平成 24 年度 RICC 利用報告書

課題名（タイトル）：

大規模並列計算機の効率的な利用・運用に関する研究

利用者氏名：○南里　豪志*, 稲富　雄一*
所属：*情報基盤センター

１．　本課題の研究の背景、目的、関係するプロジェクトとの関係

並列計算機をバッチ型で利用する場合、利用者から申請されたジョブがジョブスケジューラによって計算ノードに割り当てられる。この時ジョブスケジューラは、予め設定されたスケジューリングポリシーに従って、プロセスの配置を決定する。従来、多くのジョブスケジューラは、計算ノードをジョブクラスと呼ばれる大小の連続ブロックに分割し、それらのジョブクラス単位でプロセスを割り当てていた。これに対し、理化学研究所の RICC(RIKEN Integrated Cluster of Clusters) のメタスケジューラでは、出来るだけ空いている計算ノードが少なくなるように、計算ノードの位置に関係なくプロセスを配置する。その結果ジョブの充填率が向上するため、並列計算機全体のスループットも向上することが期待できる。一方、個々のジョブにおけるプログラム中の通信の性能は、そのジョブに割り当てられた計算ノードの相対的な位置関係によって大きく影響を受ける。例えば計算ノード同士の距離によって通信遅延時間が変動する。さらに、距離が遠い場合、他の通信と経路が競合する通信衝突が発生し、通信帯域幅が低下する可能性が高くなる。

特に、並列プログラムにおける全プロセスによる総和の計算や全対全のコピー等の、集団通信と呼ばれる通信では、この転送性能の低下が大きな問題となる。集団通信には複数の実装アルゴリズムが用意されており、状況に応じて最適なアルゴリズムを選択して使用する。従来は、事前に十分な調査を行うことにより、使用するプロセス数と転送するメッセージサイズから最適なアルゴリズムを選択している。しかし通信性能が安定しない環境では、同じプロセス数、メッセージサイズでも事前の調査時とは最適なアルゴリズムが変化する可能性が高い。そのため、実行時の状況に応じて適応的にアルゴリズムを選択する仕組みが必要となる。

そこで本課題では、集団通信の実装アルゴリズムを実行中に試しながら、最適なものを選択する動的最適化技術の研究開発を行っている。この最適化技術は、プログラム中で繰り返し呼ばれる集団通信について、その最初のほうの繰り返し時に実装アルゴリズムを変えながら実行することにより、実行時の状況における各アルゴリズムの性能を取得し、その情報に基づいて最適なアルゴリズムを選択するものである。この手法の問題として、他のアルゴリズムに対して非常に遅いアルゴリズムも試すことによるオーバヘッドの増大がある。そこで、アルゴリズムの性能予測モデルを作成し、それを用いて選択対象のアルゴリズムを絞り込む技術を開発した。本年度は、この技術について、アルゴリズム性能予測の精度を検証した。

２．　具体的な利用内容、計算方法

本研究で提案する動的アルゴリズム選択手法では、各集団通信アルゴリズムについて、トポロジとランク配置に応じた性能予測を行う。この、ランク配置を考慮する重要性を検証するため、予備調査として、ランク配置による集団通信アルゴリズムの性能への影響を計測した。

実験で用いるランク配置のパターンは、RICC のインターコネクトネットワークのトポロジを考慮して決定した。RICC のインターコネクトネットワークは 2 階層のスイッチ群による Fat Tree トポロジで構成されており、下位の 52 台の Leaf Switch が上位の 2 台の Upper Switch と 2 本ずつ、合計 4 本のリンクで接続されている。この Leaf Switch に 20 台ずつの計算ノードが接続され、合計で 1024 ノードの PC クラスタを構成している。各ノードには一意に番号が付けられており、0 番目の Leaf Switch には 0～19、1 番目の Leaf Switch には 20～39 というように、Leaf Switch 内に連続した番

号のノードが配置されている。一方、各 Leaf Switch から Upper Switch への 4 本のリンクには、0～3 の番号が付けられている。このトポロジにおける Leaf Switch を跨る通信では、Leaf Switch から Upper Switch へのリンクのうち、その通信の宛先ノードの番号を 4 で割った余りの数字と一致する番号のリンクを経由して、目的のノードにメッセージが送信される。

このようなネットワークの特性を考慮し、全プロセスが同時に通信を行う際の平均帯域幅を計算し、その値を用いて各アルゴリズムの性能を予測する。この平均帯域幅は、あるプロセスと他の全プロセスの間で通信を行う際の個々の通信の帯域幅の平均値とする。ここで個々の通信の帯域幅としては、それぞれの通信時に経由する各リンクの基本帯域幅を、そのリンクを同時に使用する他のプロセスによる通信の数の期待値で割った値を用いる。今回対象とする RICC のトポロジでは、Leaf Switch と Upper Switch の間のリンクにおいて、同時に使用するプロセス数が方向によって異なる。Leaf Switch から Upper Switch への方向は、その Leaf Switch から他の Leaf Switch への通信のうち、宛先ノード番号を 4 で割った値がそのリンク番号と一致するものが使用する。一方 Upper Switch から Leaf Switch への方向は、他の Leaf Switch からこの Leaf Switch への通信のうち、宛先ノード番号を 4 で割った値がそのリンク番号と一致するものが使用する。このうち、今回は、Upper Switch から Leaf Switch への方向の平均帯域幅を用いる。この平均帯域幅は Leaf Switch と Upper Switch の間のリンク毎に変動するため、全てのリンクについて計算した後、全リンクで最も小さい値を、システム全体の通信を律速する平均帯域幅として用いる。

次に、各リンクの平均帯域幅の算出方法を示す。ここで、ノード数を Nnode、ノード内プロセス数を Pn、i 番目の Leaf Switch を LSi、i 番目の Leaf Switch 内と Upper Switch の間のリンクのうち j 番目のものを ULij、プログラムが使用するノードのうち LSi に配置されたものの数を Ni、そのうちノード番号を 4 で割った値が j であるものを NLij とする。また、今回はノード内、Leaf Switch 内、Leaf Switch 間、それぞれ基準となる帯域幅は一定値 B として、平均帯域幅を計算する。

まず、LSi のノードの一つに割り当てられたプロセスが他の全プロセスから受信する場合の各通信の帯域幅を見積もる。ノード内のプロセスから受信する場合の帯域幅は、他の通信に妨げられないと仮定し、B のままとする。一方、Leaf Switch 内のノード間通信は、Leaf Switch からノードへの 1 本のリンクをノード内の全プロセスの受信で共有するため、平均帯域幅は B/Pn とする。また、Leaf Switch を跨ぐ通信の受信では、同じ Leaf Switch 内の各プロセスの通信のうち、同じリンクを経由して他の Leaf Switch から受信するものが 1 本のリンクを共有する。このプロセスが配置されたノードの番号を 4 で割った余りが j である場合 ULij を使って受信するので、同時に同じリンクを使って受信する LSi 内のプロセス数の期待値 RCVij は以下で計算できる。

```
RCVij = 1 + (NULij * Pn - 1) * (Nnode - Ni)
 * Pn / (Nnode * maxn - 1)
```

これらより、ULij を経由して受信する LSi 内のプロセスについて平均帯域幅 BRij を、以下で計算する。

```
  BSij = B/((Pn - 1 + (Ni-1) * Pn * Pn
 + (Nnode - Ni) * RCVij)
/ (Nnode * Pn - 1))
```

これを、0≦i<使用スイッチ数、0≦j<4、の各 i、j で計算し、その最小値をシステムの平均帯域幅とする。

各アルゴリズムの性能は、上記で得られた平均帯域幅をそれぞれの性能モデルに適用して予測する。今回の実装に用いた Alltoall の各アルゴリズムの性能モデルを以下に示す。

これらのモデルは、Hockney モデルによる一対一通信の性能モデルをもとにしている。Hockney モデルでは、個々の一対一通信の所要時間を、遅延時間 L とバイト当たりの所要時間 B、すなわち帯域幅の逆数を用いて L + M*B と表す。これを、各アルゴリズム内の一対一通信に適用して、性能モデルを作成した。

このモデル自体は、通信の衝突による影響が考慮

されていない。しかし、前述の通り帯域幅をプロセスの配置に応じて調整することにより、衝突の影響を加味した所要時間を見積もることができる。

3． 結果

前節で説明した予測手法をもとに、各アルゴリズムの性能を予測した結果を図 1 に示す。

Algorithmf	Measured (msec)	Predicted (msec)	Candidate?
Simple	1.352	0.606	Yes
Pair	1.974	0.606	yes
Ring	1.865	0.606	yes
Pair_light_barrier	2.650	0.776	yes
Ring_light_barrier	2.310	0.776	yes
Pair_one_barrier	1.912	0.661	yes
Ring_one_barrier	1.989	0.661	yes
Pair_mpi_barrier	6.432	2.366	no
Ring_mpi_barrier	6.551	2.366	no
Bruck	3.053	1.150	yes

図 1 各アルゴリズムの所要時間の実測値と予測値の比較

図中で Measured が実測値、Predicted が予測値である。また、この予測値を基にアルゴリズムを選択候補に残すか否かを判断した結果を Candidate に示す。

全体的な傾向として、予測値が実測値の半分以下となっており、まだ十分な精度が得られていないことが分かる。原因としてまず考えられるのは、基本通信性能モデルとして、単純な線形モデルを用いていることが挙げられる。実際の通信に要する時間は、特にメッセージサイズが小規模～中規模の範囲では線形と大きく異なる傾向を示す。また、ノード内での通信処理に伴うソフトウェアオーバヘッドの影響、さらに、アルゴリズムによってはメモリコピーの影響も、加味する必要があると考えられる。

なお、アルゴリズムを絞り込むうえで重要となる、アルゴリズム間の相対的な性能の優劣については、実測値と予測値でほぼ同じ傾向を示した。その結果、非常に遅い 2 つのアルゴリズムを候補から除外することが出来た。

4． まとめ

本年度は、昨年度までに提案していた集団通信アルゴリズム選択技術におけるアルゴリズムの性能予測モデルについて精度を検証した。その結果、絶対的な精度自体には問題があるものの、アルゴリズムの候補を絞り込むための相対的な精度としては、有効であることが分かった。

5． 今後の計画・展望

現在の予測モデルは非常に簡単なものであり、改善の余地がある。メッセージサイズによる詳細な通信時間の見積もりや、ソフトウェアオーバヘッド、メモリコピー等の影響を加味することによる精度の向上を図りたい。

6． 利用がなかった場合の理由

本年度は、主に昨年度までに計測していた実験結果をもとに精度の検証を行うことが出来たため、結果的に計算機をほとんど利用しなかった。来年度、継続利用が認められれば、今年度の検証により判明した精度の問題を解決した新しいモデルでの実験を行いたい。

平成 24 年度　RICC 利用研究成果リスト

【国際会議、学会などでの口頭発表】

"Efficient Runtime Algorithm Selection of Collective Communication with Topology-Based Performance Models," Takeshi Nanri and Motoyoshi Kurokawa, The 2012 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'12), Jul 2012.

【その他】