審 査 の 結 果 の 要 旨

氏　名　　斎 藤　秀 雄

本論文は、「Design and Implementation of Scalable High-performance Communication Libraries for Wide-area Computing Environments」（広域計算環境における並列計算用のスケーラブルな高性能通信ライブラリの設計と実装）と題し、非均質な広域計算環境における並列計算をこれまでにない規模で効率良く実行できるようにすることを目的とし、高性能な通信を、環境に適応して自動的に達成する通信ライブラリの設計と実装を提案し、実証実験によりその有効性を論じたものであり、全体で 7 章から構成されている。論文は英語で書かれている。

第 1 章は「Introduction」（緒言）であり、Wide Area Network（WAN）のバンド幅の増加に伴って広域計算環境において並列計算を行う機会が増加したことについて述べ、広域計算環境における並列計算を支援する通信ライブラリが満たすべき接続性・スケーラビリティ・性能等に関する要件について説明している。そして、本研究で提案している通信ライブラリの構成技術である接続管理手法とランク割当て手法について述べている。また、それらを用いて実装したメッセージパッシングライブラリ Multi-Cluster MPI とソケットライブラリ Scalable Sockets について述べている。

第 2 章は「Related Work」（関連研究）と題し、本研究に関連する研究について述べている。まず、Virtual Private Network（VPN）や SOCKS などについて述べ、それらが小規模な環境で接続性を確保するためには有用であるが、大規模な環境で任意のノードが互いに通信できるようにするためには多くの設定が必要であることを指摘している。次に、スケーラビリティを向上させるための手法として Peer-to-Peer（P2P）オーバレイネットワークについて述べ、性能上の理由で並列計算には向かないことを指摘している。また、第 6 章の実験で Scalable Sockets の比較相手として用いている広域計算環境用の通信ライブラリ SmartSockets について述べている。

第 3 章は「Message Passing」（メッセージパッシング）と題し、メッセージパッシングモデル及びメッセージパッシングライブラリの構成法について説明している。まず、メッセージパッシングモデルの概要を述べ、メッセージパッシングのデファクトスタンダード Application Programming Interface（API）である Message Passing Interface（MPI）について説明している。次に、広域計算環境においてメッセージパッシングを行う場合の局所性と接続性に関する課題について述べ、MPICH-G2 や GridMPI などの既存の広域計算環境用の MPI ライブラリがそれらの課題にどのように対処しているかについて説明している。そして、スケーラビリティと通信性能を向上させるために確立する接続の数を制限する必要性について説明し、既存の MPI ライブラリの用いている単純な接続管理は全対全通信を伴うようなアプリケーションではうまくいかないことを指摘している。最後に、MPI ライブラリによるプロセスへの ID 割当て（ランク割当て）が MPI アプリケーションの性能に大きな影響を与えることを説明し、既存のランク割当て手法はアプリケーションや計算環境に十分に適応できないということを指摘している。

第 4 章は「Design and Implementation of MC-MPI」（MC-MPI の設計と実装）と題し、提案する接続管理手法とランク割当て手法について説明し、それらの手法を用いて実装されている MPI ライブラリ

Multi-Cluster MPI（MC-MPI）について説明している。MC-MPI は、アプリケーションを短時間実行するプロファイリング実行によって遅延行列と通信行列を取得し、本実行でそれらの行列を用いて最適化された接続管理とランク割当てを行う。接続管理は、オーバレイネットワークを構築することによって接続性の問題を解決し、各プロセスが確立する接続の数を O(log n)に制限することによってスケーラビリティを向上させる。また、それら O(log n)個の接続を遅延行列と通信行列に基づいて選択することによって限られた数の接続で高い通信性能を出す。一方、ランク割当ては、遅延行列と通信行列を基に作成した二次割当て問題を解くことによって通信オーバヘッドの低い割当てを行う。二次割当て問題は NP 困難であるが、ヒューリスティクスを用いる既存のライブラリを用いることによって良い近似解を得る。

第 5 章は「Performance Evaluation of MC-MPI」（MC-MPI の性能評価）と題し、4 クラスタ 256 コアからなる実広域計算環境を用いて MC-MPI の性能を評価している。ベンチマークとしては、MPI ライブラリの性能評価に広く用いられている NAS Parallel Benchmarks（NPB）と MPI で記述されたモデル検査ツールである Distributed Verification Environment（DiVinE）を用いている。接続管理については、MC-MPI の性能と全対全で接続を確立した場合の性能・手動で中継プロセスを設定した場合の性能・ランダムに接続の数を制限した場合の性能を比較することによって、限られた数の接続で既存の手法に劣らない性能が出ることを示している。また、NPB の Integer Sort（IS）や DiVinE のように多数のプロセスが同時に通信するアプリケーションでは、接続の数を制限することによってコンジェスチョンが回避できて、多数の接続を用いるよりむしろ高い性能が出ることを示している。一方、ランク割当てについては、MC-MPI の性能とホスト名に基づく割当てを用いた場合の性能・ランダムな割当てを用いた場合の性能を比較することによって、自動的に行った割当てで既存の手法と同等もしくはそれ以上の性能が出ることを示している。

第 6 章は「Scalable Sockets」（スケーラブルソケッツ）と題し、Socket API を提供することによって MC-MPI をより汎用的にしたライブラリ Scalable Sockets（SSOCK）について説明している。SSOCK は、MC-MPI の接続管理手法を用いて中継デーモンのオーバレイネットワークを構築し、そのオーバレイネットワークを用いて connect や send などを実現する。13 クラスタ 338～1,264 コアからなる実広域計算環境で行った評価実験では、SSOCK が既存の広域計算環境用の通信ライブラリよりスケーラビリティが高いことを示している。また、1 対 1 通信性能は同等で、全対全通信性能は SSOCK の方が高いことを示している。

第 7 章「Conclusion and Future Work」（結論と今後の課題）では、本論文の主たる成果をまとめるとともに、今後の方向性について述べている。

以上を要するに、本論文は、非均質な広域計算環境における並列計算をこれまでにない規模で効率良く実行できるようにすることを目的とし、高性能な通信を、環境に適応して自動的に達成する通信ライブラリの設計と実装を提案し、実証実験によりその有効性を論じたものであり、電子情報学上貢献するところが少なくない。

よって本論文は博士（情報理工学）の学位請求論文として合格と認められる。