2P−5　分散並列OS「Orion」の試作 システムの性能評価

吉澤聡，岩嵜正明，千葉寛之，宇都宮直樹，薗田浩二，山内雅彦
(株)日立製作所　中央研究所

1. はじめに

メッセージ通信に基づく分散並列OS「Orion」のプロトタイプは、現在ネットワーク接続したワークステーション上で稼働している。本稿ではSPMD(Single Program Multiple Data stream)型アプリケーション・プログラムを用いて行ったOrionシステム・プロトタイプの評価について報告する。

2. 評価アプリケーションの概要

(1) 目的

Orionシステム開発の一環として、同システム上で動作する評価アプリケーションを作成した。評価アプリケーション作成の目的は以下を実施し、またその結果をOrionシステムの設計にフィードバックすることにある。

(a) Orionシステムの基本機能の確認

(b) 外部インタフェース(API)仕様の確認

(c) 問題点の抽出（特に、性能面での）

(d) Orionシステムの第一次性能評価

本稿では、上記(d)を中心に報告する。

(2) 問題の概略

評価アプリケーションとして、2次元Poisson方程式をGauss-Seidel法[1,2]で解く問題を取り上げた。本問題の並列化は2次元メッシュ状の離散データを空間的に等分割し、反復演算処理を分割したメッシュ毎に異なるノード上で並列に実行することによって行う。また各反復演算毎に、メッシュの境界データを、隣接するメッシュの演算処理を行うノードとの間でメッセージ通信により交換する。この問題は、各ノードに割り当てたメッシュの一辺のデータ点数をNとすると、演算量のオーダN^2に対し、ノード間の通信量がオーダNとなる*。尚、全てのプロセスの生成は、アプリケーション・プログラムの初期化処理部分で実行しておく。

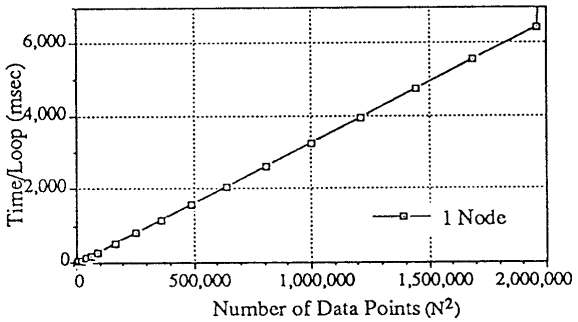

図1. 総データ数 vs. 1ループ当たり処理時間

Orionシステム・プロトタイプの性能評価は、問題規模Nとノード台数をパラメータとし、評価アプリケーションの実行時間を計測することによって行った。同問題を1ノードで処理した場合の実行時間を図1に示す。

図1は、総データ数（2次元メッシュ上のデータ点数）に対して、反復演算ループの1ループを処理するのに要する処理時間をグラフ化したものである。総データ数が約2,000,000（1400×1400のメッシュに相当）以下の領域では、1ループ当たりの処理時間は総データ数に比例する。但し総データ数が約2,000,000を越えると、ページングが発生し実行性能が極端に低下する。

* 例えば1000×1000のメッシュ(1点当たり8Byte)を4ノードに分割した場合、各ノードに割り当てられるデータ量は2MByteに、また1ループ当たりのデータ送信量は8kByteとなる。各ループでは、割り当てられた全てのメッシュ点について、それぞれ四近傍点との平均値を求める演算を行う。

(3) インプリメンテーション

並列版評価アプリケーションは、Orionシステムの提供する以下の機能[3]を使用して作成した。

- プロセスサーバ(PS)による、プロセス起動。
- メッセージサーバ(MS)を介した、メッセージ通信。
- ポートに付与したユーザ定義名(UDN)による通信相手の指定。

即ち、メッセージ通信に関してアプリケーション作成時に意識する必要があるのは、ポートに付与したユーザ定義名のみである。そのため各プロセスが実行時にどのノードにマッピングされるか等をコーディング時に意識する必要はない。

3. システム性能の評価結果

上記並列版評価アプリケーションを用い実行時間の測定を行った。尚メッセージサーバについては、ロケーション情報をサーバ内部にキャッシングするバージョン[4]を使用した。2及び4ノードについて、測定結果を図2に示す。

総データ数200,000（450×450のメッシュに相当）以上の領域で、4ノードに問題を分割して並列処理を行うことによって、1ノードで実行する場合の3.5倍程度の性能が得られている。またノード当たりのデータ量が小さくなるので、ページングの発生点もほぼノード数倍に延びる。

但し、総データ数40,000（200×200のメッシュに相当）以下の領域では、メッセージ通信のオーバヘッドによって1ノードで処理を行った方が高速である。

Prototyping of Distributed-Parallel Operating System "Orion"
- Evaluation of System Performance -
Satoshi YOSHIZAWA, Masaaki IWASAKI, Hiroyuki CHIBA,
Naoki UTSUNOMIYA, Kouji SONODA, Masahiko YAMAUCHI
Central Research Laboratory, Hitachi, Ltd.

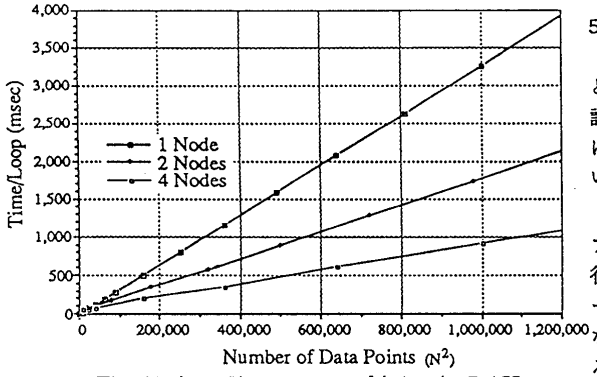

図2. 総データ数 vs. 1ループ当たり処理時間

図3に、4ノード並列化時の処理時間の内訳（通信時間・演算時間）を示す。

4. ポーリング間隔とシステム性能

メッセージサーバ(MS)では、メッセージ到着を検知するために、受信ポートに対してポーリングを行っており、その処理負荷がアプリケーションの実行性能に影響を与える可能性がある。

そこでメッセージサーバのメッセージ受信検知ポーリング間隔を変えながら（他の設定は図2のデータ測定時と同一）4ノードを用いて並列版評価アプリケーションの実行時間を測定した。各ノードに500×500のメッシュを割り当てた場合（総データ数＝1,000,000）について、測定結果を図4に示す。

ポーリング間隔を短縮していくと、演算処理開始から終了までの時間が長くなる。これはメッセージサーバがポーリング処理を実行するためにCPU資源を消費し、アプリケーションのCPU使用率が低下するためである。しかしながら、逆にポーリング間隔を延ばし過ぎると、メッセージが到着してからそれを検知する迄の時間が長くなり、実効性能は低下する。本例に於けるメッセージサーバのメッセージ受信検知ポーリング間隔の最適値は10msec程度である。

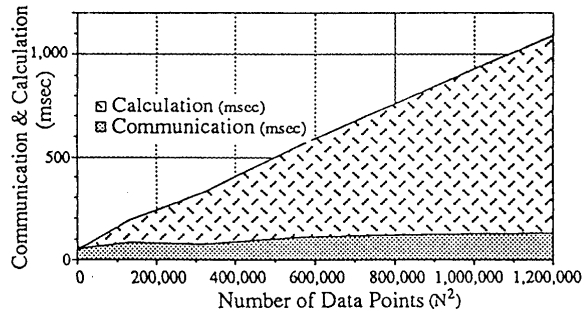

図3. 総データ数 vs. 1ループ当たり処理時間の内訳
（4ノード時：通信時間と演算時間）

5. おわりに

Orionシステムでは、機能分散型のマルチ・サーバ構成をとるが、これはオーバヘッド増加要因となり得る。今回、評価アプリケーションを用いたシステムの第一次性能評価について報告した。今後、各サーバのオーバヘッド等について、より詳細な分析を行っていく予定である。

また本稿では「静的」なスケージューリングが可能なアプリケーションを取り上げたが、今後アプリケーション実行中に動的なプロセス生成を行ったり、通信相手や通信データ長、通信頻度が実行と共に変化していく様な「動的」なアプリケーションにも評価の範囲を拡げていく予定である。その結果を、システム設計にフィードバックすると共に、Orionシステムの提供する機能に基づく並列プログラミング手法の具体化を進めていく。

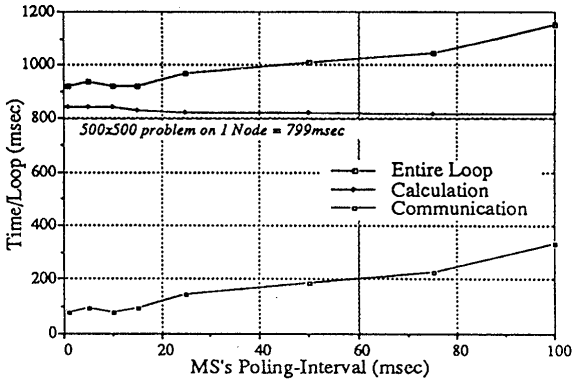

図4. MSメッセージ受信検知ポーリング間隔の影響

参考文献

[1] Hoshino,T., Kawai,T., et al., "PACS: A Parallel Microprocessor Array for Scientific Calculations", ACM Transactions on Computer Systems, Vol.1, No.3, Aug. 1983.

[2] Fox,G., Johnson,M., et al., "Solving Problems on Concurrent Processors", Vol.1, Prentice Hall, 1988.

[3] 岩崎, 他, 「分散並列OS [Orion] の試作－システムの概要」, 情報処理学会第45回全国大会予稿集, 2P-01, 1992.

[4] 薗田, 他, 「分散並列OS [Orion] の試作－メッセージサーバの実装」, 情報処理学会第45回全国大会予稿集,