次世代超並列計算機開発
— 連続体向け超並列計算機の開発 —
Next-Generation Massively Parallel Computers
— Development of Massively Parallel Computers for Continuous Physical Systems —

岩崎洋一 (筑波大学副学長 (研究担当))

1 研究組織

リーダー	岩崎洋一	筑波大学副学長 (研究担当)	
コアメンバ	宇川彰	筑波大学物理学系教授	(AIOV)
	金谷和至	筑波大学物理学系助教授	(AIOV)
	青木慎也	筑波大学物理学系助教授	(AMOC)
	吉江友照	筑波大学物理学系助教授	(AMOC)
	梅村雅之	筑波大学物理学系助教授	(AIOV)
	中本泰史	筑波大学物理学系助手	(AIOV)
	大川正典	高エネルギー加速器研究機構素粒子原子核研究所助教授	(AIOV)
コアメンバ	坂井修一	東京大学大学院工学系研究科電気工学専攻助教授	(MOC)
	朴泰祐	筑波大学電子・情報工学系助教授	(IOV)
	山下義行	筑波大学電子・情報工学系助教授	(IOV)
	和田耕一	筑波大学電子・情報工学系助教授	(MOC)
	安永守利	筑波大学電子・情報工学系助教授	(IOV)
	千葉滋	筑波大学電子・情報工学系講師	(IOV)
	星野力	筑波大学構造工学系教授	(IOV)
	白川友紀	筑波大学構造工学系助教授	(IOV)
	中村宏	東京大学先端科学技術研究センター助教授	(MOC)
	渡瀬芳行	高エネルギー加速器研究機構計算科学センター教授	(IOV)
	中澤喜三郎	明星大学情報学部教授	(MOC)
	中田育男	図書館情報大学図書館情報学部教授	(IOV)

( ) は主な役割分担；
- AIOV: 物理応用計算における並列入出力・並列可視化
- AMOC: 物理応用計算におけるプロセッサ・メモリ混載型 LSI
- IOV: 並列入出力・並列可視化
- MOC: プロセッサ・メモリ混載型 LSI

2 研究の目的

計算科学の最近の発展は、超並列計算機による計算機の能力の向上と強く結びついている。超並列計算機はベクトル計算機と比較してそのアーキテクチャが多様であり、問題を明

確に設定して初めて、最先端の半導体技術を駆使した超高性能なシステムを実現できる。素粒子物理学をはじめ、宇宙物理学、物性物理学などの物理学各分野には、連続体系が多く現れるが、先の CP-PACS プロジェクトはこれらに焦点をあてて、高性能な超並列計算機の開発に成功した。本研究開発では、この方向をさらに押し進め、連続体向けの超並列計算機に関して、 (1) 超並列計算機の課題である、高速かつ柔軟な入出力機構・可視化機構・マンマシンインタフェースを実現すること、 (2) 計算物理学の難問解決の要請に応え、超並列計算機の計算速度を現在の 1TFLOPS から 100 倍向上させるための計算機アーキテクチャを考案・検証すること、を目標とする。

並列入出力・並列可視化　超並列計算機における科学技術計算の過程で生じる、大量のデータに対する入出力処理の高速化と、計算の中間及び最終結果に対し、その正当性・妥当性の直感的理解を助けるデータの可視化は、超並列計算機をより利用し易くするための重要な技術的要件である。これらを実現するために、我々は高性能かつ柔軟で、しかも安価な超並列計算機向け入出力システム及び可視化システムを構築することを目的としている。特に、超並列計算機が備えている、多数の入出力プロセッサを並列運用し、かつユーザからはそれらの運用を特に意識せずに、外部環境との柔軟なやりとりを容易に実現できるようなアプリケーションインタフェースを提供する。また、ハードウェア開発にコストと時間をかけず、近年の commodity 化したネットワーク媒体や接続技術を積極的に利用することにより、性能価格比に優れたシステム構築手法を発見することも重要である。

次世代プロセッサアーキテクチャ　素粒子・宇宙・物性等、計算物理学の主要分野で利用可能な計算速度は、現在 1TFLOPS に達しており、これによって大幅な進歩がもたらされているが、さらに数百 TFLOPS の計算性能の実現によりはじめて真に現実的な計算が可能となる問題も数多くある。本研究では、このような計算物理学の要求に応え、次世代の連続体向け超並列計算機の要素となる LSI のアーキテクチャを明らかにすることを目指す。大規模科学計算においては、計算速度のみならず、この性能に見あった記憶装置から演算装置へのデータ供給性能が要求される。これを満足する有力な方法として、プロセッサとメモリを同一チップ上に混載する方法がある。 LSI の高集積化・高速化を中心とする計算機技術の発展は本研究のターゲットとする 21 世紀初頭において持続するものと考えられる。本研究では、 2004 年前後に実現されるデバイス技術の予測に基づき、プロセッサ・メモリ混載型 LSI を用いた計算機アーキテクチャを考案し、シミュレーション評価、詳細設計を行なう。

3　研究計画の概要

3.1　並列入出力・並列可視化システム

過去の超並列計算機プロジェクトでは、多くの場合、入出力系用の専用ハードウェア・ソフトウェアが導入・開発されたが、現在では、汎用のバス及びネットワーク技術は、これらの研究で目標とされた水準に十分達している。特に、 100 Mbit/ 秒程度の通信性能を持

つ 100base イーサネットや ATM は、性能価格比で上記ネットワーク技術を大きく上回っている。またイーサネットの Gigabit 化が実用化されており、そのコストも急激に低下しつつある。従って、今後の超並列計算機用入出力系としては、汎用製品 (commodity) をハードウェア面で積極的に利用し、ソフトウェア的手法でその利用率を高める方向が開発期間・コストに対する効率の点で有利であると考えられる。

分散メモリ型超並列計算機である CP-PACS には、入出力処理のための専用プロセッサが多数用意されており、大容量の内蔵ディスクに加え、多数の外部入出力チャネルをサポートすることが可能である。従って、 100base イーサネット等の commodity 技術を、多数の並列入出力プロセッサに適用することにより、全体として十分な通信性能を持ち、性能価格比に優れた並列入出力チャネルを提供することが可能であると考えられる。

また、外部環境として存在する他の計算機資源においても、その計算機性能に応じた並列なネットワーク環境が提供されつつある。ワークステーションクラスタ等では、 1 プロセッサ当たり 1 つのネットワークインタフェースが実装されているのが普通である。従って、超並列計算機とこれらのシステムを一対一に結合する際に、並列ネットワークを用いて多数のデータ流の同時転送を行うことが可能である。さらに、ネットワークと同様に低価格化が進んでいる高速スイッチを介することにより、複数の計算資源間を並列接続して、高性能かつ柔軟なネットワークシステムを提供することが可能である。

これらの要素をユーザ・アプリケーションから利用する際に、並列チャネルの存在を意識しつつ通信負荷の分散を行うことは、ユーザ・プログラムを繁雑化し、またシステム構成の変更等に対する適用性を損ねる恐れがある。従って、本システムを実用化するためには、並列チャネル数の増減や、同時実行中の他のユーザ・アプリケーションとの負荷バランス、あるいは接続先の相手計算機のチャネル数などの、データ転送負荷の制御に直接関係するシステム情報をアプリケーション側から隠蔽し、常に最適な状態で入出力チャネルの選択・バッファリング・制御を行うよう、システムを自動化することが必要となる。

可視化システムについては、現在の画像処理専用ワークステーション等を用いれば、最終的なフレームバッファの描画等の点では専用ハードウェアにより十分な高速化が達成されている。しかし、素データに対する画像処理においては、プロセッサ速度の限界等により、より一層の高速化が望まれている。並列入出力システムに統合された可視化環境においては、素データは並列データ流として可視化システムに流入する。よって、この段階でデータに対する空間分割的手法によって、画像処理そのものをできる限り並列化することにより、実時間処理に対応できるスループットを実現することが可能と考えられる。これらを前提に、可視化システムを並列入出力システムの一部として実装し、並列プロセッサを持つ専用ワークステーションにより、データ入力・素データに対するデータ処理をできる限り並列化し、スループットの向上を目指す。

3.2 プロセッサ・メモリ混載型 LSI

米国半導体工業会 (SIA, http://notes.sematech.org/ntrs/PublNTRS.nsf) のロードマップによれば、本研究のターゲットとする 2004 年の前年には、チップ I/O 数は 3000、 DRAM

ビット数 1Gbit(product)、 MPU トランジスタ数 76M、電源電圧 1.2-1.5V、クロック周波数 2.1GHz、最大消費電力 130W となる。 CP-PACS で使われている技術に比べると、 DRAM チップの記憶容量で 16 倍、プロセッサのトランジスタ数で約 17 倍、内部クロック周波数で 14 倍、 I/O バスクロック周波数で約 11 倍になる。これは、数年前に予測されていたものに比較して、トランジスタ数がやや少なくなり、クロック周波数が大幅に増えたものになっており、この点に注意が必要である。

次世代の計算科学向け計算機は、 (1) ピークの演算速度、 (2) 演算装置へのデータ供給速度、の 2 点から最適化されたアーキテクチャをとる必要がある。このうち、演算速度の向上は、一般に (a) クロック速度の向上、 (b) スーパスカラや VLIW によるプロセッサ内並列化、 (c) 高集積化によって要素プロセッサを多く実装することによる並列化、 (d) 投機的実行による並列化、などによって実現される。また、データ供給系の高速化は、メモリ階層のそれぞれにおけるバンド幅およびレイテンシの最適化によって実現される。具体的には、データサイズとデータアクセスパターンに応じて、メモリバスの拡大、メモリの多バンク化、多ポート化、 CP-PACS で開発された疑似ベクタ処理機構の改良、キャッシュの最適化などを行う必要がある。

2004 年に実現される高集積チップは、集積度が現在のそれと比較して 17 倍と大きいにもかかわらず、チップの I/O 数はたかだか 3.5 倍しかないため、メモリをプロセッサチップ外部に置く従来のアーキテクチャでは、バスネックになる可能性があり、データサイズの大きい科学計算では、特にその可能性が大きい。この問題を解決するため、プロセッサ・メモリ混載型 LSI の開発が考えられる。この LSI においては、メモリバスはチップ内部に実装されるため、バンド幅が大きくレイテンシの小さなバスが実現可能である。

科学技術計算のデータサイズは、ほぼ CPU 能力に比例する。現在の CP-PACS の PU あたりメモリ量は 256MB 程度であり、 2004 年にクロックが 14 倍になると、約 3.6GB が必要となる。 STARC(http://www.starc.or.jp) は、 2004 年のプロセッサで混載可能な DRAM の大きさを 1Gb 程度と予測しており、単純なプロセッサ・メモリ混載チップでは、メモリが大幅に不足するため、本研究の目的とする科学技術計算向けのプロセッサを作ることはできない。

我々は、この点から従来のプロセッサアーキテクチャを考え直し、大容量の主記憶を外付け DRAM で実現する一方、内部に中規模の容量の RAM を持つ新たなプロセッサアーキテクチャを提案する。外付け DRAM はバンク分けして高スループットを実現し、 LSI に内蔵された RAM は、 (1) レーテンシ隠蔽、 (2) 中間結果の格納と高速読み出し、の 2 つの目的で使われる。

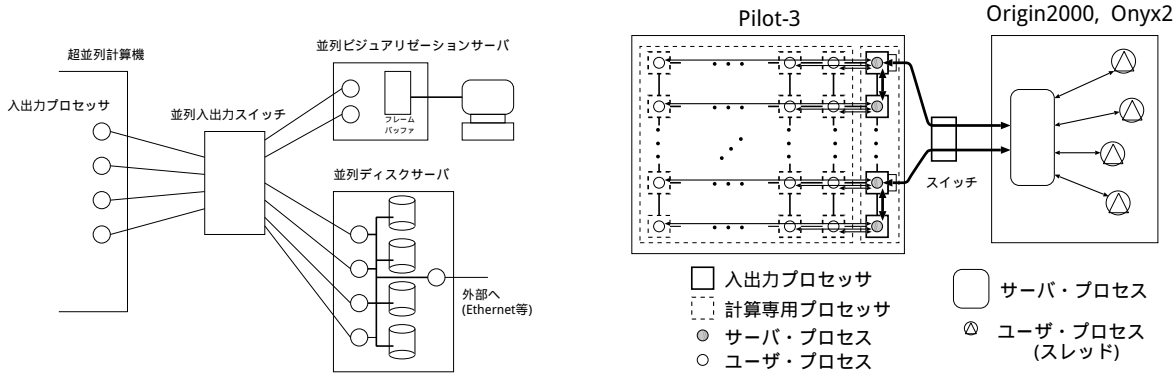
2004 年の計算機は、 CP-PACS と比較して、クロック周波数で 14 倍、チップ内並列性能で 4 倍となることが期待される。メモリバンド幅、キャッシュ速度、レジスタバスのバンド幅のそれぞれで、これに見合う速度のデータ供給を行なうことができれば、実装規模を約 10 倍と仮定した単純計算で、ピーク性能は 300TFLOPS 程度となる。

4　平成１０年度の研究成果の概要

4.1　並列入出力・並列可視化システム

4.1.1　実験環境

平成 10 年度における並列入出力・可視化システムの研究では、図 1(a)に示すようなシステムを構築し、その上に並列入出力プログラミングを行なうためのプロトタイプ API (Application Program Interface) を設計・実装し、その性能を評価した。さらに、並列データ流に基づく可視化システムのプロトタイプを設計・実装し、並列入出力システムの一部として、業界標準可視化ソフトウェアである AVS に統合した。



(a) 全体構成　　(b) ユーザプロセスとサーバの関係

図 1: 並列入出力・可視化実験システム

図中、超並列計算機は CP-PACS のプロトタイプ機 (Pilot-3) であり、並列入出力実験用に、 100base-TX イーサネット・インタフェースを実装した入出力プロセッサを 4 台配置している。これを 100base-TX スイッチを経由し、並列ディスクサーバである Origin-2000 (8 プロセッサ) 及び並列ビジュアリゼーションサーバである Onyx2 (2 プロセッサ +1 ラスタマネージャ) と結合している。これらのワークステーションにも、各々 4 ポートの 100base-TX イーサネット・インタフェースが設置されており、各入出力チャネルは互いに 4 本ずつの 100base-TX により並列結合されている。

4.1.2　並列入出力システム

並列入出力システムの概要　並列イーサネットを用いた予備的な転送実験の結果、各計算機資源におけるネットワーク入出力では、データ転送要求をなるべく独立な (相互作用のない) パケットに分割し、それらを並列に処理及び転送することによって、全体のスループットを線形に増加させることが可能であることがわかった。これを用いて、超並列計算機によって生成されたデータを、並列チャネルを用いて直接外部環境に送出するシステムを、ユーザに使い易い形で提供する。

このためには、アプリケーション・プログラムから見て、並列入出力チャネルが容易にアクセスできる必要がある。しかし、並列プロセッサから生成されるデータ流は場合によっては空間的及び時間的に粗密が生じる可能性があり、これをうまく平滑化し、チャネル負荷を均等にする必要がある。分散メモリ型アーキテクチャを持つ超並列計算機においてこのような負荷分散を自動化するために、入出力要求を並列処理しつつ協調動作する、システム・デーモンを実装する (図 1(b)参照)。これにより、ユーザは並列入出力システムの存在を意識せずに、入出力システムを効率的に運用できるようにする。

並列入出力のための API　本研究で開発する並列入出力用 API (Application Program Interface) を、 “PIO システム” と名付ける。 PIO システムは (1) システムで利用可能な並列入出力チャネルの構成要素情報の提供、 (2) データ入出力操作、 (3) 接続先システムの情報の提供、 (4) 手動または自動による通信負荷の空間的均等化 (データ生成プロセッサ毎の負荷分散)、 (5) 通信負荷の時間的均等化 (時系列上のデータに対する負荷分散)、といった機能を提供するライブラリ群からなる。

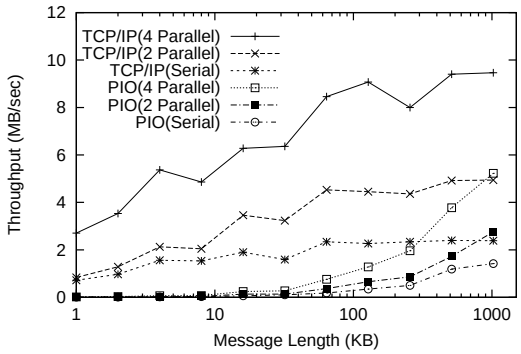
本 API では、システム構成のうち、最低限の情報のみをユーザに提供、あるいは操作させ、大部分を自動化することを目的としている。明示的な情報操作をしない限り、ユーザプログラムからの入出力要求は、並列チャネルのいずれかに自動的に割り振られる。また、PIO システムは、入出力双方の計算機における並列性を想定している。このため、一般的な TCP/IP 等の通信と異なり、ホストのアドレスだけでなく、そのホスト上の並列プロセッサまでを通信パートナーとして識別できるようになっている。

現在の実装対象は、 CP-PACS システム上の OSF-1/MPP 及び Origin-2000, Onyx2 上の IRIX-6.2 である。これらのシステム上には、各々、実行時のサポートを行なうデーモンが実装され、各システムにおいて最適な通信を行なうようになっている。その際、不必要なデータコピーによるオーバヘッドを低減するため、 CP-PACS 側では PU 間のゼロコピー通信機能を、 Origin-2000 及び Onyx2 側では共有メモリによるサーバとユーザの両プロセス間でのデータ共有を利用している。

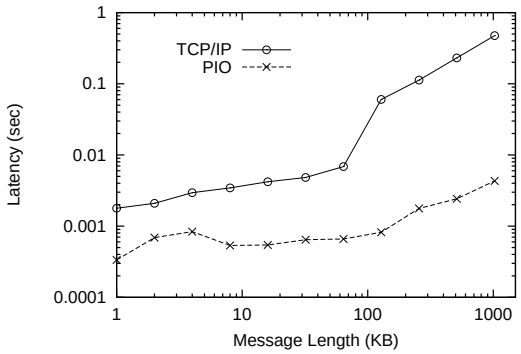
PIO システムの性能評価　PIO システムを用い、 CP-PACS と Origin-2000 間での簡単な ping-pong 転送を行なった場合の通信性能を図 2(a)に示す。ここでは両システム上で 4 対のユーザプロセス間での通信の総スループットを示している。ユーザアプリケーションのプログラムを全く変更せず、通信チャネル数を増加させた場合に、性能が向上しているのがわかる。しかし、絶対的なスループットという点で見ると、最大でも裸の TCP/IP 通信性能の約半分程度の性能しか出ていない。これは、システム中の各種レイヤでのソフトウェアオーバヘッドの影響である。

しかし、本 PIO システムの利用法としては、このような双方向の通信よりも、一方向での非同期な大量通信が主であると考えられる。この点で重要なのは、データ生成側のユーザプロセスにとって、いかに PIO へのアクセス時間が短く見えるかが重要である。この点では、 PIO システムを構成している各種デーモンがバッファの役割を果たすため、 CP-PACS

上で裸の TCP/IP 通信を用いて外部と通信した場合に比べ、非常に小さな (見かけ上の) レイテンシで処理が行なえる。一方向通信を行なった場合の、ユーザプロセスに処理が戻るまでの時間を図 2(b)に示す。図からわかるように、ユーザプロセスは出力処理から短時間で解放され、処理を継続することができる。ここでは、多数の入出力プロセッサを利用できる、超並列計算機のメリットが活かされている。



(a) ping-pong 転送におけるスループット



(b) ユーザプロセスから見たレイテンシ (単方向通信)

図 2: 並列入出力システムの性能評価

4.1.3 並列可視化システム

並列データ入力 並列可視化システムのプラットフォームは、並列画像処理専用ワークステーション Onyx2 である。平成 10 年度に導入したシステムは、 2 台の CPU と 1 台のラスタマネージャ (画像生成エンジン)、さらに 4 チャネルの 100base-TX インタフェースを備えている。 2 台の CPU を用いることにより、一連の可視化処理の負荷分散、例えば並列データ入力とその画像処理といった機能分散が可能であり、 CPU 台数のさらなる増加にも対応できるようになっている。この上に、複数プロセッサによるマルチスレッド処理により、PIO システムを介してデータ生成システムから転送される並列データ流を、最終的な画像処理プログラムに渡すための単一流にまとめるプログラム (Binder と名付けられる) を開発した。 Binder は PIO の API に基づき、 PIO デーモンとの共有メモリを介したデータ受け渡しを行ない、生成されたデータ流を画像生成システムに渡す働きを担うものである。

AVS システムのネットワーク化及び並列化 本可視化システムでは、画像生成のためのソフトウェアとして、業界標準である AVS を用いる。 AVS は単一プロセッサ上での動作を想定しており、さらにデータの供給がそのシステム (ワークステーション) 上のファイル環境として閉じていることを前提としている。このため、そのままの形では我々の目的である、並列ネットワークを介したオープンな環境での利用に適さない。そこで、 AVS のデータ入力モジュールを構築し直し、 (1) ネットワーク環境における外部からのデータ供給、 (2) 共

有メモリを利用した他プロセスからのデータ供給、の 2 機能を実現した。これらの機能は、AVS のユーザ・モジュールとして実装され、 AVS システムの一部に取り込まれるようになっている。

本データ入力モジュールを、先述の Binder と共有メモリを介して結合することにより、超並列計算機によって生成されたデータを並列入出力システムで転送し、画像処理の直前まで並列化されたデータ流を供給することが可能となっている。この様子を図 3に示す。

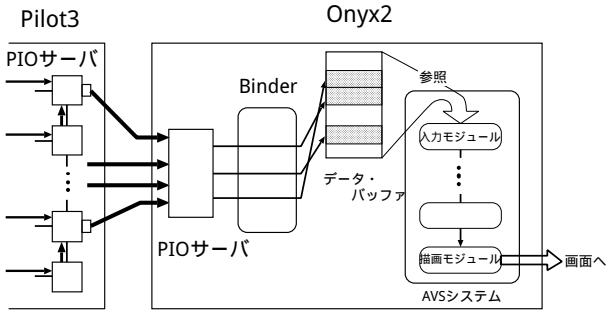


図 3: 並列入出力から並列可視化システムへのデータの流れ

4.2 プロセッサ・メモリ混載型 LSI

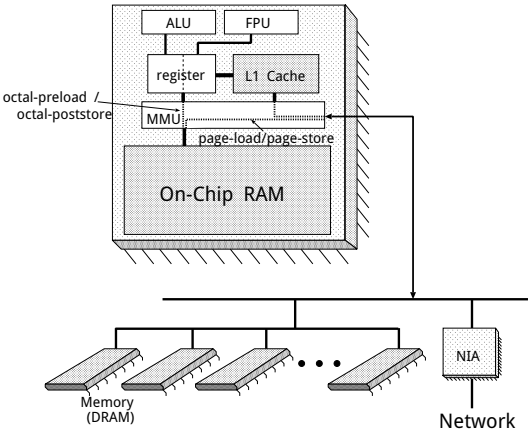
4.2.1 基本方式

アーキテクチャ　今回提案するプロセッサは、主記憶を外付け DRAM で実現し、中規模の高速 RAM をプロセッサ LSI に混載する方式とする。その基本アーキテクチャは以下のとおりである。

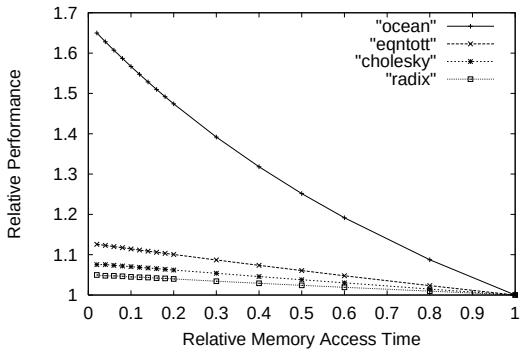
- **PU** 全体：プロセッサ・メモリ混載型チップ、 DRAM、 NIA（ネットワークインタフェース）からなる。
- 混載型チップ：スーパスカラアーキテクチャを基本とする。 64 ビット (場合によっては 128 ビット) 浮動小数点演算器を仮定し、 4 本のパイプラインを持つとする。
- 混載されるメモリ：数 MB から数 10MB 程度の高速 SRAM を想定し、多バンク化ないし多ポート化して高いスループットを確保する。

追加命令　メモリ系の制御のために以下の命令を追加する。

- `octal-preload`, `octal-poststore`: レジスタとオンチップメモリ間のデータ転送を行なう拡張命令。 512 ビットを単位として高いスループットを実現するとともに、 CP-PACS で提案・実装した疑似ベクタ処理機構を導入することで、前後の処理と重畳化し、メモリレーテンシを隠蔽する。
- `load-page`, `store-page`: オンチップメモリと外付け DRAM の間のデータ転送を行なう拡張命令。転送はページ単位で行ない、前後のページに対する計算と重畳化することで、レーテンシを隠蔽する。



(a) メモリ混載プロセッサのデータ転送



(b) メモリ混載による性能向上のシミュレーション評価

図 4: HPC 用メモリ混載プロセッサ

動作　大規模データを扱う科学計算におけるメモリ系の動作の概略は以下のとおりである。

1. 計算に用いるページを、外付け DRAM からオンチップメモリに転送する (load-page)
2. 計算に用いるデータを、オンチップメモリから、レジスタに octal-preload する
3. 計算を実行する
4. 計算結果のデータを、レジスタからオンチップメモリに octal-poststore する
5. 計算結果の入ったページを、オンチップメモリから外付け DRAM に転送する (store-page)

以上の操作のうち、前のページのロード (1) と今のページの一連の octal-preload(2) は重畳化される。同様に、今のページの一連の octal-poststore(2) と前のページのストアは重畳化される。これをページレベルのパイプラインと呼ぶ。さらに、 octal-preload, 計算実行、 octal-poststore は、 8 語を単位として重畳化される。これを、語単位のパイプラインと呼ぶ。以上の 2 階層のパイプラインによって、高い処理スループットを得ることができる。データ転送の様子を、図 4(a)に示す。本アーキテクチャでは、 (1) 外付け DRAM と混載メモリ間の高スループット転送、 (2) 混載メモリとレジスタの間の高スループット低遅延転送、によって浮動小数点演算器の稼働率を高く保ち、高い性能を実現することを目指す。

4.2.2　シミュレータ開発

平成 9 年度に引き続いて、シミュレータは、以下のように構築している。

- Stanford 大学で開発されたシミュレータ Sim OS および MXS を移植し、今回の研究目的である連続体系の計算科学向けのベンチマーク評価に用いた。
- 上記をふまえて、今回のアーキテクチャを評価するための、独自のシミュレータ開発を行なった。基本アーキテクチャを MIPS R10000 とし、これを前節で述べた基本方式にしたがって改良する形で開発した。

4.2.3 最適化コンパイラ

本年度より、プロセッサ・メモリ混載型アーキテクチャ用の最適化コンパイラの開発を開始した。`octal-preload`, `octal-poststore`, `load-page`, `store-page` という新たな追加命令を利用して、本アーキテクチャを最適に活用するコード生成を行うものである。

4.2.4 基本評価

平成 10 年度は、(1) Livermore Kernel などの基本ベンチマークを用いた解析、(2) シミュレーション評価、の 2 点から行った。結果を以下に記す。

基本ベンチマークの解析　仮定したハードウェアは以下のとおりである。

- 内部クロック：　2GHz
- I/O クロック：　1GHz
- 外付けメモリへのデータバス幅：　512 ビット (64B)
- 拡張命令：　`octal-preload`(512 ビット), `octal-poststore`, `load-page`, `store-page`
- 浮動小数点用パイプライン：　4 本　(multiply and add × 4)
- 最大性能：　16 GFLOPS/PU, 160 TFLOPS/(10000 PU system)

以上の仮定の下、以下の結果を得た。

- Livermore 1: 実効 8.2 GFLOPS/PU、82 TFLOPS/(10000 PU system)
- Livermore 21 (C version): 実効 10.6 GFOPS/PU、106 TFLOPS/(10000 PU system)

記憶系に新しい機構を導入することで、演算器の高い利用効率が得られ、実効で 100 TFLOPS を越えるシステムが 2004 年の LSI（プロダクト技術）によって構築可能であることが示された。現在、QCD などの実際の計算科学で用いられるプログラムの核となるループについての解析を進めている。

シミュレーション評価　シミュレーション評価では、SPLASH, SPEC といった標準的なベンチマークに対して、メモリ混載によるバンド幅の拡大による性能向上について調べた。結果の一部を図 4(b)に示す。この結果によると、データがすべて混載メモリに乗って、メモリアクセス時間が大幅に減る場合には、最大で 70% 近い性能向上が得られることがわかった（Ocean ベンチマークの場合）。現在、データがより大規模になった場合のシミュレータを作成中である。

4.3 SC98 出展

本研究推進の一環として、今年 11 月 7 日 ~ 13 日にかけ、米国フロリダ州オーランドにおいて開催された SC98 (高性能計算とネットワーキングのための国際会議) の研究展示部門への出典を行なった。同会議は、テクニカルセッションと企業展示及び研究展示を融合し

た、高性能計算関連の有名な会議であり、 11 年の歴史を持つ。毎年約 4000 人の出席者が参加する、この分野最大の国際会議として知られている。

本会議における出典では、約 20 フィート四方のブースにおいて、合計 14 枚のパネル及び実ハードウェアの展示及びプレゼンテーションを行ない、筑波大学を中心として約 20 年間続けられてきた PAX 研究の歴史に始まり、 CP-PACS のアーキテクチャ・ソフトウェア及び計算物理学における成果、そして本未来開拓研究の概要等を紹介した。 3 日間の出典期間に著名な並列処理システム研究者及び応用研究者を含む多数の来客があり、我々の活動の広報活動として、全体として大変有意義な出典であったと思われる。

5 今後の計画

並列入出力・可視化　平成 11 年度には、並列入出力システム PIO の実行時サポートシステムに、データ流の動的負荷分散機能を実装し、より柔軟で効率的な並列入出力環境の実現を目指す。また、並列ワークステーション上に、並列データ流を高速でディスクに格納するファイルシステムの構築や、超並列計算機から生成されたデータに対する後処理を機能分散的に同時並行処理するプログラムの開発等も課題である。これと並行して、ネットワーク・共有メモリ対応の AVS データ入力モジュールの機能を強化する。平成 12 年度以降は、これらの並列入出力・可視化システムを、宇宙物理学、素粒子物理学等の物理応用計算プログラムの実行に供するとともに、運用上の問題点の抽出・修正と性能チューニングを行ない、本システムの実用化を推進する。

プロセッサ・メモリ混載型 **LSI**　平成 11 年度には、機能レベルシミュレータによる詳細な性能評価を行い、これに基づいて、 Verilog/VHDL などの CAD を用いた LSI の回路設計を開始する。続いて平成 12 年度からは、回路設計の本格化とシミュレーションによる論理検証を行ない、 LSI の詳細設計を進め、最終的に設計したプロセッサ・メモリ混載型 LSI の詳細評価を実施する。

統合システム　計画最終年度には、並列入出力・可視化システムを持つ連続体向け超並列計算機と多粒子系超並列計算機を結合し、ヘテロジニアス・マルチコンピュータシステムの構築を目指す。また、本計画による技術開発の整理・評価を行い、次のステップの方向性を探るための会議を開催する。

6 学会発表、論文等

6.1 国際学会等における発表

[1] K. Kutoba, K. Itakura, M. Sato, T. Boku, "Practical Simulation of Large-Scale Parallel Programs and Its Performance Analysis of the NAS Parallel Bechmarks", Proceedings of Euro-Par'98 (LNCS 1470), pp.244-254, 1998.

[2] K. Kubota, M. Sato, K. Itakura, T. Boku, "Accuracy of fast performance prediction by instrumentation tool EXCIT", Proceedings of HPC Asia'98, pp.1031-1038, 1998.

[3] T. Boku, M. Mishima, K. Itakura, "VIPPES : A Virtual Parallel Processing System Simulation Environment", Proceedings of HPC Asia'98, pp.843-853, 1998.

[4] M. Matsubara, K. Itakura, T. Boku, "Large Scale Molecular Dynamics Simulations on CP-PACS", Proceedings of HPC Asia'98, pp.321-331, 1998.

[5] CP-PACS Collaboration, S. Aoki *et. al.*, "Full QCD light hadron spectrum from the CP-PACS", Proceedings of Lattice 98, to appear, 1998.

[6] CP-PACS Collaboration, S. Aoki *et. al.*, "The static quark potential in full QCD", Proceedings of Lattice 98, to appear, 1998.

[7] CP-PACS Collaboration, S. Aoki *et. al.*, "Heavy Quark Physics in $N_f = 2$ QCD", Proceedings of Lattice 98, to appear, 1998.

6.2 国内学会等における発表

[1] 松原正純, 沼寿隆, 朴泰祐, 中本泰史, 梅村雅之, 白川友紀, 宇川彰, "超並列計算機のための Commodity Network に基づく並列入出力・可視化システム", 電子情報通信学会コンピュータシステム技術報告, 1999 年 1 月.

[2] 近藤正章, 早川秀利, 坂井修一, 朴泰祐, "プロセッサ・メモリ混載型 LSI およびオンチップマルチプロセッサに関する検討", JSPP'98, pp.159, 1998.

[3] 近藤正章, 坂井修一, 朴泰祐, 中村宏, "オンチップメモリを用いた HPC プロセッサの検討", Hokke'99, 1999 年 3 月.

[4] 板倉 憲一, 朴泰祐, 久保田和人, 佐藤三久, "汎用相互結合網シミュレータ生成系による並列計算機の性能評価手法", 並列処理シンポジウム JSPP'98 論文集, pp.63–70, 1998.

[5] 松原正純, 板倉憲一, 朴泰祐, "超並列計算機 CP-PACS における大規模分子動力学法シミュレーション", 並列処理シンポジウム JSPP'98 論文集, pp.215-222, 1998.

[6] 久保田和人, 板倉憲一, 佐藤三久, 朴泰祐, "大規模データ並列プログラムの性能予測手法と NPB2.3 の性能評価", 並列処理シンポジウム JSPP'98 論文集, pp.71-78, 1998.

6.3 雑誌論文等

[1] 板倉憲一, 松原正純, 朴泰祐, 中村宏, 中澤喜三郎, "超並列計算機 CP-PACS における NPB Kernel CG の評価", 情報処理学会論文誌 39, No.6, pp. 1757–1765, 1998.