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	筑波大学機能工学系助教授	(IOV)
	和田耕一	筑波大学電子・情報工学系教授	(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	筑波大学機能工学系教授	(IOV)
	白川友紀	筑波大学機能工学系教授	(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。我々は、近年のコモディティ化したネットワーク媒体や接続技術を積極的に利用して、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する。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。

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。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 のアーキテクチャにおいては、演算器の高い演算性能と並び、それを有効に利用できるために、記憶装置から演算装置への充分なデータ供給性能と、それを実現するメモリ階層の構成が要求される。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。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と詳細設計を行なう。

3　研究計画の概要

3.1　並列入出力・並列可視化システム

本計画では、コモディティ技術により超並列計算機の並列入出力システムを構成することを目指している。その具体的実現システムとして、 100base-TX イーサネットと高速スイッチをネットワーク媒体とし、これらにより超並列計算機 CP-PACS と並列ディスクサーバ (Origin-2000) 及びグラフィックサーバ (Onyx2) を複数のチャネルにより並列接続する。

このシステムをユーザ・アプリケーションから利用する際には、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。本計画ではこのための API の開発実装を行う。

可視化システムについては、最終的なフレームバッファ描画処理等については専用ハードウェアによる充分な高速化が既に達成されているが、素データに対する画像処理に関しては、より一層の高速化が望まれている。従って、本計画では、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、これにより可能となる素データに対する可視化処理の並列化によって、高いスループット性能を持つシステムの実現を目指す。

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 1998 年版によれば、 2002 年には、マイクロプロセッサのクロック周波数 2.1GHz、トランジスタ数 76M、チップ I/O 数は 3000、電源電圧 1.2-1.5V、最大消費電力 130W となると予想されている。 CP-PACS で使われた技術に比べると、トランジスタ数で約 17 倍、クロック周波数で 14 倍であり、演算性能には大幅な向上が期待できる一方で、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が、半導体理工学研究センター (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。

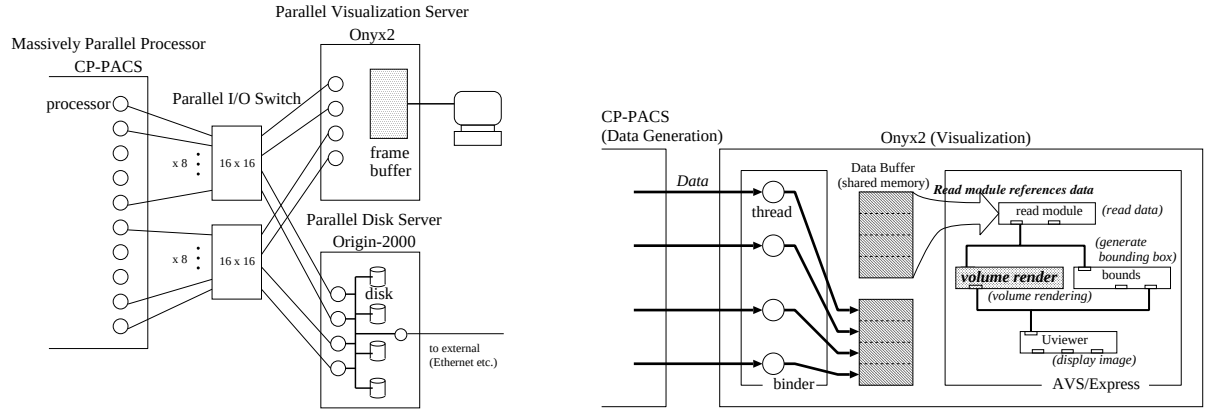
我々は、この点から従来のプロセッサアーキテクチャを考え直し、大容量の主記憶を外付け DRAM で実現する一方、内部に中規模の容量の SRAM を持ち、これをユーザプログラムから制御可能なメモリとして用いる、新たなプロセッサアーキテクチャを提案する。本計画では、この新たなアーキテクチャの仕様の詳細を定め、シミュレータ及びコンパイラの開発を行って、 Linpack 及び物理応用計算プログラムを用いた性能評価を実行した後、詳細設計を行って、大規模科学技術計算における新アーキテクチャの有用性の実証を目指す。

4 平成 11 年度の成果

4.1 並列入出力・可視化システム

4.1.1 並列入出力システム

今年度は実用システムの構築に向けて、 CP-PACS の 128 台の入出力プロセッサ (IOU: Input/Output Unit) のうち 16 台に対し、 Fast (100base-TX) Ethernet インタフェースを実



(a) PIO システムの接続図　　(b) 並列可視化システムのデータ処理の流れ

図 1: 並列入出力・可視化システム

装し、これらを 2 台の Fast Ethernet Switching Hub を経由して、並列ファイルサーバ Origin-2000 及び並列ビジュアリゼーションサーバ Onyx2 に接続した。 Origin-2000 及び Onyx2 には、各々 4 ポートの Fast Ethernet ポートが備わっている (図 1(a)参照)。このハードウェアの上に、前年度より開発を進めている並列入出力システムである PIO (Parallel Input/Output) システムを実装した。

超並列計算機実機に対する **PIO** システムの適用と性能評価　PIO システムは各並列計算機上で動作する PIO server と呼ばれるサーバプログラム群及び、それらとクライアント (ユーザアプリケーション) プログラムを結合する API (Application Program Interface) から成る。前年度までのシステム開発は、主に CP-PACS の小型機である PILOT-3 (128 PU + 16 IOU) 上で行なってきたが、今年度は CP-PACS に多数の並列ネットワークインタフェースを設置したことにより、 2048 PU + 128 IOU を用いた大規模な実験が可能となり、これらを用いて実アプリケーションに対する PIO の性能評価を行なった。

実験の結果、 CP-PACS の 128 台の IOU 全てで PIO server 群を実行させた場合、 2048 PU 全てを用いたユーザアプリケーションからの出力サービスは余裕をもって処理できることがわかった。これは、 PIO server 群を成す多数のプロセスが十分な容量を持つバッファ機能を提供するため、ユーザアプリケーションの出力にストレスを与えず、効率的なパイプライン処理が実現できているためである。また、バッファリング処理のオーバヘッドを加えても、 PIO server は裸の TCP/IP 通信に対し、約 80% の性能を達成することがわかった。従って、 PIO システムを使用することにより、例えば 16 台の IOU に設置された Fast Ethernet インタフェース群により、最大で TCP/IP 1 本分に比べ約 13 倍の通信性能が達成可能である。

動的負荷分散機能の実装と評価　PIO システムの持つ動的負荷分散機能に対する性能評価を、 CP-PACS の 256 PU を用いて行なった。 PIO システムは、米国の HPSS のような従

来の大規模並列入出力システムと異なり、静的なファイル転送だけでなく、ユーザアプリケーションからの直接的な入出力を並列処理することを目的としている。このため、静的なファイルの入出力とは異なり、事前に負荷バランスを予測することが難しいため、並列ネットワークを効率的に使用するには動的負荷分散機能が不可欠である。分散メモリ型並列計算機で、情報の一局集中を避けつつ動的な入出力負荷分散を実現するのは困難であるが、 PIO システムでは外部に接続された共有メモリワークステーションを負荷バランス情報サーバとして用いることでこれを解決している。

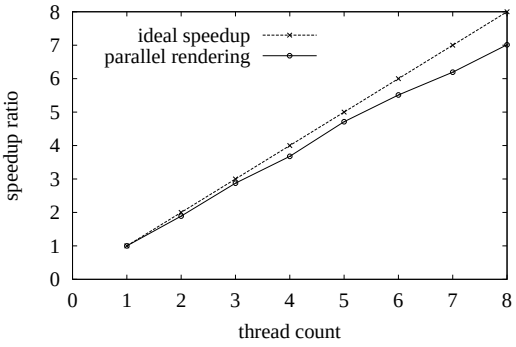
この機能を用いて、各並列プロセスの出力データ量が動的に変化するテストプログラムを CP-PACS 上で実行し、 PIO サーバによって Origin-2000 に並列出力する実験を行なった。実験では送出パケットの平均サイズを 64KB から 2048KB のいずれかに固定し、平均サイズに対し最大で 2.1 倍までパケットサイズを増加させる (ただし総データ量は一定) ように出力を調整した。結果として、動的負荷分散を行なった場合、静的負荷分散 (プロセスの位置に基づく均等割付) に対し、 1.40 倍から 2.08 倍までの性能が得られた。平均パケットサイズを大きくするほど動的負荷分散の効果が顕著になり、最終的に理論的な上限値とほぼ同じ値が得られた。

4.1.2 並列可視化システム

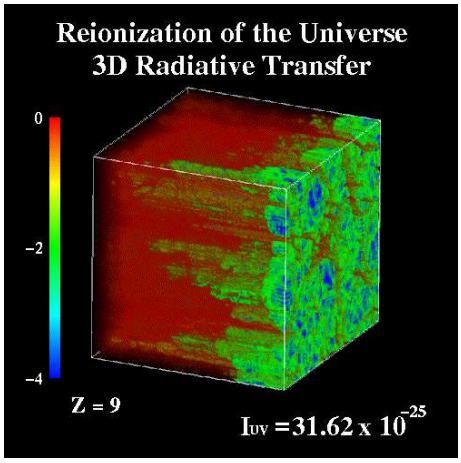
本年度は並列ビジュアリゼーション用ワークステーションである Onyx2 上の並列プロセッサ数を 4 台に増強し、より高いスループットで可視化を行なえるようになった。 Onyx2 上ではコモディティ化されている可視化ソフトウェアである AVS/Express が稼働している。昨年度は、 PIO システムと AVS/Express を結合するための並列入力モジュールを、 AVS の 1 モジュールとして実装し、データ入力と AVS による可視化処理を機能的に並列化するシステムを実現したが、本年度はさらに AVS 側の処理を高速化すべく、ボリュームレンダリング処理の並列化を行ない、さらに超並列計算機と並列可視化システムを統合的に利用する環境を構築した。そして、これら全ての機能を AVS/Express の GUI 環境から簡単に使用可能にした。

並列ボリュームレンダリングモジュール　図 1(b)に、並列ビジュアリゼーションサーバ Onyx2 における PIO システムからのデータ入力から最終画像の描画までの流れを示す。 binder は PIO から供給される並列データストリームをマルチスレッド化された並列プロセスによって処理し、 AVS が通常に処理できるデータに変換する。このデータは AVS に組み込まれた任意のモジュールに流され、適宜処理される。 AVS の各モジュールは基本的にシステムによって予め用意されているが、特定のモジュールを任意のユーザモジュールに差し替えることも可能である。我々はこの機能を用い、 3 次元ボリュームレンダリング (以下 VR と省略) を実行する AVS モジュールに対する高速化を行なった。

AVS は基本的に全体を逐次処理することしかできず、特に 3 次元 VR のような重い処理が伴なうとその性能は一気に低下する。本システムでは、 AVS の行なっている VR 処理を解析し、それと等価な働きをする、マルチスレッドによって並列化された高速モジュールを



(a) 並列ボリュームレンダリングモジュールにおける速度向上率

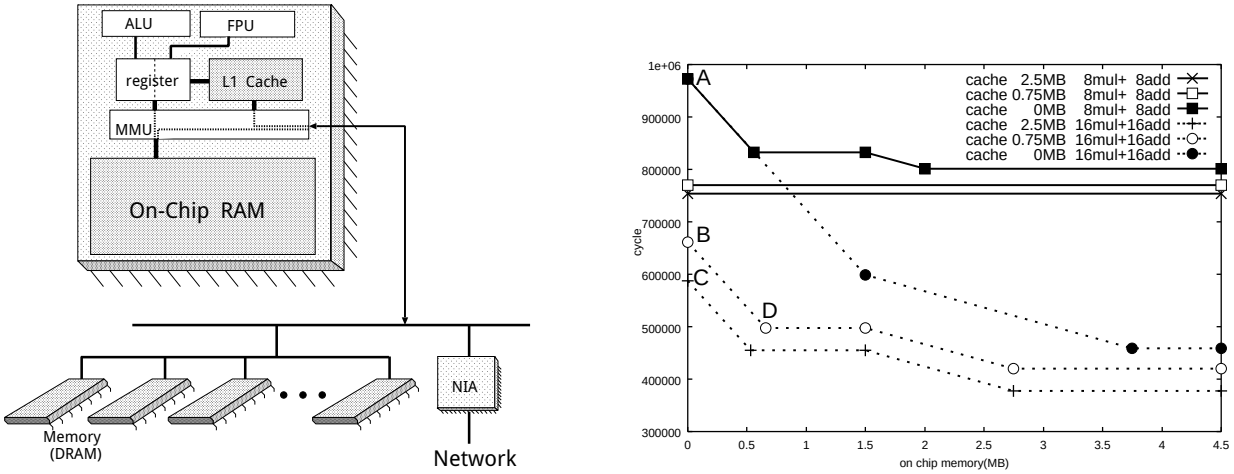


(b) 並列ボリュームレンダリングを用いた出力例

図 2: 並列可視化システムの性能評価

開発した (図 1(b)中の *volume render*)。このモジュールは、3 次元 VR における処理をほぼ完全に並列化しており、使用可能プロセッサ数にほぼ比例した非常に効率の高い並列処理を実現している。従来のソフトウェアによる並列 VR の研究では、主として静止画像に対する高速 VR が研究されてきた。ここでは、医学等の応用分野において、一旦取り込んだ画像をいろいろなパラメータの下で高速描画することが目的になっている。しかし、我々が目指すのは動的に流入してくるデータ流のスナップショットに対する可視化であり、従来のデータ蓄積型・中間表現型のアルゴリズムが適用できない。その代わりに、3 次元データに対する補完作業を効率良く処理し、全体の負荷バランスを重視する実装を行なった。図 2(a)に、最大 8 スレッドを用いた場合の、VR 処理の速度向上率を示す (台数効果を見るため、8 プロセッサを持つ Origin-2000 上で AVS を稼働させた) 。このような高い並列処理効率により、例えば後述する計算宇宙物理における 128^3 の空間データを 512^2 の 2 次元データに変換する VR 処理において、約 1 秒 / フレームの実時間描画が可能となった。

並列入出力・可視化システムの統合制御　PIO 及び並列可視化システムを実並列計算機と共に快適に使用する環境を提供するため、超並列計算機 CP-PACS をはじめ、関連する計算機上で各種プロセスを起動し、全体のデータ流を構築する作業を全てスクリプトとして事前に記述し、AVS の操作コンソールから簡単なメニュー選択によって自動的に起動できるようなシステムを構築した。また、ビジュアリゼーションサーバの CPU 数及び性能に限りがある場合、binder 等の他の並列プロセスとの負荷バランスが問題となる。これを簡単に調整できるよう、AVS 上の並列 VR 制御画面では、使用するプロセッサ数 (立ち上げるスレッド数) を実時間的に変更可能なスライドバー等が用意されており、実際に CP-PACS で行なわれている計算負荷・ PIO におけるデータ転送負荷・並列可視化処理の負荷等を総合的に判断して、可視化処理時間の最適化が行なえるようになっている。



(a) メモリ混載プロセッサの構成図　　(b) QCD 計算における性能評価

図 3: メモリ混載プロセッサ

4.1.3 CP-PACS における実応用アプリケーション

これまで述べた PIO システムと柔軟な制御性を持つ並列可視化システムを総合的に使用した例として、 2048 台の全 PU を使用した CP-PACS 実機において、 128 台の IOU 上での PIO サーバの稼働、 16 チャネルの Fast Ethernet を経由した並列データ転送、ビジュアリゼーションサーバにおける並列データ入力及び並列可視化の全処理を、実際のアプリケーションを対象に実行した。対象アプリケーションは計算宇宙物理学における宇宙の再イオン化シミュレーション及び、大量のアルゴン分子に対する分子動力学法シミュレーションである。いずれも 2048 台の PU から生成されるデータストリームを、 16 本の Fast Ethernet リンクで負荷分散しながら転送し、最終的に 1 秒 / フレーム程度の 3 次元 VR 及び描画を実現できた。これにより、超並列計算機における計算結果の実時間可視化が可能であることが確認された。宇宙の再イオン化シミュレーションを実行している際の並列可視化システムの出力画像を図 2(b)に示す。この例では、再イオン化の進行の様子が確認できるように、輻射が一方向のみから入射した場合を扱っている。これは、全ての方向からの入射を取り扱う現実的シミュレーションの第一歩であると同時に、素過程の解析に役立っている。

4.2 プロセッサ・メモリ混載型 LSI

4.2.1 基本アーキテクチャ

概要　提案するメモリ混載型プロセッサの構成図を図 3(a)に示す。シングルチップ内にロジック部および 1 次キャッシュだけでなく、アドレス指定可能な高速なオンチップメモリも搭載する。一般の科学技術計算における大容量のデータセットに対応するため、チップ外にもメインメモリとして DRAM を配置する。

オンチップメモリと従来のキャッシュとの相違点は、データアロケーションとリプレース

メントの制御がキャッシュでは自動的に行なわれるのに対し、オンチップメモリではその制御をソフトウェアで明示的に行える点である。この点から、提案するアーキテクチャを SCIMA(Software Controlled Integrated Memory Architecture) と呼ぶ。ハードウェア制御のキャッシュでは、ユーザが意図しないデータアロケーション、リプレースメントによるキャッシュミスが発生することによる性能低下が問題となる。オンチップメモリを採用する狙いは、多くの科学技術計算がデータアクセスに規則性を持つことを利用し、ユーザ (あるいはコンパイラ) が明示的にデータアロケーションとリプレースメントを行なうことでこの問題を回避することである。

メモリ階層　SCIMA では、従来のアーキテクチャに対して、メモリ階層に関して以下の拡張を行う。まず、浮動小数点レジスタを増やす。これは、集積度の向上による搭載可能な演算器の増加とチップ内のキャッシュ / オンチップメモリの高バンド幅を活用するためである。オンチップメモリ上でのデータアロケーションとリプレースメントの制御をソフトウェアに解放するため、オンチップメモリは論理アドレス空間上に定義する。但しオンチップメモリ領域は uncacheable とするので、キャッシュとの包含関係は生じない。

追加命令　上記のメモリ階層の拡張を有効に利用するため以下の命令を追加する。

- `load-multiple, store-multiple`: オンチップメモリの連続アドレス上と、連続するレジスタ間で複数データの転送を行う命令。オンチップメモリとレジスタの間に 1000 bit 程度のデータパスを設けることは将来的には可能と考えられるため、この命令の転送サイズとしては 2 , 4 , 8 , 16 double-word 程度を想定している。
- `page-load, page-store`: オフチップメモリとオンチップメモリ間でデータ転送を行なう命令。データ転送はページ単位で行う。この転送とチップ内部の計算処理とを重畳化することで、オフチップメモリアクセスのレーテンシを隠蔽する。また、ブロック幅とストライド幅を指定するブロックストライド転送の機能を付け加える。この機能により、オフチップメモリ上の不連続領域のデータをパッキングして、オンチップメモリに持ってくることを可能にする。

4.2.2　シミュレータ作成

提案する SCIMA は既存アーキテクチャの拡張として定義可能であるが、本研究では具体的には MIPS アーキテクチャをそのベースアーキテクチャとして採用する。本年度は、 SCIMA の命令レベルシミュレータとクロックレベルシミュレータを作成した。前者は与えられた命令列を実行し結果が正しいかを確認するためのものであり、後者は与えられた命令列を実行するのに要するクロック数を求め性能を測定するためのものである。

4.2.3　最適化コンパイラ

SCIMA の内蔵 SRAM は、ソフトウェアで制御するため、これを活かすには、コンパイラによるコード最適化が必須である。本年度は、 (1)SCIMA 向けの言語仕様の拡張、 (2) フ

ロントエンドにおける最適化、 (3) load-multiple、 store-multiple, page-load, page-store という追加命令の生成手順、などを検討した。なお、ここでは言語として C を用いている。

4.2.4 性能評価

QCD 計算の解析　実アプリケーションである QCD(量子色力学) 計算を解析し、提案するアーキテクチャの有効性を検討した。この評価では、オンチップメモリとキャッシュのスループットを 16 double-word/cycle、オフチップメモリのスループットを 4 double-word/cycle と仮定し、また演算能力として (8mul+8add)/cycle と (16mul+16add)/cycle の 2 つの場合を想定し、計算に必要となる演算回数とメモリアクセス回数から性能を評価した。その結果を図 3(b)に示す。図 3(b)においては、横軸はオンチップメモリ容量を示し、縦軸は所要サイクルを示す。実線は演算能力が (8mul+8add)/cycle の場合を、点線は演算能力が (16mul + 16add)/cycle の場合を示す。それぞれグラフが 3 本ずつあるのは、パラメータとしてキャッシュサイズを 3 通りに変化させたためである。

実線と点線を比較することで、演算能力が (8mul+8add)/cycle の場合は演算能力が不足していて、キャッシュやオンチップメモリを増加させても性能が向上しないことがわかる。次に、このグラフの A 点と B 点の比較より、キャッシュを 0MB から 0.75MB に増加することは性能向上に大きく貢献すること、 B 点と C 点の比較より、しかしキャッシュを 0.75 MB から 2.5MB に増加してもそれほどは性能向上に効かないことがわかる。また、 C 点 (キャッシュ 2.5MB, オンチップメモリ 0MB) と D 点 (キャッシュ 0.75MB, オンチップメモリ 0.66 MB) とを比較することで、 0.75MB のキャッシュがある場合には、キャッシュを増加するよりもオンチップメモリを増加した方が効果があることがわかる。

シミュレータを用いた行列積の評価　行列積の評価を、本年度開発したシミュレータを用いて行った。結果を表 1に示す。評価では、キャッシュ 32KB のみを持つ構成 (表中では CACHE) とオンチップメモリ 32KB のみを持つ構成 (表中では SCIMA) の性能を、行列サイズを変えながら求めた。それぞれの構成において、データアクセスの局所性を向上するブロッキング手法を取り入れて最適化を行っている。現在の実装技術の元でのオンチップメモリの効果を調べるため、演算能力は (1mul+1add)/cycle、キャッシュとオンチップメモリのスループットは 1 double-word/cycle でレーテンシは 1 サイクル、オフチップメモリのスループットは 0.5 double-word/cycle でレーテンシは 10 サイクルとした。クロックは 180MHz である。

表 1: 行列積の性能 (オフチップメモリレーテンシ : 10 サイクル)

行列サイズ	32×32	64×64	96×96	128×128	160×160	192×192	224×224
CACHE [MFLOPS]	192.3	184.9	210.8	191.4	213.1	211.4	213.5
SCIMA [MFLOPS]	243.9	265.8	273.1	280.4	285.9	288.7	291.2
性能向上率 (SCIMA/CACHE)	1.27	1.44	1.30	1.46	1.34	1.37	1.36

この結果より、提案する SCIMA は従来のキャッシュのみを持つ場合に対して 3 割から 4 割程度性能が良いことがわかる。また、上記仮定では理想的なピーク速度は 360MFLOPS であるので、 SCIMA は行列サイズが大きくなるとその 8 割程度の実効性能を達成していることがわかる。また、オフチップメモリのレーテンシを 60 サイクルにした場合、 SCIMA の性能はほとんど変わらないが、キャッシュのみをもつ構成では性能低下が著しく、 SCIMA はキャッシュのみの構成に対して約 2.9 倍の速度向上を達成できることもわかった。

5 今後の計画

並列入出力・可視化 並列入出力システムに関しては、現在の動的負荷分散機能が持つオーバヘッドをさらに低減させ、 PIO システムの機能をより充実させていく。現在は負荷分散機能を並列ワークステーション中心に実装しているが、その分の負荷を減らすためのチューニングを行なう。また、実アプリケーションでの PIO システムのさらなる応用を重ね、不足している機能や性能的に満足できない点等をより充実させる。特に、 C や Fortran 等の一般的な言語において、従来のファイル入出力関数と同等な機能を持つ並列ファイルサービス環境を整備し、 CP-PACS のように並列ファイルシステムを内部に持つような超並列計算機において、ユーザにファイルの分割化・並列アクセス化を意識させず、さらにファイルシステムが計算機上にあるのか計算機外にあるのかも区別しなくてすむような環境を実現してゆく。

並列可視化システムについては、 AVS コンソールからのより柔軟な全体システム制御を実現する。現在はジョブの起動等のワンタイム性のある処理のみが自動化されているが、今後は実時間で処理されている可視化の流れを一時的に記録し、記録容量の許す任意の範囲での画像のロールバックや、その指示を計算サーバである超並列計算機までフィードバックできるような総合的な処理環境の構築を目標とし、これらを平成 12 年度中に実現する。最終年度には各種実アプリケーションでの運用実績を作り、システムの総合評価を行なう予定である。

プロセッサ・メモリ混載型 LSI 平成 12 年度には、クロックレベルシミュレータの改良を行い、外付けメモリをインターリーブ構成にした場合の評価ができるようにし、またシミュレーション速度を向上することでより大規模な科学技術計算も評価できるようにする。その上で、素粒子物理学や宇宙物理学等における実アプリケーションの性能評価を行う。また、提案するアーキテクチャの命令パイプライン構成を決定し、その機能レベル設計を Verilog / VHDL 等のハードウェア記述言語で行う。その後、機能レベルのシミュレーションによる性能評価を行うと共に、自動合成ツール等を用いて論理設計も行い、各パイプラインステージのクリティカルパスに関する検討を加える。また、適切な言語インタフェースを設定し、コンパイラのフロントエンドを設計する。

続いて平成 13 年度には、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。同時にコンパイラのバックエンドを製作し、ソースプログラムをコンパイルして SCIMA

アーキテクチャにマップする技術を確立するとともに、得られたオブジェクトコードをじっさいにシミュレータにかけ、総合的な評価・検証を行う。

以上のプロセッサ・アーキテクチャの研究に並び、次世代超並列計算機のもう一つの重要な検討要素であるプロセッサ結合網の検討を平成 12 年度から 13 年度にかけて行い、平成 13 年度には、連続体向けの次世代超並列計算機の全体構想を纏める。

統合システム　以上の研究と平行し、平成 12 年度から連続体向け超並列計算機と多粒子系向け超並列計算機の結合方式の検討を行う。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] M. Matsubara, H. Numa, and T. Boku, "Commodity Network based Parallel I/O System for Massively Parallel Processors", Proceedings of PDPTA'99, pp.2424-2429, Las Vegas, June 1999.

[2] Y. Iwasaki, "CP-PACS Project and the Future", Science from Tsukuba 999, Tsukuba, November 1999.

[3] A. Ukawa "Aspects of next generation parallel computers", Japan-German Seminar '99 "Lattice Field Theory on TFLOPS Supercomputers" Kanazawa, October 1999.

6.2　国内学会等における発表

[1] 松原正純, 沼寿隆, 板倉憲一, 朴泰祐, "コモディティネットワークに基づく並列入出力システム", 情報処理学会ハイパフォーマンスコンピューティング研究会, 99-HPC-76, 1999, pp.1–6.

[2] 松原正純, 沼寿隆, 板倉憲一, 朴泰祐, "超並列計算機のための並列入出力・可視化システム", 並列処理シンポジウム JSPP'99, p.210, 1999.

[3] 沼寿隆, 松原正純, 板倉憲一, 朴泰祐, "並列入出力機構を用いた可視化システムの提案", 情報処理学会ハイパフォーマンスコンピューティング研究会 (SWoPP'99), 99-HPC-77, 1999, pp.53–58.

[4] 近藤正章, 坂井修一, 朴泰祐, 中村宏, "HPC 向けプロセッサのメモリ・アーキテクチャの基本構成", 情報処理学会計算機アーキテクチャ研究会 (SWoPP'99), 99-ARC-134, 1999, pp.1–6.

[5] 服部直也, 荒木拓也, 坂井修一, 田中英彦, “実行サイクル数予測に基づく大域的命令スケジューリングの実装と評価”, 情報処理学会 ARCH133-3 (1999).

[6] 飯塚大介, 小沢年弘, 坂井修一, 田中英彦, “C コンパイラにおけるループ最適化手法”, 情報処理学会ハイパフォーマンスコンピュータ研究会 (鞆海峡メッセ下関), 99-HPC-77, Vol.99, No.66, pp.65–70 (Aug. 1999).

[7] 坂井修一, “新世代 LSI におけるコンパイラ技術の動向”, 電子情報通信学会第 3 回システム LSI 琵琶湖ワークショップ 招待講演, 1999.

[8] 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”，情報処理学会計算機アーキテクチャ研究会 (HOKKE'99)， ARC-132-15 (HPC-75-15)， 1999

[9] 大河原英喜, 中村宏, 吉江友照, 金谷和至, ”ハイパフォーマンスコンピューティングに適したメモリ階層の検討”, 情報処理学会計算機アーキテクチャ研究会， ARC-133-10， 1999

[10] 大河原英喜, 中村宏, 吉江友照, 金谷和至, ”ハイパフォーマンスコンピューティングに適したメモリ階層の初期評価”, 並列処理シンポジウム '99， pp.214,1999

6.3 雑誌論文等

[1] 三島健, 朴泰祐, 中村宏, 中澤喜三郎, “超並列計算機用多段結合網における転送性能の解析”, 情報処理学会論文誌 Vol.40, No.5, pp.1985–1995, 1999.

[2] 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”, 情報処理学会論文誌 Vol.40, No. 5, pp. 2172–2182, 1999.

[3] 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”, 情報処理学会論文誌 Vol.40, No. 5, pp. 2293–2304, 1999.

[4] 岡本一晃, 松岡浩司, 廣野英雄, 横田隆史, 佐藤三久, 坂井修一, “超並列計算機のための同期処理機構とその評価”, 情報処理学会論文誌 Vol.40, No.3, 1999, pp.1245–1456.

[5] P. R. Panda, H. Nakamura, N. D. Dutt, and A. Nicolau, “Augmenting Loop Tiling with Data Alignment for Improved Cache Performance”, IEEE Transactions on Computers, Vol.48, No.2, 1999, pp.142–149.

[6] T. Manke, H. P. Shanahan, A. A. Khan, S. Aoki, R. Burkhalter, S. Ejiri, M. Fukugita, S. Hashimoto, N. Ishizuka, Y. Iwasaki, K. Kanaya, T. Kaneko, Y. Kuramashi, K. Nagai, M. Okawa, A. Ukawa, T. Yoshie, “Hybrid Quarkonia on Asymmetric Lattices”, Phys. Rev.Lett. Vol. 82, 1999, pp.4396-4399.