計算機アーキテクチャ 142−27
ハイパフォーマンス
コンピューティング 85−27
（２００１．３．９）

Cometによる OC48cクラスタの性能評価

小林 伸治[†1]、的場 宏純[†2]、都筑 俊秀[†3]、陣崎 明[†1]

[†1]新情報処理開発機構 並列分散システム富士通研究室
[†2](株)トライテック
[†3](株)富士通コンピュータテクノロジ

コンピュータ通信の高速化を目的として Comet通信アーキテクチャを開発している。Comet通信アーキテクチャはプロトコル処理をホストプロセッサからネットワークアダプタにオフロードすることで通信を高速化することを特徴としている。FPGA を利用して Comet通信アーキテクチャの一部を実現するアダプタを試作し、OC48cネットワークでクラスタを構築して性能評価を行ったのでその結果を報告する。また、その結果を基に現在開発中の ASIC を使用したアダプタの性能予測を行う。最後に、これらの結果から Comet通信アーキテクチャの方式を評価し、従来のソフトウェアによる処理方式との比較を行う。

Performance Evaluation of the OC48c cluster using Comet

Shinji Kobayashi[†1], Hirozumi Matoba[†2], Toshihide Tsuzuki[†3], Akira Jinzaki [†1]

[†1]Parallel and Distributed Systems Fujitsu Laboratory, RWCP
[†2]Tritech Inc.
[†3]Fujitsu Computer Technology Limited

To achieve fast computer communication, we are developing the Comet communication architecture. Its characteristic feature is to off-load the protocol processing from the host processor to the network adapter. We made the prototype network adapter using FPGA which realizes a part of the Comet communication architecture, and evaluated its performance by constructing the cluster system using OC48c network. In this paper, we report the results of the evaluation and estimate the performance of the next prototype using ASIC which is under development now. Finally, we evaluate the techniques the Comet communication architecture adopts and compare the Comet communication architecture with the common software processing method.

1. はじめに

計算機クラスタの性能向上を図るためには、プロセッサ等計算機単体の高速化だけではなくコンピュータ通信の高速化が重要である。コンピュータ通信に必要なプロトコル処理等は通常ホスト計算機のソフトウェアで行われるため、計算機単体の高速化は通信性能の向上にも寄与する。しかし、近年のネットワークの高速化はプロセッサの高速化を上回る速度で進んでおり、プロトコル処理のオーバヘッドが通信性能に大きな影響をおよぼすようになってきている。

我々はコンピュータ通信の高速化を目的として Comet 通信アーキテクチャを開発している[1][2][3][5]。Comet 通信アーキテクチャは、プロトコル処理をネットワークアダプタにオフロードすることで通信を高速化することを特徴としている。今回、FPGA を利用して Comet 通信アーキテクチャの一部を実現するアダプタを試作し、2.5Gbps の OC48c ネットワークで相互結合したクラスタシステムを構築した。このクラスタを利用して性能評価を行ったので、その結果を報告する。また、その結果に基づいて、現在開発中のネットワークプロセッサを使用したアダプタの性能予測を行い、Comet とソフトウェア処理方式との比較を行う。

2. コンピュータ通信の高速化

2.1. Comet通信アーキテクチャ

Comet はプロトコル処理をネットワークアダプ

タにオフロードすることで高速な処理を可能にする通信アーキテクチャである。プロトコル処理をネットワークアダプタで行う方式としては、アダプタ上の汎用プロセッサで処理を行うインテリジェントアダプタ方式がある。しかし、インテリジェントアダプタ方式ではアダプタ上のバッファメモリに格納してから処理を行うため遅延が大きくなる、アダプタ上に搭載できるプロセッサはコストや消費電力の問題から性能が限られるためホストプロセッサで処理した方が速い、といった問題があった。

Comet 通信アーキテクチャでは、インテリジェントアダプタとは異なり、SP (Stream Processor) [4] を用いることで DMA 転送中にパケット処理を行う。このため処理の遅延を最小限に抑えることができ、搭載する制御プロセッサもさほど高速である必要はない。Comet 通信アーキテクチャでの処理の流れは図 1 のようになる。

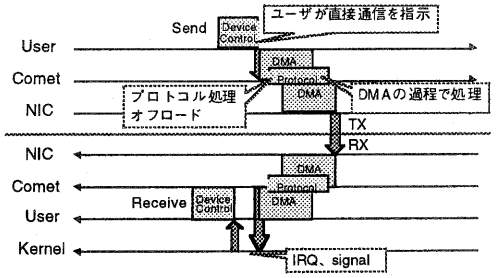

図 1: Comet での通信処理の流れ

図のように各要素が並行動作できるため、ユーザレベルでの送信処理に要する時間、DMA 設定時間、DMA 実行時間のうちの最大値が全体のバンド幅を決定する。一般には、長いパケットでは DMA 時間が支配的となり、短いパケットでは DMA 設定時間がボトルネックとなる。

Comet 通信アーキテクチャの特徴をまとめる。

- プロトコル処理をアダプタ上の SP で行う。
- ユーザレベル通信を行い、アプリケーションが直接アダプタを操作する。
- データ転送にはアダプタ上の DMA エンジンを用いる。
- 送受信用バッファアドレスの通知等、ホストとアダプタとの情報交換には FIFO メモリを用いたインタフェースを使用する。

2.2. Comet-CP アダプタ試作

Comet 通信アーキテクチャの性能評価を行うため、SP の機能の一部を実現する FPGA を持つインテリジェントアダプタ、Comet-CP ネットワークアダプタを試作した。Comet-CP はフルサイズの PCI カードであり、PMC (PCI Mezzanine Card)規格のネットワークインタフェースカード(NIC)を装着して使用する。今回は NIC として 2.5Gbps の OC48c カードを利用した。Comet-CP のブロック図を図 2 に示す。

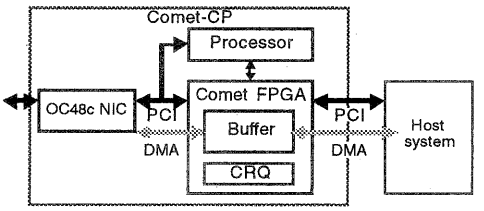

図 2: Comet-CP アダプタブロック図

Comet FPGA は DMA エンジン、バッファメモリ、ホストインタフェース用の CRQ (Command Response Queue)を持つ。ホスト、Comet-CP 間の DMA には Comet FPGA の DMA エンジンを用い、Comet-CP、NIC 間の DMA には NIC の DMA エンジンを用いる。Comet-CP 全体の制御は汎用プロセッサ (StrongARM) で行う。

Comet-CP は SP を持たないので DMA 転送中にプロトコル処理を行うことはできないが、DMA 設定やバッファ管理のオーバヘッドを評価することができる。今回の実装では IP プロトコル処理は Comet-CP では行っていない。OC48c のフレーミング処理や OC48c スイッチの制御は Comet-CP の制御プロセッサで行っている。Comet-CP の主な諸元を表 1 に示す。

表 1: Comet-CP アダプタの諸元

制御プロセッサ	StrongARM 233MHz
PCI	64bit/33MHz
バッファメモリ	32KB
CRQ (FIFO メモリ)	32bit×256 ワード×4
ネットワーク	OC48c (2.5Gbps)

ホストプロセッサとのインタフェースは、FIFO を用いたコマンド/レスポンス方式を採用した。

Comet-CP は CRQ と呼ぶ 32bit 長 256 ワードの FIFO メモリを 4 本備えており、これらのうちの 2 本をホストから Comet-CP への通知、Comet-CP からホストへの通知にそれぞれ使用する。

送信時には、送信バッファのアドレスや長さを含んだ数ワードからなる送信要求コマンドをホストが CRQ に書き込む。Comet-CP は DMA を起動して送信バッファの内容を Comet-CP に転送し、完了後にそれを送信要求コマンドのレスポンスとして CRQ 経由でホストに通知する。Comet-CP は CRQ に情報を書き込むとホストに割り込みをかける。ホストは送信要求コマンドのレスポンスにより、送信バッファを解放する。

受信用には、ホストがあらかじめ受信バッファを割り当てておき、そのアドレスと長さを受信バッファ設定コマンドにより Comet-CP に通知する。Comet-CP はネットワークからのデータをそこに転送しおえると受信バッファ設定コマンドのレスポンスとしてホストに通知する。

OC48c では独自形式で IP パケットを転送している。MTU は 8160bytes である。

3. OC48c クラスタの性能評価

3.1. Comet-CP アダプタの性能評価

Comet-CP アダプタで 64bytes の ICMP パケット送受信に要した時間の内訳を表 2 に示す。

表 2: Comet-CP 処理時間の内訳

処理内容		時間(μs)
送信	送信要求検出・DMA 設定	1.91
	ホスト側 DMA 実行	2.23*
	DMA 完了検出・ホスト通知	1.59
	NIC 側 DMA 設定	1.82
	NIC 側 DMA 実行	7.36*
受信	受信検出・ホスト側 DMA 設定	2.29
	ホスト側 DMA 実行	1.18*
	DMA 完了検出・ホスト通知	1.58
	NIC 側 DMA 設定	0.20

* パケット長に依存するが、ここでは最短値

DMA 実行時間はホストの構成およびパケット長に依存する。64bit/33MHz PCI を持つ Pentium-III-Xeon 733MHz 機で測定した DMA のバンド幅を図 3 に示す。

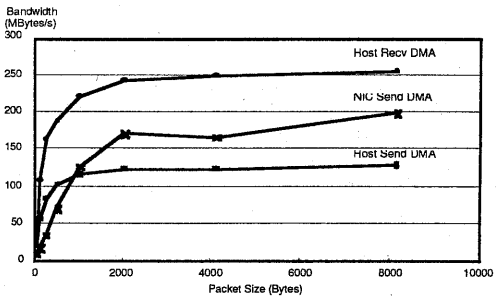

図 3: Comet-CP DMA のバンド幅

64bit/33MHz の PCI バスを使用しているので、最大バンド幅は 267MB/s である。ホスト側受信 DMA、すなわち Comet-CP からホストメモリへの書き込みは最大バンド幅に近い性能が得られている。これに対して、ホスト側送信 DMA、すなわちホストメモリから Comet-CP への読み出しでは約半分の DMA 性能しか得られず、最大で 128MB/s である。また、NIC 側 DMA は起動や完了検出のオーバヘッドが大きく、小さいパケットでは効率が悪いことがわかる。

3.2. OC48c クラスタの性能評価

ノード間接続に Comet-CP を用いたクラスタシステム(表 3) で性能測定を行った。

表 3: OC48c クラスタの構成

プロセッサ	Pentium III-Xeon 733MHz
メモリ	512MB RDRAM
OS	RedHat Linux 6.2 (kernel-2.2.14)
ノード数	4

測定には Linux 標準の TCP/IP プロトコルスタック、および新情報処理開発機構が開発したクラスタ向け通信ライブラリ、PM[6]を使用した。

まず、2 ノード対向でアプリケーションレベルのバンド幅を測定した。netperf[7]で TCP/IP および UDP/IP のバンド幅を測定し、PM のバンド幅は rpmtest で測定した。ftp では MTU を変化させながら 100MB のファイルを転送した。NFS では、リモートの RAM ディスクを NFS マウントし、マウントオプションのサイズを変化させて測定した。測定結果を図 4 に示す。

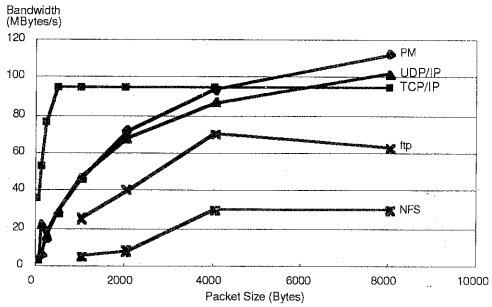

図 4: OC48c クラスタのバンド幅

TCP/IP で 95MB/s、PM では 114MB/s を達成している。これは、ホスト側 DMA の送信最大バンド幅、128MB/s に近い。

次に、PM 上で NAS Parallel Benchmark (NPB) [8] の Class A を実行した。4 ノードでの台数効果を表 4 に示す。比較対象として、通常の Fast Ethernet (100Base-TX) NIC での測定結果も示す。

表 4: NPB の台数効果 (4 ノード)

プログラム	台数効果	
	OC48c	Fast Ethernet
cg	3.61	2.36
ep	3.97	4.00
ft	1.91	2.05
is	0.17	0.77
lu	3.75	3.69
mg	3.38	3.12
sp	3.53	3.47
bt	3.36	3.35

バンド幅の差が性能を左右する cg で台数効果が上がっているのがわかる。is の性能が悪い理由は今後調査する必要があるが、Fast Ethernet での性能と大きな違いがないことから、4 ノード規模の NPB では処理ネックであり、ネットワーク性能は影響しないことがわかる。

4. Comet-NP の性能予測

4.1. Comet-NP 概要

現在、我々は Comet 通信アーキテクチャに基づいたネットワーク処理用プロセッサ、Comet-NP を ASIC として開発中である。評価用 LSI は既に入手済みであり、評価用ボードでデバッグおよびファームウェアの開発を進めている。

Comet-NP ASIC は 64bit/66MHz の PCI バスを備え、SP を 2 つ搭載している(図 5)。

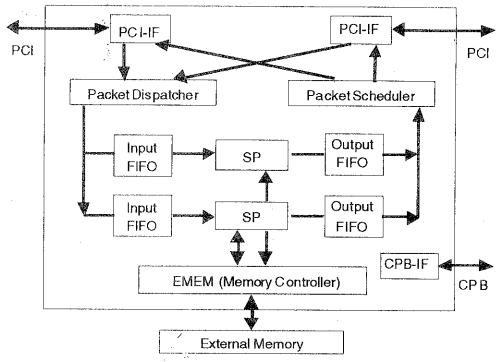

図 5: Comet-NP ASIC ブロック図

Comet-NP ASIC を使用した Comet-NP ネットワークアダプタは、SP により IP や UDP/IP のプロトコル処理を高速に実行できる。SP はプログラマブルなので IPv6 にも容易に対応できる。Comet-NP ネットワークアダプタのブロック図を図 6 に示す。

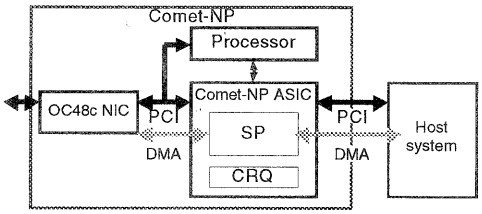

図 6: Comet-NP アダプタブロック図

Comet-NP ネットワークアダプタは、Comet-CP ネットワークアダプタと比較して次のような特徴を備えている。

- SP の搭載
- 64bit/66MHz PCI

SP の搭載により、DMA 転送中にプロトコル処理を実行できる。たとえば、SP はチェックサム計算やテーブル検索の機構を備えているため、パケット毎のチェックサム計算や経路表検索を高速に実行できる。時間のかかる処理をホストから Comet-NP アダプタにオフロードできるため、ホストが通信処理に要する時間は小さくなる。ユーザレベル通信を採用することでコンテキストスイッチのオーバヘッドも削減できる。

4.2. Comet-NP アダプタの性能予測

前節で述べた Comet-CP アダプタでの性能測定結果から、Comet-NP アダプタの性能予測を行う。Comet-NP を利用することで性能上影響するのは以下の点となる。

NIC 側 DMA は常に SP に対して行うため、ディスクリプタの設定は非常に簡略化できる。NIC 側 DMA エンジンの構造にも依存するが、多くの場合 1 つのディスクリプタを使いまわすことができるので NIC 側 DMA の設定時間をほぼ無くすことができる。

PCI が 64bit/66MHz になることで、DMA の最大バンド幅が 2 倍の 533MB/s になる。実際の実効バンド幅はホストシステムのチップセットやメモリの性能に依存するが、ここでは図 3 の 2 倍の DMA 性能が得られると仮定する。

バンド幅を決定するのは、ホスト側での処理時間、Comet-NP 制御プロセッサによる DMA 設定等の時間、DMA 実行時間の最大値である。ホスト側処理の多くを Comet-NP にオフロードできること、および図 4 でプロトコル処理の差が小さいことから、ホスト側での処理時間は十分小さいと仮定できる。Comet-NP の制御プロセッサとして Comet-CP と同じ StrongARM 233MHz を使用すると、DMA 設定等の時間は表 2 の値から NIC 側 DMA 設定時間を除いた値で見積もることができる。以上をまとめて、無限大のバンド幅を持つネットワークを利用した場合の Comet-NP アダプタの性能を予測した(図 7)。

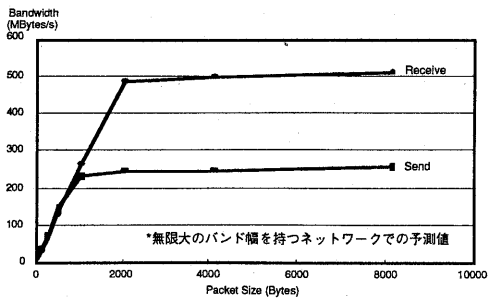

図 **7: Comet-NP** アダプタ性能予測

Comet-NP アダプタでは 1000bytes 以上のパケットで最大 250MB/s 程度の性能が見込める。

5. 考察

5.1. I/O バスと DMA

Comet はネットワークアダプタ上に DMA エンジンを持ち、データ転送にはこの DMA エンジンを用いることを基本にしている。また、現在の試作では I/O バスとして一般的な PCI を採用している。図 3 に示したように、PCI バスに対する書き込みでは高いバンド幅が得られている。特に、2000bytes 以上のパケットでは計算上の最大バンド幅に近い性能を発揮する。

しかし、PCI バスからの読み出しではバンド幅が約半分に低下してしまう。これは、I/O バス越しにアクセスする場合、書き込みより読み出しの方が遅いことを意味している。

データ転送に DMA を利用することで高いバンド幅を実現できることは確認できたが、送信時のバンド幅を向上させるためには、ホストプロセッサ側にも DMA エンジンを持たせるようなアーキテクチャや、ホストプロセッサによるプログラム I/O を検討する必要がある。特に、2000bytes 以下の小さいパケットを送信するときはプログラム I/O の方が高いバンド幅を得られる可能性がある。I/O バスによるバンド幅の制限を解決する方法としては、DIMMnet-1[9]等が提案されているが、Comet 方式はこのような場合にも適用可能である。

5.2. CRQ を利用したインタフェース

アダプタ上の DMA エンジンでデータを転送する場合、ホストメモリ上にあるデータのアドレスや長さをアダプタが知る必要がある。このために広く使われているのがディスクリプタ方式である。

ディスクリプタ領域をホストメモリに置く場合、DMA エンジンが I/O バス越しにディスクリプタを読むのがオーバヘッドになる。ディスクリプタ領域をアダプタ上のメモリに置けばこのオーバヘッドは解消できるが、ホストプロセッサがディスクリプタ領域を管理する必要がある。

アダプタ上の CRQ を利用する方式では、ホストプロセッサはディスクリプタ領域等を管理する必要はない。ホストからアダプタに情報を通

知する場合、I/O バスに対しても書き込み操作となる。これは前節で議論したように効率がよい。しかし、DMA 完了通知のようにアダプタからホストに情報を渡す場合は I/O バスに対する読み出し操作となってしまう。アダプタからホストへの情報伝達手段には改善の余地がある。

5.3. ソフトウェア処理方式との比較

第4節の予測により、Comet 通信アーキテクチャで高いバンド幅が得られることがわかった。特に、2000bytes 以上の大きなパケットではボトルネックは PCI バスのバンド幅である。将来、より高速な I/O バスが利用できるようになれば、さらに高いバンド幅を実現できると考えられる。

一方、ホストプロセッサのソフトウェアでプロトコル処理を行う従来のアーキテクチャでも、今回測定を行った環境ではプロトコル処理がボトルネックとはなっていないと考えられる。これは、図 4 で TCP/IP と PM の差があまりないことから推測できる。ソフトウェア処理の最大バンド幅を求めるため、3.2節の測定と同じホストで自分自身のループバックインタフェースに対して UDP/IP 通信を行う実験をした結果、UDP チェックサム有りで 150MB/s、UDP チェックサム無しで 180MB/s という値が得られた。1台で送受信双方を行っていることを考慮すると、2 倍の 300MB/s、および 360MB/s 程度が 733MHz プロセッサによるソフトウェア処理による UDP/IP の最大バンド幅と考えられる。すなわち、64bit/66MHz PCI バスで送信 DMA の実効バンド幅が 250MB/s 程度であれば、ソフトウェア処理方式でも現在のプロセッサで対応可能である。

しかし、これは 250MB/s 程度の通信を行うとプロセッサはほとんど通信処理にかかりっきりになるということでもある。NPB の測定結果を見ても、通信性能よりも個々の計算性能の方が全体の性能に寄与するプログラムが多いことがわかる。プロセッサの処理能力の大半が通信処理にまわってしまうと本来の計算が行なえず、台数効果が上がらないといった状況も予想される。この意味で、通信処理をアダプタにオフロードすることは意味があると考えられる。

6. まとめ

通信処理をアダプタにオフロードすることを特徴とする Comet 通信アーキテクチャの評価を目的として FPGA による Comet-CP と OC48c ネットワークを用いたクラスタを開発し、通信性能や並列計算性能を測定した。また、この測定結果をもとにプロトコル処理を専用のネットワークプロセッサで行う Comet-NP の性能予測を行った。その結果 Comet-NP は I/O 帯域を考慮しても 250MB/s 以上の性能が実現可能なことがわかった。これは 700MHz クラスのプロセッサで 100%通信処理する負荷を Comet-NP にオフロードすることに相当する。このことから Comet 方式は有効と考えられる。今後は Comet-NP ASIC を使用したシステムで性能評価を行う。

謝辞

性能測定に協力していただいた新情報処理開発機構の水野裕識氏に感謝いたします。

参考文献

[1] 小林伸治, 陣崎明: Comet における VIA 的インタフェース, 信学技報, Vol. 98, No. 234, CPSY98-63, pp. 23-28, 1998.

[2] 小林伸治, 陣崎明: Virtual Interface Architecture の Internet 拡張方式, 並列処理シンポジウム JSPP'99, pp. 23-30, 1999.

[3] 陣崎明, 中村修, 村井純: 並列ネットワークサーバ Comet のアーキテクチャとその応用, 信学技報, Vol. 98, No. 234, CPSY98-62, pp. 15-22, 1998.

[4] 陣崎明: Stream Processor, 並列処理シンポジウム JSPP2000, 2000.

[5] 小林伸治, 陣崎明: Comet-VIA の評価, 信学技報, Vol. 100, No. 249, CPSY2000-53, pp. 9-16, 2000.

[6] http://pdswww.rwcp.or.jp/

[7] http://www.netperf.org/

[8] http://www.nas.nasa.gov/Software/NPB/

[9] 田邊昇, 山本淳二, 工藤知宏: メモリスロット搭載型ネットワークインタフェース DIMMnet-1 における細粒度通信機構, 情処研報, Vol. 2000, No. 23, 2000-ARC-137, pp. 65-70, 2000.