2003－OS－94　(9)
2003／8／4

10Gb Ethernet上の通信プロトコル作成支援技術

住元真司[†]　久門耕一[†]　石川　裕[††]

本論文では、10ギガビットクラスの高速ネットワーク上での通信プロトコル作成支援技術について述べる。信頼性の高いシステムを高速ネットワークを用いたクラスタシステム上で短期に実現するには、通信性能のボトルネックの把握と障害発生時のシステムの稼働検証を効率的に行う開発支援ツールが重要になる。本論文では、プロトコル開発支援ツールに必要な技術と検討中のシステムの概要について述べる。

Supporting Technologies to Develop High Performance Communication Protocols Using 10Gigabit Ethernet

Shinji Sumimoto,[†] Kouichi Kumon[†] and Yutaka Ishikawa [††]

This paper presents supporting technologies to develop high performance communication protocols using 10 Gigabit class network. It is very important to provide development support tools to show bottle-necks of communication performance and to verify whether the system runs well or not when system error occurs in order to develop highly reliable system using a cluster system with high speed network. This paper discusses about technologies to develop the development support tools and an overview of our prototype system.

1. はじめに

近年、PCやEthernetネットワークなどコモディティハードウェアの高性能化と低価格化はとどまるところを知らない。その圧倒的なコストパフォーマンスを武器にコモディティハードウェアを組み合わせて大規模システムを実現する事例が増えている。特に、PCと高速ネットワーク、そしてハードウェア性能を引き出すシステムソフトウェア (ex. SCore[1]) を用い高い計算性能を実現するPCクラスタはハイパフォーマンスコンピューティング (HPC) の分野で既に主要なプラットフォームとなっている。

PCのプロセッサの動作周波数は3GHzを越え、コモディティネットワークであるEthernetについても、100MB/s以上のデータ転送能力を持つGigabit Ethernetが普及し、1GB/s以上のデータ転送が可能な10Gigabit Ethernetについても製品化が始まっている。コモディティネットワークの性能の向上は目覚しく、システムバスの転送能力に迫ろうとしている。

我々は、コモディティネットワークの高性能化により、現在共有メモリ型の大型サーバで実行されているデータベースや業務アプリケーションなどのビジネスアプリケーションについてもPCとコモディティネットワークを用いたシステム上で実行可能になると考えており、そのためのアーキテクチャの研究開発を行っている。

我々が研究を進めている次世代高性能アーキテクチャにおいては、現在のサーバマシンである共有メモリ型の並列コンピュータの特徴を備え、機器の障害や動的ハードウェアの増減に対処し高信頼性を提供する予定である。[2]

しかし、高性能、かつ、高信頼なシステムを短期に開発するには、 性能上のボトルネックの把握、ならびに、実際に障害を発生させての動作検証が容易にできる環境の実現が重要になる。このため、我々は性能上のボトルネックを容易に把握し、障害を模擬的に発生することができる開発ツールを開発して、次世代高性能アーキテクチャを実現することにしている。

本論文では、次世代高性能アーキテクチャを構成する基盤ソフトウェアモジュールである基盤通信機構を実現するための開発支援技術についてその概要を述べる。第2章で次世代高性能アーキテクチャと開発支援技術について述べ、第3章で次世代高性能アーキテクチャ上での通信機構の設計上の課題について述べる。第5、6章で開発支援ツールの設計と実現、第7章で関連研究について述べる。

† 富士通研究所
FUJITSU LABORATRIES
†† 東京大学
The Tokyo University

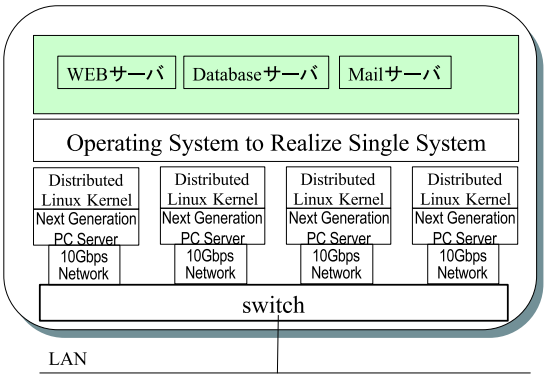

図 1 次世代高性能アーキテクチャ

2. 次世代高性能アーキテクチャと開発支援技術

コンピュータ技術の発展により、企業では、販売・生産・物流・財務・人事など企業活動全般にわたる業務を全社的に統合した企業情報システムが構築されている。このようなシステムは、企業活動の基幹に関わる業務であり、高信頼性が要求される。このため今まで、専用に開発された高価な並列コンピュータが使われてきた。

しかし、安価なプロセッサやディスクなどを次世代超高速通信ハードウエアで結合することにより、従来のハードウエアコストを 1/10 にした基幹業務向け並列コンピュータが実現できるようになる。このようなシステムを実現する次世代高性能アーキテクチャ [2] とそれを実現するシステムソフトウエア開発が急務となっている。

図 1 に次世代高性能アーキテクチャの構成図を示す。仮想マシンを用いた OS 核を 10Gbps クラスの超高速ネットワークを用いたボード間接続機器により結合して、全体として単一システムイメージを実現する。既存の共有メモリ型計算機と同様のシステム利用を想定しているため、遠隔プロセス実行、プロセスマイグレーションなど、アプリケーション実行に必要な大容量データが OS 核間で相互に転送される。

従来のソフトウエアシステムでは、ハードウエアは故障しないものと仮定してソフトウエアが開発されてきた。しかし、多数のプロセッサやディスクが接続される次世代高性能アーキテクチャにおいては、構成要素の一部であるプロセッサやディスクが動かなくなっても他の正常なハードウエアを使用して、単一並列コンピュータとして動作させることが可能となるソフトウエアシステムが必要となる。

このようなソフトウエアシステム開発においては、様々なハードウエア故障を系統的に模擬し、開発したソフトウエアが耐故障性を有する高信頼性を保っているかどうかを検証するための開発支援環境がとても重要になる。また、効率化という点では、開発だけでなく実運用時においての障害解析の効率化も重要になっており、開発と障害解析を支援する環境が重要になると我々は考えている。

このため、我々は次世代高性能アーキテクチャ実現のため、これを実現するための開発支援ツールを開発することにした。このツールに必要な開発支援技術として、サーバ障害の模擬技術 [3] と、ネットワーク障害の模擬技術の検討を進めている。

3. 次世代高性能アーキテクチャにおける通信機構設計

本章では次世代高性能アーキテクチャを構成する技術の中で高性能通信機構の設計について述べる。

3.1 設計方針

2 章で述べたように、次世代高性能アーキテクチャの開発については、単なるシステムの研究開発ではなく、開発支援技術を考慮したアーキテクチャを考えている。この方針を踏まえ、次世代高性能アーキテクチャで実現する通信機構の設計では、システムバス転送能力に匹敵する高い通信性能を実現すると共に、開発・デバッグ、試験、実運用時における通信プロトコルの高性能化、高信頼化を効率良く行うための開発支援ツールの実現を考える。

これまでの通信プロトコルの開発においては、設計、開発、試験、実運用時の障害解析を統合的に考慮した開発支援ツールは準備されなかった。我々は、開発支援装置と通信プロトコルとの連携によって、開発と障害調査が効率化できると考えている。

3.2 通信方式

第 2 章に記述したように、次世代高性能アーキテクチャにより実現されるシステムにおいては、大容量データが OS 核間で相互に転送される。遠隔プロセス実行やプロセスマイグレーションにおいてはアプリケーションプロセスのプロセスイメージを最短時間で転送する必要がある。このデータ転送は、主にある OS 核のメモリ上から別の OS 核のメモリ上へのデータコピーになる。10Gbps クラスのネットワークハードウェアの転送性能を最大限に引き出し、かつ、ホストプロセッサのプロセッサ利用率を最低限に抑える必要がある。

現在のサーバマシンである共有メモリ型の並列コンピュータが利用している通信方式は TCP/IP や unix-domain socket が主流であり、ホストプロセッサによるデータコピーを伴うため、10Gbps クラスのネットワークにはオーバヘッドが問題になる。

このため、データ転送時のホストプロセッサによるコピーを伴わない Remote Direct Memory Access による通信を採用して実現する。

4. 通信プロトコル作成支援ツールの目標と課題

本章では、第 3 章で述べた通信機構実現のための開発支援支援ツールについて、その目標、要件、課題を述べる。

4.1 通信プロトコル作成支援ツールの目標

次世代高性能アーキテクチャを効率的に開発するための通信プロトコル作成支援ツールでは以下に述べる機能を実現する。

(1) ネットワーク故障模擬の機能: 通信ソフトウェア、ならびに、通信ハードウェアと独立して故障を発生させることができる。

(2) 通信プロトコルの動作解析機能: 通信プロトコルの動作を容易に図示できる。

第 4.2、4.3 節に、それぞれの機能の要件を述べる。

4.2 ネットワーク障害模擬機能の要件

ネットワークの障害として想定されるものとしては、ネットワークインターフェイスカード (NIC) の故障、ネットワークスイッチの故障、ネットワークケーブルの故障がある。故障の原因には、ハードウェア障害の他、ネットワークスイッチや NIC のバグが想定される。

これらが故障すると、

- メッセージのデータ化け
- メッセージデータ長の不一致
- メッセージ廃棄
- メッセージ配送先の違い (ブロードキャスト)

といった現象が発生する。以上の現象は、不定期あるいは永続的に通信データや ack などの制御メッセージにおいて発生する。

ネットワーク障害機能は、上記のネットワーク故障を通信プロトコルの持つデータのタイプ毎に発生させる機能を持ち、機能挿入によるメッセージ処理能力への影響を最小限に抑える必要がある。

4.3 通信プロトコルの動作解析機能の要件

通信プロトコルの動作解析では、通信を行うアプリケーションがメッセージの送受信パターン、通信プロトコルの内部状態、物理ネットワーク上のメッセージ転送状態の把握が重要である。特に、通信性能上のボトルネックを把握するためには、以下の 3 つの状態を把握することが重要になる。

(1) 物理ネットワーク上のパケットギャップの把握:
物理ネットワークが idle 状態である割合

(2) データ再送の把握:
転送されているデータと冗長なデータの割合

(3) 制御パケットの把握:
データパケットと制御パケットの割合

従来はこれらを把握するために、個別に通信プロトコル処理のコードに状態ダンプ用のコードを個別に埋め込んで表示させ、状態を把握していた。このため、コードの埋め込みにより処理タイミングが変わって通信プロトコルの把握が困難になる場合があった。通信プロトコルを開発支援するツールとしては、定常的に通信プロトコルの状態把握が可能であると共に、状態を把握するための処理が通信プロトコルの動作に影響を与えてはならない。

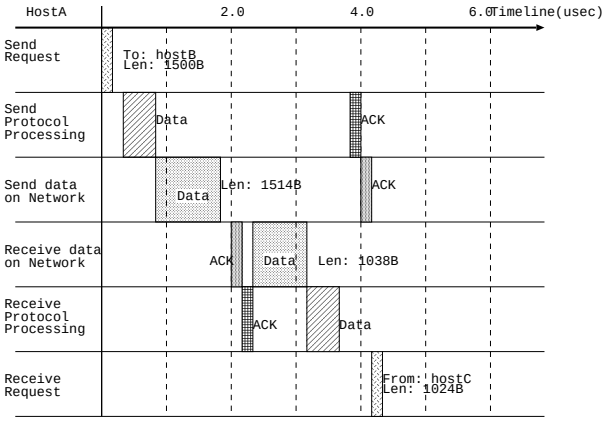

図 2 メッセージ通信状態図

さらに、通信プロトコルの状態が時系列に記録され、例えば、図 2 のようなメッセージ通信の状態がわかるように容易にグラフ化が可能である必要がある。図 2 は、あるホスト (HostA) において、1 つのメッセージを HostB に送信し、HostC から 1 つのメッセージを受信したときの、メッセージ通信の状態図の例を示している。図 2 では、アプリケーションからの送受信要求、プロトコル処理のタイプ、ネットワーク上の送受信データが経過時間と共に表されている。このような図をホスト毎に採取して、比較・合成することによりプロトコルの状態解析が可能になる。

以下に、通信プロトコルの動作解析機能の要件をまとめる。

(1) アプリケーションのメッセージ送受信パターンの把握: 送受信メッセージのサイズ、バッファアドレス、時間等の情報を一定時間リアルタイムに採取できること

(2) 通信プロトコルの内部状態の把握: データを格納したメッセージ、制御情報を格納したメッセージ、内部状態の情報を一定時間リアルタイムに採取できること

(3) 物理ネットワーク上のメッセージ転送状態の把握: 物理ネットワーク上の送受信メッセージを一定時間リアルタイムに採取できること
特に、10Gbps ネットワークでは 64 バイトメッセージの転送は 50ns しか時間がかからない [4]。このような短い時間内でログ採取ができることが重要になる。

以上のデータ採取が、あるイベントの発生後に、一

定時間ログの取りこぼしが無いことが要求される。

4.4 実現課題

以上の議論より、本通信プロトコル作成支援ツールの実現上の課題をまとめる。

(1) 低オーバヘッドのメッセージプロファイル方式:
10Gbps クラスネットワークにおいて通信プロトコル処理への影響を最小限に抑えた処理方式

(2) 低オーバヘッドのメッセージプロトコル状態収集方式:
通信プロトコル内部の状態採取をホストプロセッサ負荷を最小にして実現する方式

5. 通信プロトコル設計支援ツールの設計

本章では、第 4 章の議論を踏まえて、第 4.4 節にあげた、2 つの課題とネットワーク故障模擬の実現方式ついて議論する。

5.1 低オーバヘッドのメッセージプロファイル方式

10Gbps クラスのネットワークでは 64 バイトメッセージの転送に 50ns しか時間がかからない [4]。更に 10Gb Ethernet は全二重ネットワークであるため、送受信側両方で処理する必要がある。

この時間で、ホストプロセッサへの負荷を最小限に抑えたメッセージプロファイルは、現状では専用のハードウェア以外には実現困難である。このため、専用のハードウェアによる実現を考える。

5.2 低オーバヘッドのプロトコル状態収集方式

従来のプロトコル開発においては、ホストプロセッサによりプロトコルの状態を採取してホスト上のメモリにダンプする方式を利用していた。この方式はホスト上のメモリとプロセッサ資源を消費する他、複数のホスト上の状態を比較する場合には、データを一旦どこかのホスト上に集める必要がある。この方式をとった場合、実運用時の障害解析時に適用する場合には、ログ採取を常時行う、もしくは、外からのトリガーで採取できるようにする必要がある。

本研究では、開発支援ツールと通信プロトコルとの連携による新たなアプローチとして、あるホストの通信プロトコル状態を外部から採取可能な通信プロトコル (通信プロトコル状態採取プロトコル) により通信プロトコルの内部状態を採取する方式を検討している。

この通信プロトコルは、メッセージパケットヘッダのデータ領域を利用してプロトコル内部の情報をメッセージの送信時に埋め込む (piggy-back する) ことにより、常時プロトコルの状態を送出する。送出するデータの変更には、同様に外部からの特殊なパケットを用いる。

5.3 ネットワーク故障模擬方式

ネットワーク故障を模擬する方式として、ホストプロセッサによる方式と、NIC 上プロセッサによる方式、専用のプロセッサあるいはハードウェア処理による実現方式がある。

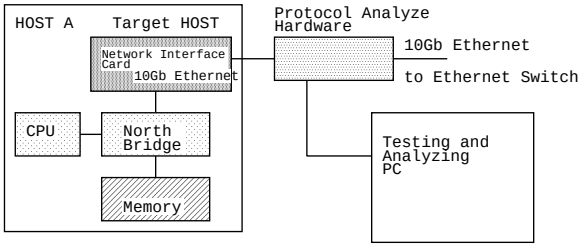

図 3 通信プロトコル作成支援ツールのシステム構成

本研究では RDMA プロトコルを採用しているため、NIC 上でプロトコル処理の実現を検討している。NIC 上でプロトコル処理を行う場合、ホストプロセッサによる方式では適用が難しく、NIC 上のプロセッサによる方式ではネットワーク故障模擬のための処理自体が通信プロトコルに影響を与える場合が想定されるため適切ではない。専用のプロセッサあるいはハードウェア処理による実現方式の採用が必要である。

6. 通信プロトコル作成支援ツールの実装

本章では、実現を予定している通信プロトコル作成支援ツール実装について述べる。

図 3 に通信プロトコル作成支援ツールのシステム構成図を示す。図 3 で、通信プロトコル作成支援ツールは、ターゲットマシン (Target HOST)、プロトコル解析･故障模擬ハードウェア (Protocol Analyze Hardware)、テスト、プロトコル解析用の計算機 (Testing and Analyzing PC) から構成されている。

ターゲットマシン： ターゲットマシンでは、プロセッサ、メモリ、10Gb Ethernet ネットワークインターフェイス他、サーバに 必要なコンポーネントが実装されている。

プロトコル解析・故障模擬ハードウェア： 10Gb Ethernet NIC とスイッチの間に挿入して、故障模擬、メッセージプロファイルを実現するハードウェアを持つ。

テスト・プロトコル解析用計算機： プロトコル解析ハードウェアとのインターフェイスを持ち、プロトコル解析ハードウェアからメッセージプロファイルデータをダウンロードして解析を行う。

図 4 にプロトコル解析･故障模擬ハードウェアの構成を示す。高速ネットワークポートを 2 ポート持ち、2 つの物理ネットワークと内部バス変換用のブリッジ (PHY/Internal bus Bridge) の間にネットワーク故障模擬用のハードウェア (Network Failure Generator) とメッセージ解析とログ用のハードウェア (Message Analyzer and Logger) を備える構成になっている。受信したメッセージは一旦内部バスを通って中央部のネットワーク故障模擬装置･メッセージ解析ログ装置

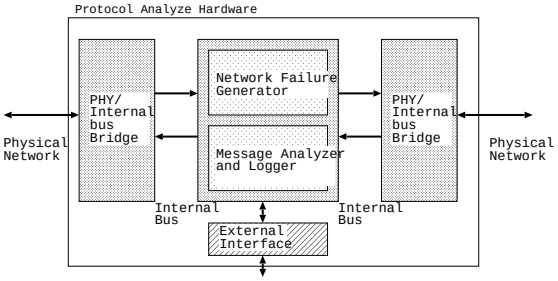

図 4 プロトコル解析・故障模擬ハードウェアの構成

を通って処理される。

実現する故障模擬機能については、第 4.2 節に述べた、 メッセージのデータ化け、 メッセージデータ長の不一致、 メッセージ廃棄、 メッセージ配送先の違い (ブロードキャスト) の実現について検討している。

また、通信プロトコルの動作解析機能については、第 4.3 節に述べた要件 の実現の詳細検討を進めている。 第 5.2 節で述べた 通信プロトコル状態採取プロトコルについても、設計を進めている。

6.1 開発支援ツールの利用イメージ

ネットワーク開発支援ツールの利用は、図 3 において、ターゲットホスト上に開発中のプロトコルを稼働させ、 テスト・プロトコル解析用計算機より、プロトコル解析・故障模擬ハードウェアに GUI もしくは、コマンドにより動作を指示することにより利用する。

故障模擬の利用

(1) テスト・プロトコル解析用計算機より与える障害を選択して、障害発生ボタンを押す。(あるいは障害発生コマンドを実行する。)
(2) プロトコル解析・故障模擬ハードウェアが設定された障害を発生させる。
(3) 障害の発生によりプロトコルの挙動を 通信プロトコル解析機能により解析する。

通信プロト コル解析機能の利用

(1) テスト・プロトコル解析用計算機より、採取するパケットプロファイラの条件を設定する。
  (a) パケットプロファイル開始、もしくは停止条件
  (b) 採取するデータの種類 (データの一部、制御情報、シーケンス番号他)
  (c) 時間の粒度と採取ステップ数
(2) プロトコル解析・故障模擬ハードウェアにおいてログが採取され、 テスト・プロトコル解析用計算機にデータが転送される。
(3) テスト・プロトコル解析用計算機上の表示ツールでプロトコルの挙動を調べる。

図 5 に 2 台のターゲットシステムを 1 つの テスト・プロトコル解析用計算機に接続した例を示す。2 つの結果を 1 つの図に時系列に表示して、比較することにより通信プロトコルの挙動を容易に把握することが可能になる。

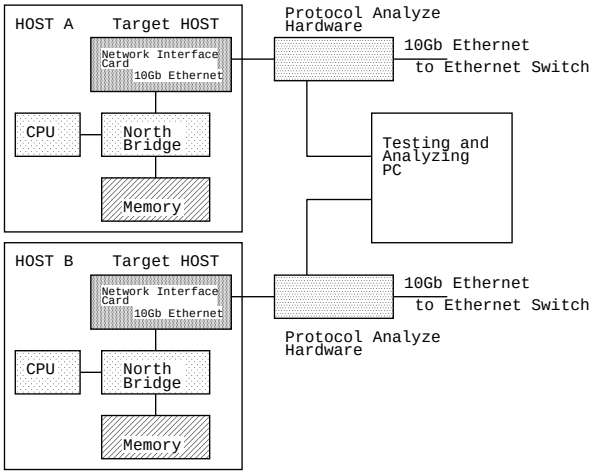

図 5 通信プロトコル解析機能の応用例

7. 関 連 研 究

7.1 通信向け開発支援ツールに関する関連研究

通信向け開発支援ツールについては、専用ハードウェアによるものと、ソフトウェアによるものとがある。

専用ハードウェアによるもの： ロジックアナライザ、パターン発生器など専用のハードウェアの組合せにより、故障模擬機能の実現、ならびに、メッセージの収集を行うことが可能である。しかし、1 台数千万と高価で装置が巨大な上、インタラクティブに独自プロトコルの障害模擬とプロトコル処理解析を行うのは適していない。我々の目指している開発支援ツールは、容易に持ち運び可能で、かつ、データ採取だけでなく、インタラクティブに独自プロトコルのデバッグと障害解析操作が可能なツールである。

ソフトウェアによるもの： パケットプロファイルは tcpdump など、多くのソフトウェアでの実装例があるが、10 ギガビットクラスのネットワークでの実現は困難である。

7.2 RDMA に関する関連研究

RDMA に関する関連研究としては、以下に述べる仕様がある。

RDDP プロトコル [5] IETF において iWarp から発展して charter になった。IP 上での RDMA プロトコルの標準化を進めている。

RDMA コンソーシアム [6] EMC, Cisco, NetApp などが推進する TCP/IP 上で RDMA を実現しようとして標準化団体、RDMA 自体では verb と

呼ばれる動作仕様の検討を進めている。2003 年 4 月末に公開された Verb 仕様は Infiniband の仕様が採用されている。

WinSock Direct(WSD) protocol[7] Windows socket 上で RDMA による通信において利用されるプロトコル

以上の通信プロトコルはいづれも標準 API の策定が主でネットワークの障害解析まで踏み込んだプロトコルとはなっていない。また、既存実装ベースの標準化ではないため、これらのプロトコルを用いて 10Gbps クラスの高速ネットワークでの実装はなされていない。我々が検討しているプロトコルは開発支援ツールを含めてネットワークの障害解析まで踏み込んだプロトコルとなっている点が異なる。

8. おわりに

10 ギガビットクラスの高速ネットワークを用いたクラスタシステム上で次世代の高性能システムを開発するための、通信プロトコル作成支援技術とそれを用いた通信プロトコル開発支援ツールについて述べた。

本通信プロトコル開発支援ツールを用いれば、高い性能とシステムの稼働検証を効率的に行うシステム動作検証とデバッグを効率的に行うことが可能になる。

今後、プロトタイプシステムを完成させ、開発支援ツールの開発を進める。

謝辞 本研究の一部は、文部科学省「eSociety 基盤ソフトウェアの総合開発」の委託を受けた東京大学石川研究室および東京大学石川研究室と富士通研究所との共同研究契約に基づいて行なわれた。

参考文献

1) http://www.pccluster.org/.
2) 石川裕, 住元真司, 岡家豊, 久門耕一, 木村かず子. 次世代高性能計算機アーキテクチャ上のシステムソフトウェア開発環境. 情報処理学会研究報告 03-OS-94 (SWoPP'2003). 情報処理学会, August 2003.
3) 岡家豊, 木村かず子, 石川裕. IPMI 規格に基づく管理保守系システムソフトウェア. 情報処理学会研究報告 03-OS-94 (SWoPP'2003). 情報処理学会, August 2003.
4) 住元真司, 工藤智宏. 高速ネットワークとシステムソフトウェア. 並列処理シンポジウム JSPP'2002 チュートリアル. 情報処理学会, May 2001.
5) http://www.ietf.org/html.charters/rddp-charter.html.
6) http://www.rdmaconsortium.org/home.
7) http://www.microsoft.com/whdc/hwdev/tech/network/san/WSD-SAN.mspx.