP 型モデル上で実行した Grid アプリケーション実行時間の短縮量を図 4 に示す。この結果からバッチ型モデルでは 5 台の時最短 Grid アプリケーション実行時間となり、測定結果と一致した。OLTP 型モデルは式(7)が 0 にならないため、更に PC を追加することで Grid アプリケーション実行時間を短縮できる。

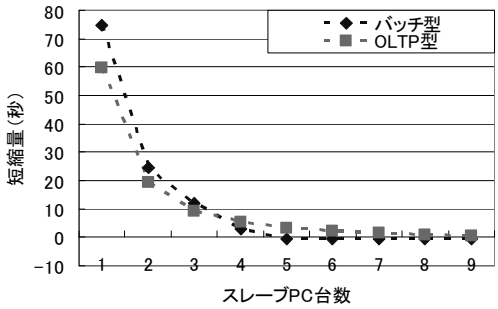

図 4 Grid アプリケーション最短実行時間見積もり

5. おわりに

本稿では、Grid システムのオーバヘッドの所在と、スレーブ PC 台数とその処理時間の関係と Grid アプリケーション実行時間の見積もり関数を示した。今後は異なる性能の PC で構成されるスレーブのオーバヘッド処理時間を見直し、ネットワークやディスクアクセスを加えて Grid アプリケーション実行時間を見積もる。

参考文献

[1] Daniel A.Menasce and Virgilio A.F. Almedia, “Capacity planning for Web Performance”, Prentice Hall PTR, 1998

[2] 坂巻 雅実, 野々田 峰寛, 伊藤 剛, 中 健一, 中村 太一, Grid Computing の性能見積もり手法の研究, 第 68 回情報処理学会全国大会講演論文集(1), pp. 93-94, 2006.

[3] Ian Foster, Carl Kesselman：The Grid2:Blueprint for a New Computing Infrastructure, 2003