社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

λコンピューティング環境構築のための共有メモリシステムの実装と評価

谷口　英二†　　馬場　健一††　　村田　正幸†

† 大阪大学 大学院情報科学研究科　〒 565–0871 大阪府吹田市山田丘 1–5
†† 大阪大学 サイバーメディアセンター　〒 567–0047 大阪府茨木市美穂ヶ丘 5–1
E-mail: †{e-tanigu,murata}@ist.osaka-u.ac.jp, ††baba@cmc.osaka-u.ac.jp

あらまし　ネットワークにおける高速かつ大容量な伝送を可能とする技術への要求を満たすために、光伝送技術を用いた研究が活発に進められているが、パケット交換技術に基づいたアーキテクチャをとる限り、個々のコネクションに対する高品質通信の実現は非常に難しくなっている。そこで、各ノード計算機を接続している光ファイバを、インターネットとして利用するのではなく専用の通信路として利用し、WDM 技術を用いた高速な通信チャネルとして活用するλコンピューティング環境を提案する。すなわち、各ノード計算機と接続しているネットワークを仮想的なリングネットワークとして利用し、このリング上にデータを載せる、あるいは伝送することにより、高速・高品質通信を可能とする技術である。本稿では分散計算を行う場合に、これらの技術のうちの一つである、各ノード計算機上に存在する共有メモリを高速にアクセスする手法を実装し、その性能を明らかにする。具体的には、日本電信電話株式会社フォトニクス研究所が開発している情報共有ネットワークシステム (AWG–STAR システム) を用いる。その結果、AWG–STAR システムによる分散計算は、共有メモリへの書き込み回数に大きく依存し、現状ではボトルネックとなっていることがわかった。そこで、効率よく共有メモリへの書き込みを行うことで AWG–STAR システムの性能を向上させることが可能であることを示した。

キーワード　λコンピューティング環境，共有メモリ，分散計算，AWG–STAR システム

Implementation and Evaluation of Shared Memory System for Establishing λ Computing Environment

Eiji TANIGUCHI†, Ken-ichi BABA††, and Masayuki MURATA†

† Graduate School of Information Science and Technology, Osaka University
1–5 Yamadaoka, Suita, Osaka 565–0871, Japan
†† Cybermedia Center, Osaka University　5–1 Mihogaoka, Ibaraki, Suita, Osaka 567–0047 Japan
E-mail: †{e-tanigu,murata}@ist.osaka-u.ac.jp, ††baba@cmc.osaka-u.ac.jp

Abstract　Optical transmission technology is studied actively in order to realize high-speed transmission and broadband networks. However, conventional packet-based switching technology cannot realize the true high quality communication for each connection. Then we propose λ computing environment which has the high-speed channels utilizing optical fibers connecting computing nodes. That is, this technology makes high-speed and high-quality transmission possible by utilizing networks, connected to each computing node, as the virtual ring network. In this paper, we implement high speed access method to shared memory which is on each computing node, and show its performance when we use it for distributed computing.We use AWG–STAR system, which is developed by NTT Photonics Laboratory, as an instance. As a result, we can show that the performance of AWG–STAR system is highly depend on the number of write access times, and it is bottleneck. So, by decreasing the number of write access times, we can improve the performance of AWG–STAR system.

Key words　λ computing environment,, shared memory, distributed computing, AWG–STAR system

1. はじめに

近年、画像処理や遺伝子解析、地球環境のシミュレートなど 1 台の計算機では実用的な時間内で解を算出できないような問題や 1 台の計算機では保持できない膨大なデータを扱う問題を計算する要求が生じている。このような計算を実現する方法として、多数の計算機を高速なネットワークで接続して計算機間で協調動作を行いながら計算する PC クラスタや、インターネット上の多数存在する遊休 PC を利用するグリッドコンピューティングと呼ばれる技術がある。グリッドコンピューティングでは、現在のインターネットの TCP/IP 上の Globus や MPI (Message Passing Interface) を用いてデータ交換を行いながら計算を行う。しかしながら、TCP/IP のようなパケット単位のデータ交換ではパケット処理に要するオーバヘッドが大きく、大規模計算で行われる大量のデータ共有やデータ交換を行うには十分な性能を得ることは非常に難しい。さらにこのような技術では高速かつ高品質な通信が要求される。

高速ネットワークを実現する手段として光の波長を用いて多重化を行う WDM (Wavelength Division Multiplexing) 技術が研究、開発されている。また WDM を利用してインターネットの高速化を実現する IP over WDM ネットワークの研究が行われている。しかしながら、現在のネットワークにおいてはルーティングを行う際に、光信号を電気信号に変換し、もう一度光信号に変換する処理を行っており、光の高速性を有効に活用できていない。そのため、WDM 技術以外のさまざまなフォトニック技術を下位のレイヤの通信技術とする GMPLS (Geralized Multi-Protocol Label Switching) と呼ばれるインターネットのルーティング技術や、フォトニックネットワークの真の IP 化を実現するためのフォトニック技術に基くフォトニックパケットスイッチの研究がさかんに研究されている [1], [2]。しかしながら、これらの技術はパケットを情報を扱う粒度として用いており、いかにして高速に転送するかに焦点をおくベストエフォート型通信であるため、高品質性を達成するのは困難である。

そこで各計算機を接続している光ファイバを、インターネットとして利用するのではなく専用の通信路として利用し、高速な通信チャネルとして活用する λ コンピューティング環境を提案している [3]。すなわち、各計算機と接続しているネットワークを仮想的なリングネットワークとして利用し、このリング上にデータを載せる、あるいは伝送することにより、通信を意識することなくデータ共有あるいはデータ交換を可能とし、高速性、高品質性を両立する技術である。

本稿では、分散計算を行う場合に、これらの技術のうちの一つである、各ノード計算機上に存在する共有メモリを高速にアクセスする手法を実装し、その性能を明らかにする。具体的には、日本電信電話株式会社フォトニクス研究所が開発している情報共有ネットワークシステム (AWG–STAR システム) [4], [5] を用いる。このシステムでは、各ノード計算機が波長可変光源を通じて光ファイバにより AWG (Arrayed Waveguide Grating) と呼ばれるルータに接続され、物理的にはスタートポロジを、論理的にはリングトポロジを形成している。また、各ノード計算機は共有メモリボードを搭載しており、共有メモリボード上のメモリは、AWG–STAR システム上でリングネットワークを構成している全ノード計算機で同一のデータを保持している。すなわち、このシステムは AWG ルータと波長可変光源をベースとした動的な波長ルーティングを使用し、複数端末ノード計算機の共有メモリを共有する、多対多マルチキャストシステムである。

以上より、本稿では、AWG–STAR システムを用いた λ コンピューティング環境を構築し、実際の分散計算アプリケーションを動作させることにより共有メモリのアクセス手法の性能を明らかにする。分散計算アプリケーションには、並列計算のベンチマークプログラムを用い、基本性能を調べたのち、さらなる改善手法を検討し、その効果をはかる。

以下、2 章では λ コンピューティング環境とそのシステムについて述べる。3 章では本稿で用いた AWG–STAR システムについて説明する。4 章で AWG–STAR システムを用いて分散計算を行った場合の共有メモリシステムの性能を評価する。最後に 5 章で本報告についてのまとめと今後の課題について述べる。

2. λ コンピューティング環境における分散計算システム

本章では、提案している新たな分散計算環境である λ コンピューティング環境における分散計算システムついて述べる。λ コンピューティング環境では、分散計算を行う複数の計算機を接続するネットワークを、パケット交換に基づく既存のインターネットではなく、専用の通信路として利用し、WDM 技術を用いた高速な通信チャネルとして活用する。すなわち、λ コンピューティング環境では、各ノード計算機を接続するネットワークを仮想的な光リングネットワークとして利用し、このリングネットワーク上にデータを載せるあるいは高速に伝送することにより、通信を意識することなくデータ共有あるいはデータ交換を可能とする技術である。

λ コンピューティング環境においては、次の二つのアーキテクチャを対象としている。ひとつは、λ コンピューティング環境における仮想光リングネットワークを、共有メモリとして利用し、各ノード計算機内のローカルメモリをキャッシュなどに利用するアーキテクチャ（共有メモリ型アーキテクチャ）[3]、もうひとつは、仮想光リングネットワークを、高速な伝送路として利用し、共有すべきデータは各ノード計算機のメモリ内におくアーキテクチャ（高速チャネル型アーキテクチャ）である。以下では、高速チャネル型アーキテクチャとメモリアクセス手法について述べる。

高速チャネル型アーキテクチャは各ノード計算機にそれぞれ共有メモリ領域を用意し、すべてのノード計算機が同じデータを持つ手法である（図 1）。このアーキテクチャでは、データ更新の際に高速チャネルを利用して、全ノード計算機に対して更新されたデータの配送を行う。従って、共有メモリへ書き込む場合は、高速チャネルへのアクセスが生じるが、読み出し時には各ノード計算機の共有メモリからデータを取得すればよく、

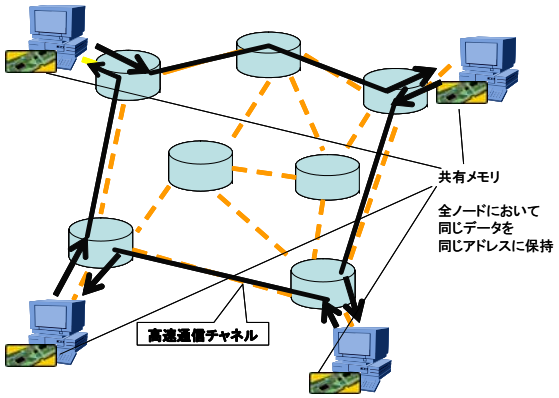

図 1　高速チャネル型アーキテクチャの構成

高速チャネルへのアクセスは発生しない。本稿において、実験に使用している情報共有ネットワークシステムは、このタイプのアーキテクチャを実現するひとつの例である。

3. AWG–STAR システムを用いた共有メモリシステムおよびメモリアクセス手法の実現

3.1 AWG–STAR システムの概要

AWG–STAR システムは、日本電信電話株式会社フォトニクス研究所により開発されたシステムであり、WDM 技術によるデータ転送と AWG ルータによる波長ルーティング技術によって実現された情報共有ネットワークシステムである。AWG ルータは波長に基づいたルーティングを行っており、電気信号に変換せず光信号をそのまま処理するため、高速なネットワークを構築することができる。また、各ノード計算機は、すべてのノード計算機で同一のデータを保持する共有メモリを持ち、AWG ルータおよび WDM を利用して構成された高速な光リングネットワークを利用することによりノード計算機間で共有メモリ上のデータ交換をリアルタイムに行うことができる。従来ならば、データを共有するためには何らかの明示的なデータ転送が必要であったが、AWG–STAR システムでは共有メモリに書き込まれたデータを光リングネットワークを用いて伝送するため、データ転送を意識することなく、高速に全ノード計算機の共有メモリの更新が行われる。

そこで、本稿では、λ コンピューティング環境を実現するひとつの方法として AWG–STAR システムを用いる。すなわち、各ノード計算機においてはそれぞれが独自に計算を行い、分散計算に必要なデータは AWG–STAR システムにより共有される。しかしながら、全ノード計算機間でデータを共有するためには、データ更新時にデータが光リングネットワークを 1 周回する必要があり、共有メモリへのアクセス手法によっては、この遅延時間が分散計算の性能に影響を与えることが考えられるため、それらの評価を行う。

3.2 AWG–STAR システムを用いた共有メモリシステムの構成

AWG–STAR システムの構成図を図 2 に示す。このシステムでは、各ノード計算機は波長可変光源を通じて光ファイバにより AWG ルータに接続され、光リングネットワークを形成している。AWG–STAR システム上の全ノード計算機は、共有メモリボードを搭載し共有メモリはこのボード上にある。以降、特に断らない限り共有メモリは共有メモリボード上のメモリを指す。表 1 に共有メモリボードの仕様を示す。

表 1　共有メモリボードの仕様

光インタフェースの伝送速度	2.152 Gbps
ノード計算機の 1 回当たりの転送データ量	1 KByte
ノード計算機でのフレーム転送処理遅延	500 ns
共有メモリへの書き込み	最大 約 67 MBytes/s
共有メモリから読み出し	最大 約 60 MBytes/s

3.3 共有メモリへのアクセス

共有メモリのある共有メモリボードは、計算機と PCI バスで接続されている。すなわち、共有メモリへの読み出しおよび書き込みは PCI バスを経由して行われるため、ローカルメモリへアクセスする場合よりも遅延時間が大きくなる。

光リングネットワークを構成している全ノード計算機は同一のデータを共有メモリに保持している。あるノード計算機が共有メモリのデータ更新を行うと、この時更新されたデータが光リングネットワークを周回し、光リングネットワークに接続された全てのノード計算機上の共有メモリの同一アドレスのデータ更新を行う。共有メモリからのデータ取得については、自ノード計算機の共有メモリから読み出すため、光リングネットワークの通信路に負担をかけない。

3.3.1 共有メモリアクセス手法

共有メモリへのアクセス手法は二通りある。ひとつは共有メモリボードの機能を用いた DMA (Direct Memory Access) アクセスであり、もうひとつはポインタを用いたアクセスである。これらは、共有メモリの先頭からのオフセットもしくは直接アドレスを指定することで共有メモリへのアクセスが可能である。

また共有メモリの更新には二つの場合がある。ひとつは自ノード計算機の共有メモリに書き込む場合であり、もうひとつは他ノード計算機からの共有メモリの更新情報を受信した場合である。光リングネットワーク上では、常にひとつのトークンが流れており、各ノード計算機はそのトークン上に更新を行ったデータに関する送信フレーム（アドレス、データ、制御コード、CRC）を付加し、次のノード計算機に転送する。

（1） 自ノード計算機の共有メモリに書き込む場合

この場合、まず自ノード計算機の共有メモリに書き込み、その後トークンが回ってきた際に、送信フレームをトークンに附随している送信フレームの最後尾に付加し、次のノード計算機にトークンを送出する。この時に、送信フレームに付加できるデータのサイズは 1KB である。リングを 1 周し、トークンが再度回ってきたら先ほど付加した送信フレームを削除する。ただし、送信中にエラーが発生すれば、リトライが行われる。

（2） 他ノード計算機からの更新情報を受信した場合

トークンが回って来ればトークンに付加されている他ノード計算機の送信フレームを確認する。他ノード計算機の更新情報がトークンに附随していれば、データを読み込み自ノード計算機の共有メモリを更新し、次のノード計算機に向けてトークンを

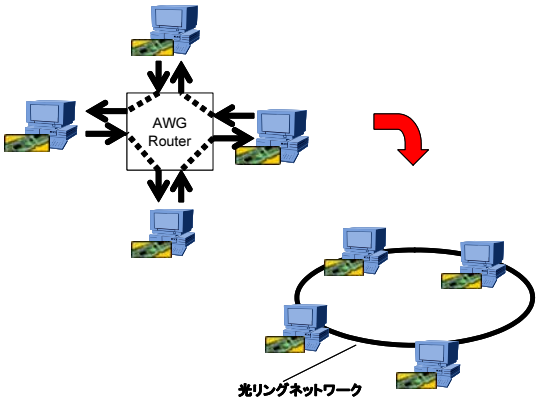

図 2　AWG–STAR システムの物理構成と論理構成

送出する。

3.3.2　共有メモリアクセスにおける遅延時間

各ノード計算機の共有メモリを利用するには、ローカルメモリにアクセスする以上に遅延時間を要する。その要因は、PCI バスを経由する遅延時間とデータ共有を行うための遅延時間である。例えば、更新されたデータを全ノード計算機が共有するためには、少なくともデータが光リングネットワークを 1 周は周回しなれけばならない。

データ共有のための遅延の要因は二つある。ひとつは共有メモリボードにおける転送処理遅延であり、もうひとつは光ファイバによる伝搬遅延である。ノード計算機において前のノード計算機から転送されてきたフレームを、次のノード計算機に転送するために処理時間を必要とする。具体的には送信フレームの削除と追加、共有メモリへの反映のために 500ns の時間を必要とする。また光ファイバによる伝搬遅延時間は 5ns/m となっている。

以上のことから AWG–STAR システムを用いた共有メモリシステム上で分散計算を行う場合、データ共有のための通信時間、共有メモリへの書き込み回数が性能を左右すると考えられる。次章では実際にアプリケーションを動作させ、共有メモリへのアクセス手法について検証する。

4.　実験と評価

本章では、並列アプリケーション集である SPLASH2 [6] の中のいくつかのプログラムを動作させ、実行時間を測定し、AWG–STAR システムを用いて構成した共有メモリシステムおよびメモリアクセス手法の性能を評価する。また、従来の TCP/IP による MPI を用いた場合の実行結果も併せて示す。

4.1　実験システム環境

今回の実験では、ノード計算機数に応じて光リングの長さを変えている。具体的にはノード計算機数を P とすると、光リングネットワークの長さは $20P$ m としている。MPI を用いた方式による実験でも、使用した計算機は AWG–STAR システムによる実験と同じ計算機を使用した。その際、計算機は 100Mbps の Ethernet で接続され、ひとつのスイッチングハブに全て接続されている。表 2 に実験に用いた計算機の仕様を示す。

表 2　実験に用いた計算機の仕様

CPU	Xeon 3.06 GHz
メインメモリ	SDRAM 1 GB
1 次キャッシュ	512 KB
2 次キャッシュ	512 KB
NIC	Intel PRO/1000MT
PCI バス	64 bit / 66 MHz
PCI 転送速度	533 MBytes/sec
OS	Redhat Linux 9
コンパイラ	gcc 3.2
MPI ライブラリ	MPICH 1.2.5

4.2　評価に用いるアプリケーション

SPLASH2 は、スタンフォード大学で開発された分散計算用のベンチマークアプリケーションである。プログラム中には分散計算を行うために必要となるバリア同期関数などは実装されていないため、AWG–STAR システムの機能を使用した関数を作成した。本稿では、SPLASH2 のアプリケーションの中から基数ソートプログラム（以下 RADIX）、LU 分解プログラム（以下 LU）、高速フーリエ変換プログラム（以下 FFT）を実験に使用した。

4.3　共有メモリシステムの性能評価

4.3.1　基数ソートプログラムによる実行結果

図 3 に RADIX の場合の実行時間を示す。MPI を用いた場合の結果も併せて示す。

ノード数が 2 の時に、AWG–STAR システムが MPI を用いた場合よりもよい性能を示している。その理由は、MPI を用いた場合では計算過程における通信量が AWG–STAR システムを用いた場合よりも大きいためである。MPI を用いた場合の計算の過程で生じる、1 回当たりのデータ共有のための通信量は以下の通りである。ブロードキャストされるデータ量はパラメータとして与えられる基数 r で決定され、全ノードがブロードキャストを行うため、ノード数を P とすると、データ共有 1 回あたりの総通信量は全ノード計算機において $O(rP)$ となる。またこの場合、他ノード計算機からのブロードキャストを待つため並列化が行うことができない。一方で、AWG–STAR システムの場合は、各ノード計算機が独立して共有メモリに書き込めばデータの共有が実現できるため、計算量は全ノード計算機で $O(r)$ となる。

さらに、計算結果の集約においても、MPI を用いた方式が AWG–STAR システムを用いた方式よりも計算量を必要とする。MPI では結果の集約には各ノード計算機からの通信が必要となり、これは先と同様に並列化が行えない。従って、結果の集約にはソート対象の要素数を n とすると $O(n)$ の計算時間を必要とする。一方で、AWG–STAR システムでは結果の集約は各ノード計算機がそれぞれ独立に行えるため、結果の集約に要する計算時間は $O(\frac{n}{P})$ となる。すなわち、MPI 用いた時の通信量が AWG–STAR システムを用いた時の通信量より大きいため、AWG–STAR システムがよい性能を示したと考えられる。

以上のことから、データ共有のための通信量が AWG–STAR システムの共有メモリシステムの性能を左右すると考えられ

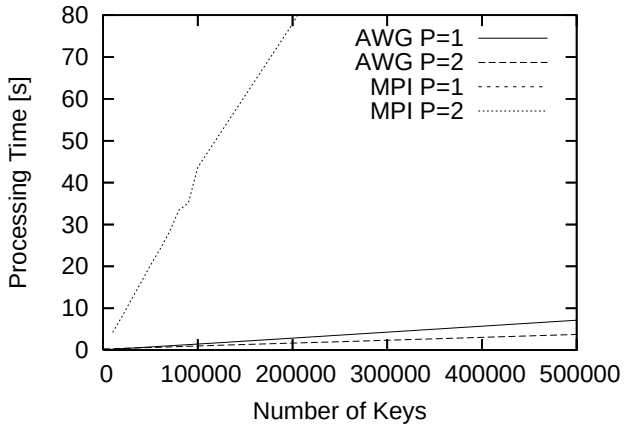

図 3　基数ソートの実行時間

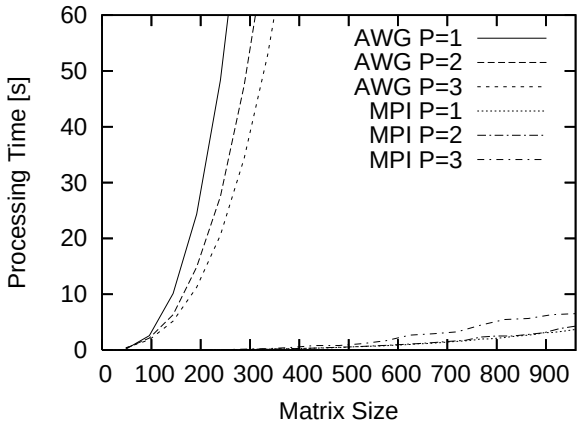

図 4　LU 分解の実行時間

る。AWG–STAR システムにおいてはデータを共有メモリに書き込むことで通信が発生するため、共有メモリへの書き込み回数が AWG–STAR システムの共有メモリシステムの性能として表れる。

4.3.2　LU 分解プログラムによる実行結果

図 4 に LU の場合の実行時間を示す。LU の実行時間は、AWG–STAR システムを用いた場合はノード計算機数の増加に伴い実行時間が減少しているが、MPI を用いた場合は、ノード計算機数の増加に伴い実行時間が増えている。これは MPI のデータ共有のための通信が並列化できないこと、およびノード計算機数が増加することによりデータ共有のための通信回数が増えるためである。

AWG–STAR システムを用いた場合が MPI を用いた場合に比べて性能が十分でない。その理由は次のように考えられる。AWG–STAR システムにおいて LU を用いた場合、共有メモリへの書き込みアクセスが多く行われる。ノード計算機数が 3 の時に行列サイズ 480 の場合を考えると、共有メモリへの書き込み回数が 1 ノード計算機あたり約 1166 万回あるが、そのうちの 1106 万回が共有する必要のないデータの書き込みである。これは書き込み回数全体の約 95%にあたる。従って、この全書き込み回数の 9 割にも及ぶ不必要な書き込みによって生じる通信の遅延のため、AWG–STAR システムの性能が十分に生かせずに性能の低下を招いたといえる。プログラムのチューニングにより共有メモリへのアクセス回数を減らすことで AWG–STAR システムの性能の向上が可能であると考えられる。

4.3.3　高速フーリエ変換プログラムによる実行結果

図 5 に FFT の場合の実行時間を示す。FFT を用いた場合の実行結果も LU を用いた時と同様に、AWG–STAR システムを用いた場合が MPI を用いた場合に比べて性能がよくない。これも LU の時と同様に、実行中に共有メモリに 1 要素づつ書き込む処理が多いため、通信量が多くなってしまい、それによる遅延のためである。

4.4　共有メモリアクセス手法の高速化

前節において、LU および FFT において AWG–STAR システムによる共有メモリシステムでは十分な性能が得られないことがわかった。その主な要因が共有メモリへの書き込みアクセス

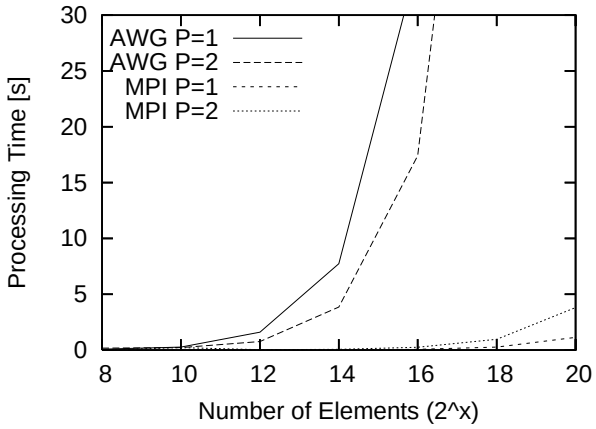

図 5　FFT の実行時間

の多さによる遅延である。すなわち、ある処理において共有メモリへの書き込み回数が増大するため、光リングネットワークの周回数が増大し実行時間の増加につながっている。そこで本節では、共有メモリアクセス手法の性能向上のための手法について考察する。

4.4.1　プログラムの改善による性能向上手法

AWG–STAR システムを用いた共有メモリシステムの性能が十分でない要因として、共有メモリへの集中的な書き込みアクセスがある。このような共有メモリへの書き込みが頻繁に発生するとデータの周回に伴う遅延が発生する。特に小さなデータを逐次書き込む場合、その度にリングネットワークへのアクセス遅延が生じる。この遅延を減らすには、共有メモリにデータをまとめて書き込むことが考えられる。まとまったデータを一度に書き込むことで周回の回数が減り遅延の減少、ひいてはプログラムの実行時間の減少につながる。

まず、LU の場合を考える。LU においては各ノード計算機はブロックを割り当てられ、ブロック単位で処理を行う。4.3.2 節における実験では、ブロック内のデータを共有メモリから読み出し演算を行い共有メモリに書き込むようになっているため、この操作がボトルネックとなっている。このボトルネックを解消するための実装の改変として、ブロックを共有メモリからローカルメモリにコピーし、必要な演算はコピーしたローカ

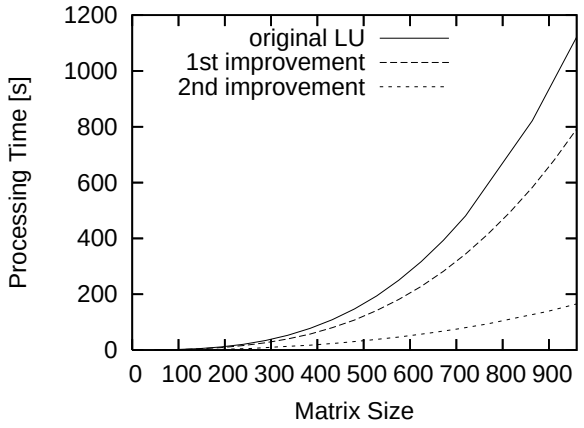

図 6　チューニング後の LU 分解の実行時間（ノード計算機数 3）

ルメモリ上のデータを用いて演算を行い、ブロックの処理が終了した段階で、共有メモリに書き込む。プログラムの改変前は、実際に行う処理に依存するが、ブロックサイズを B とすると 1 ブロック当たりの処理において最大で B^3 回書き込みを行っていたが、ブロック単位で共有メモリに書き込むように改変することで、1 ブロックのデータの共有メモリへの書き込みを、1 回にまで減らすことができる。

LU 分解においてはさらにチューニングを進めることが可能である。LU 分解では、共有メモリからデータを読みだし、計算を行い、共有メモリにデータを書き出す処理を繰り返し行うことで計算を行う。しかしながら、LU 分解では共有メモリに書き出されたデータを再び読み出して計算を行うことが頻繁にある。そのため、処理を再度行われるデータに関しては、一度共有メモリからデータを読み出した後はローカルメモリ上に保持しておき、以降のそのデータの処理に関してはローカルメモリ上のデータを使用し、処理が施されなくなった時点で共有メモリ上に書き出すようにすれば、共有メモリへのアクセス回数が削減され実行時間の短縮が行える。

4.4.2　プログラムの改善を行った場合の性能評価

図 6 にブロック単位での共有メモリへのアクセスを行うようにチューニング行った時の実行時間 (1st improvement)、それに加えてブロック単位でのアクセスに加えて処理が行われなくなった時点で共有メモリに書き出すようにチューニングを行った時の実行時間 (2nd improvement) を示す。共有メモリへの書き込みをブロック単位で行うように改善した結果、ノード計算機数が 3 で行列サイズが 480 の場合、書き込み回数は 1160 万回から 3000 回に減少し、その結果、実行時間を改善前の約 70%にまで減少することに成功した。さらにブロック単位での共有メモリへの書き込み、および処理が行われなくなった時点での共有メモリへの書きだしの 2 つを併用するようにプログラムを改善した結果、ノード計算機数が 3 で行列サイズが 480 の場合、書き込み回数は 300 回にまで減少し、実行時間は改善前の 15%にまで短縮ができた。

また、FFT についても同様に一定量のデータををまとめて共有メモリに書き込むことで実行時間を最大で 20%まで短縮が可能となった。

5. おわりに

本稿では、λ コンピューティング環境として、AWG–STAR システムを利用した場合の共有メモリシステムの性能の評価を行った。AWG–STAR システムを用いて光リングを構成し高速な通信チャネルとして利用し、各ノードの共有メモリを分散計算におけるデータ共有手段として用いて分散計算のベンチマークアプリケーションを実際に実行することでその評価を行った。その結果、AWG–STAR システムのようなモデルの共有メモリシステムを λ コンピューティング環境として利用する場合、共有メモリへの書き込みアクセス回数が性能に影響を与えることがわかった。一方、共有メモリへのアクセス時間の短縮、光リングネットワークへの転送処理時間の短縮ができれば、性能の改善が図られるため、ハードウェアの改善についても検討する必要がある。

今後は、今回は小さなデータを交換するベンチマークアプリケーションを用いて評価を行ったが、実用的な分散計算を行うアプリケーション、ならびに大きなデータを取り扱うアプリケーションを用いた場合の性能評価も行っていく予定である。

謝　辞

本研究を進めるに当たり、日本電信電話株式会社フォトニクス研究所 界義久氏、岡田顕氏に多大なご支援を頂いた。深く謝意を示す。

文　献

[1] K. Baba, R. Takemori, M. Murata and K. Kitayama: “A packet scheduling algorithm for the 2x2 photonic packet switch with fdl buffers”, in Proceedings of ECOC2002 (2002).

[2] T. Yamaguchi, K. Baba, M. Murata and K. Kitayama: “Scheduling algorithm with consideration to void space reduction in photonic packet switch”, IEICE Transactions on Communications, **E86-B**, 8, pp. 2310–2318 (2003).

[3] 中本博久，馬場健一，村田正幸：“λ コンピューティング環境における共有メモリアクセス手法の提案”, 電子情報通信学会技術研究報告 (CS2004-24), **第 104 巻**, 81 号, pp. 43–48 (2004).

[4] 岡田顕，田野辺博正，松岡茂登：“波長ルーティング技術を用いたダイナミックに再構成可能な情報共有ネットワーク”, 電子情報通信学会技術研究報告 (IN2003-332), **第 103 巻**, 692 号, pp. 423–427 (2004).

[5] A. Okada, H. Tanobe and M. Matsuoka: “Dynamically reconfigurable real-time information-sharing network system based on a cyclic-frequency AWG and tunable-wavelength lasers”, in Proceedings of ECOC2003 (2003).

[6] S. Cameron, M. Ohara, E. Torrie, J. P. Singh and A. Gupta: “The SPLASH–2 Programs: Characterization and Methodological Considerations”, in Proceedings of the 22nd Annual International Symposium on Computer Architecture, pp. 24–36 (1995).

[7] 天野英晴：“並列コンピュータ”, 昭晃堂 (1996).

[8] P. パチェコ 著 秋葉 博 訳：“MPI 並列プログラミング”, 培風館 (2001).