

資 料

ペタフロップスマシン技術に関する調査研究

平成9年3月

財団法人日本情報処理開発協会
先端情報技術研究所

KEIRIN



この事業は、競輪の補助金を受けて実施したものです。



ペタフロップスマシン技術に関する調査研究

目 次

第1章	まえがき	3
第2章	米国のペタフロップスマシン技術開発状況	6
第3章	技術的展望と課題	35
3.1	アーキテクチャ	35
3.1.1	結合網の技術からペタフロップスを考える	35
3.1.2	ペタフロップスマシン実現に立ちはだかるもの	43
3.1.3	ペタフロップスマシンの展望	47
3.1.4	メモリ・アーキテクチャの展望と課題	54
3.2	ソフトウェア	61
3.2.1	並列システムソフトウェアの課題	61
3.2.2	並列化コンパイラから見たペタフロップスコンピューティング	65
3.2.3	ペタフロップスマシンにおけるプログラミングインターフェース	71
3.3	アプリケーションとアルゴリズム	78
3.3.1	ペタフロップスコンピューティングにおけるアプリケーション技術	78
3.3.2	ペタフロップスマシンとその応用	83
3.3.3	PetaFlops計算機の適用分野とアプリケーションからの要望	91
3.3.4	並列数値シミュレーションとペタフロップス計算機	101
3.4	デバイステクノロジー	107
第4章	あとがき	117
付属資料		
付属資料1	ペタフロップスマシンWG配付資料概略	120
付属資料2	Frontier'96出張報告書	130
付属資料3	Frontier'96参加報告書	139

ペタフロップスマシン技術調査ワーキンググループ

- 主査 山口 喜教 通産省工業技術院 電子技術総合研究所
情報アーキテクチャ部 計算機方式研究室長
- 委員 天野 英晴 慶應義塾大学理工学部 情報工学科 助教授
- 委員 石川 裕 技術研究組合 新情報処理開発機構 つくば研究センタ
超並列ソフトウェア研究室長
- 委員 稲上 泰弘 (株) 日立製作所中央研究所 プロセッサシステム研究部 主任研究員
- 委員 笠原 博徳 早稲田大学理工学部 電気電子情報工学科 助教授
- 委員 久門 耕一 (株) 富士通研究所 マルチメディア研究所
コンピュータシステム研究部 主任研究員
- 委員 妹尾 義樹 日本電気(株) C&C研究所 主任
- 委員 関口 智嗣 通産省工業技術院 電子技術総合研究所 情報アーキテクチャ部
主任研究官
- 委員 中島 浩 京都大学工学研究科 情報工学専攻 助教授
- 委員 中島 克人 三菱電機(株) 情報技術総合研究所 アーキテクチャ部
知識処理チームリーダー
- 委員 福井 義成 (株) 東芝 情報システム部 主査
- 委員 横川三津夫 動力炉・核燃料開発事業団 技術協力部 情報システム室
客員研究員
- 幹事 河西 和美 (財) 日本情報処理開発協会 先端情報技術研究所 技術調査部
主任研究員
- 幹事 高張 邦夫 (財) 日本情報処理開発協会 先端情報技術研究所 技術調査部
主任研究員

第1章 まえがき

計算機に対する処理能力の向上に対する要求は際限がないように思える。大規模な科学技術計算や材料分野でのデバイスシミュレーション、分子設計やDNA解析などに必要な分子・原子の科学的な振る舞いのシミュレーション、大規模な探索問題などスーパーコンピュータの能力を必要としている多くの需要が存在している。科学技術計算の分野では、実際に多くの研究開発でスーパーコンピュータが利用されているが、シミュレーションの精度の向上やさらに多くの状況でシミュレーションを行いたいという要求が常にある。このようなことから、ギガフロップス、テラフロップスといったスーパーコンピュータを実現するための技術開発が行われてきたわけである。スーパーコンピュータの研究開発は常に多くの技術的問題へのチャレンジを伴うが、それが成功するとそこで開発された技術は、商用機のハードウェアやソフトウェアのコスト・パフォーマンスの向上や関連する半導体技術など、多方面にわたる波及効果も生み出してきた。ただし、最近では、マイクロプロセッサに関する技術開発の方に、最先端技術の牽引車としての役割が移ってきていることは否めない。

現在、スーパーコンピュータの最大処理能力は、ギガフロップス (10^9) からテラフロップス (10^{12}) を伺うまでになってきており、テラフロップスマシンの実現は、時間の問題であろう。このような状況において、スーパーコンピュータにおける次の目標として、「ペタフロップスマシン」が掲げられるようになってきた。ペタフロップスとは、テラフロップスのさらに1000倍であり、10の15乗の浮動小数点演算が1秒間に行える計算スピードに相当する。このような、超高速の計算機を実現し、しかも有効に活用するためには、さまざまな問題を解決する必要があると考えられる。また、そもそもペタフロップスという膨大な計算能力が必要であるのか、ハードウェアあるいはアーキテクチャ上のネックは何か、ソフトウェアやアプリケーションはどうするのかといったさまざまな疑問がすぐに提起される。

米国では、数年前までは、テラフロップスの計算機の開発とその有効活用を目指した、「HPCC」プロジェクトを、国をあげて推進してきたわけであるが、テラフロップスの実現にはほぼ見通しがついた現時点で、さらにその1000倍の能力を持つペタフロップスの計算機に向けた議論がすでに始まっている。具体的には、1994年にPasadenaでCaltech (California Institute of Technology) 主催のペタフロップスコンピュータに関するワークショップが開かれている。この会議は、招待者のみのクローズな会議であったが、この会議録が公表されペタフロップスコンピュータに関する認識が一気に加速された。そして、1995年には、Petaflops Frontiers '95 が、さらに1996年には Petaflops Architecture、Petasoft'96、Petaflops Frontiers '96 などが次々と開催され、また Supercomputing '96 においても、Petaflopsに関するパネル討論が行われるなど、ペタフロップスに関する米国での活動は、年を追って活発になってきている。

わが国においては、まだ、このようなペタフロップスマシンを目指す組織的な動きはなく、

また、このようなテーマはメーカーが独力で研究を開始するには無理のある長期的な研究テーマであると考えられる。したがって、ペタフロップスマシンの実現性やそのアーキテクチャ研究の方向付けや、ペタフロップスマシンを想定したソフトウェアに関する研究動向について調査し、独自に検討を加えておくことは、極めて重要である。このような認識から、先端情報技術研究所（AITEC）内に「ペタフロップスマシン技術調査ワーキンググループ」が設置され、調査や議論を行うことになった。平成8年度における調査としては、わが国の、情報処理産業におけるステータスを考慮して見たときに、どのようなスタンスあるいは何に重点をおいて技術開発を行っていくべきかについて、何らかの指針を導くための準備段階であると考えている。

本報告書は、この「ペタフロップスマシン技術調査ワーキンググループ」での委員の方々による議論を基に、各委員の「ペタフロップスマシン」に関する意見をまとめたものである。本ワーキンググループ（WG）は、具体的な調査を行うために、大学、メーカー等の若手研究者、11名から構成されている。各委員には、アーキテクチャ、ソフトウェア、アプリケーションの各分野において実際に研究や開発に携わっている専門家として、活発な議論を行ってもらった。また、AITECを中心にして、海外特に米国におけるペタフロップスマシンに関する研究状況や会議について調査を行った。本ワーキンググループの会合は、平成8年10月から平成9年2月まで、月1回のペースで合計5回行われた。このWGにおいては、基本的に各委員の各自のスタンスからペタフロップスマシンに関する自由な問題提起をしてもらい、これを題材にして議論を行った。また、海外調査として、先に述べた米国でのペタフロップスに関する活動のうち、Petaflops Frontiers'96およびSupercomputing'96に参加し、それらの会議におけるPetaflops関連の発表や議論などに関して、参加した委員からの報告を行った。

本報告書では、本委員会で行われた討論をふまえて、各委員に自由にペタフロップスマシンに関する、イメージや何をこれから調査すべきであるかなどについて記述して頂いた。基本的に、どのような観点から各委員の意見を記述してもらうかについては、解説記事にみられるような安全・中立的なものではなく、将来に向けての各委員の信じる技術論・方法論に関する積極的記述を書いていただくように要請した。そのために、各報告にはそれを報告した各委員の記名を行うこととした。また、記述にあたっては特に次のような点を重視してもらうように要請した。

- (1) 日米における技術的現状の比較とわが国での問題点
- (2) ペタフロップスに向けた将来予想と解決策に関して、特にその要素技術、ニーズ、パラダイムシフトなどに関する意見
- (3) わが国として取り組むべき課題

(4) さらにどのような技術調査をすべきか

このうち、(1)に関しては、第2章において、米国におけるペタフロップス・マシン開発状況について、AITECによる調査内容を掲載してあるので、参考にしてほしい。また、本報告書の記述にあたっては、各委員の意見の調整などを行わなかったため、委員ごとに意見の違う点も生じているはずであるが、これらについては、むしろ今後の議論の種となるものであらうと考えている。

また、本委員会でのヒアリング講師として、東芝マイクロエレクトロニクス研究所(ME研) 所長である海野陽一氏による講演をお願いしたが、この講演をもとに、「デバイステクノロジー」に関する将来予測をAITEC事務局でまとめ、本報告書の1つの章として掲載してある。

(山口喜教主査)

第2章 米国のペタフロップスマシン開発状況

2.1 まえがき

米国においては、インターネットによる政府機関の情報公開が進んでおり、政府機関関連の詳細な研究開発情報がWebにて入手可能となっている。ペタフロップスマシン開発状況に関しても、NASA（National Aeronautics and Space Administration）本部に設置されたホームページから多くの情報を得ることができる。そこでは、ペタフロップスイニシアティブと呼ばれる活動によりWebにホームページが開設され、コミュニティの拡大、活動の活発化を目指していることが宣言されている。この研究開発活動は、HPCC（High Performance Computing and Communications）プログラムの関連研究開発としてNCO（National Coordination Office）のホームページと互いにリンクをめぐらしている。

本章では、上記ホームページをはじめ、Webで得られる情報を中心に集積し、整理した結果を示す。ペタフロップスイニシアティブについて記述した後、彼らが開催した各種ミーティングやこれまでの研究成果について報告する。

次に、ペタフロップスマシン開発の一手手前の、現在、開発にしのぎが削られているテラフロップスマシン開発のうち、米国のASCI（Accelerated Strategic Computing Initiative）プログラムとCSCC（Concurrent Supercomputing Consortium）について報告する。また、（サブ）テラフロップスマシンの実例として、日米各社の並列ベクトル型スーパーコンピュータ、超並列型スーパーコンピュータを紹介する。さらに、並列、超並列アーキテクチャの汎用機応用へのアプローチを示す例として、並列メインフレーム、超並列サーバを紹介する。

2.2 ペタフロップスイニシアティブ

2.2.1 組織

ペタフロップスイニシアティブは、NASA 本部を中心として、NSF（National Science Foundation）、DOE（Department of Energy）、DARPA（Defence Advanced Research Program Agency）、NSA（National Security Agency）、BMDO（Ballistic Missile Defense Organization）が参加し、活動を推進している。ここで、ジェット推進研究所（Jet Propulsion Laboratory）はペタフロップス研究の推進役とされてきた。

NASAはペタフロップスコンピューティングシステム開発を支持し、必要技術を探究する

リーディング省庁であるが、次の二つの観点を持ち、活動している。

- (1) ペタフロップスを確立するため、困難な技術開発に挑戦する機会を創り、かつニーズへの適用を阻む要因を見いだす。
- (2) ペタフロップス開発の研究日程を設定するための手引きを作成する。

このイニシアティブ結成の動きは、1991年、コンピュータアーキテクチャにおけるグランドチャレンジに関するPurdue大学でのワークショップに始まる。

2.2.2 活動方針

HPCCプログラムでは、テラフロップスコンピューティングを目指した研究開発がなされてきた。現在、テラフロップスマシンはIntelのRed SYSTEMに見られるように現実となったが、現在の米国の研究者達（政府機関、学会、企業）は、テラフロップスレベルのシステムでは、将来における問題はもとより、現状の多くの科学技術計算に対しても能力不足であると主張している。そのため、イニシアティブはペタフロップスコンピューティングを今後20年に亘る高性能コンピューティングを目指す象徴的な目標と捉え、研究活動を推進している。

2.2.3 活動状況

ペタオーダのコンピューティング技術開発は、1991年、コンピュータアーキテクチャにおけるグランドチャレンジに関するPurdue大学でのワークショップで初めてそのテーマの重要性が議論された。以降、関連のワークショップが毎年開催されるようになった。特に、1994年にPasadenaで開催されたワークショップでは、名称に初めて「ペタフロップス」の言葉が明示され、かつ事前に組織された各分野でワーキンググループの研究結果が発表されるなど、ペタフロップスレベルのシステム、アプリケーション開発を進展させる基礎を築いた。また、このワークショップでは、ペタフロップス研究活動の情報センタとしてWeb上にホームページを開設することが決定された。ここで、デバイス技術、アーキテクチャ、システムソフトウェア、アプリケーションの専門分野の横断的な情報を提供するようになった。

一方、NSFは“Point Design Study”と称するFunding（実際は奨励金に近い）テーマを1995年に公募し、その結果が1996年に開催されたいくつかのワークショップ、シンポジウムで発表された。これらの研究結果はペタフロップスイニシアティブ活動の研究結果の最先端を表

している。

2.2.3.1 PETAホームページ

P.E.T.A. (PetaFLOPS Enabling Technologies and Applications) は、これまでにない巨大なコンピューティング技術の研究コミュニティを形成することによって、ペタフロップスコンピューティングを効果的に確立するために提供されたWWWベースの情報源である。P.E.T.A. はNASAのHPCCプログラムオフィスによって後援され、学際的な協力の奨励、促進が行われている。ハイパーリンクを利用し、米国中に散らばったペタフロップス関連の研究者をつないでいる。この情報ハブはペタフロップスの発展とアプリケーションに対しインパクトを与えることが期待されている。このホームページは、初め NASA の CESDIS (the Center of Excellence in Space and Information Sciences) のホームページの中にあったが、現在はNASA本部のホームページの中に移設されている。

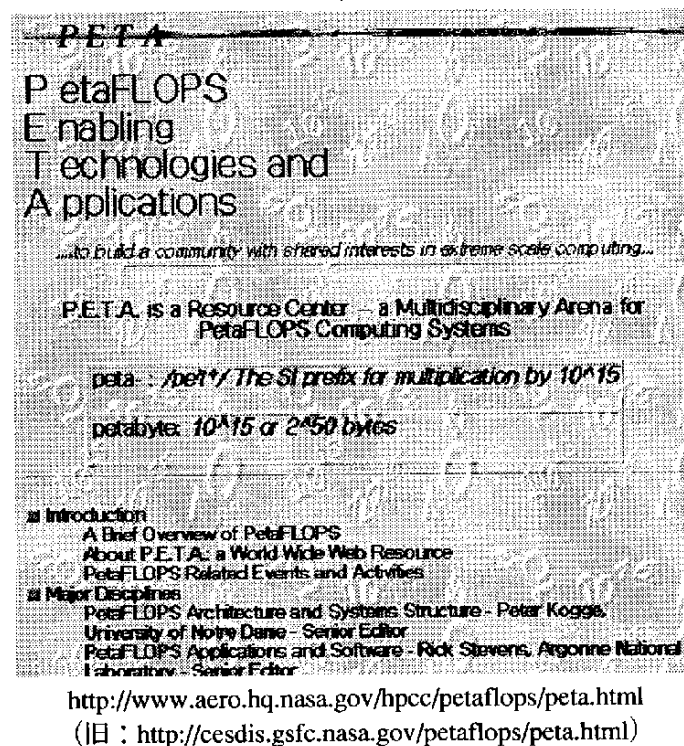


図1 PETAホームページ

2.2.3.2 各種ミーティング開催

◆1991年 Purdue Workshop開催

このワークショップは、NSFをスポンサーとした高性能コンピューティングのため

のコンピュータアーキテクチャ分野でのグランドチャレンジに関するもので、今後10年間に於ける重要な研究トピックスとして、次の4つを挙げている。

- ・理想的な並列計算機モデル
- ・Peta-ops (10^{15} ops) 性能の実現
- ・HDTV、ギガバイトネットワーク、可視化の時代におけるコンピューティング
- ・アーキテクチャを作成するためのインフラ

ここで、最初にペタが研究対象として現れてきた。このワークショップでの招待講演 "The Purdue Workshop on Grand Challenges in Computer Architecture for the Support of High Performance Computing" は、The Journal of Parallel and Distributed Computing、Volume 16、Number 3、November 1992、pages 199 - 211 に掲載されている。

◆1992年 Workshop and Conference on Systems Software and Tools for High Performance Computing Environmentsが開催される。

◆1993年5月 Pittsburg Workshop and Conference on Grand Challenge Applications and Software Technology が開催される。

◆1994年2月 The Workshop on Enabling Technologies for Peta(FL)OPS Computingが開催される。

このワークショップはNASA、DOE、NSF、Advanced Research Projects Agency、National Security Agency、BMDOによって後援され、Pasadena での開催であった。事前にアプリケーションとアルゴリズム、デバイス技術、並列アーキテクチャ、システムソフトウェアとツールの各分野でワーキンググループが組織された。会議では、各ワーキンググループの研究結果発表が行われ、招待者のみの会合ではあったが、多くの成果を出し、会議録が公表された。この時、ホームページP.E.T.A.の設置が決まり、ペタフロップスコンピューティングに関する広範な研究活動が公になった。この会議の会議録 "Enabling Technologies for PetaFLOPS Computing" は、Web上で閲覧できる (<http://nhse.npac.syr.edu/roadmap/petaflops/peta.html>) が、MIT Pressから単行本が発売されている。これは、平成9年5月末に筑波出版会により邦訳が出版される予定である。

◆1995年2月 Petaflops Frontiers '95が参加自由な第5回超並列処理フロンティアシンポジウム (IEEE後援) として開催される。開催目的は、システム技術とアプリケーションからの要求という観点で計算能力の限界を見極めることであった。そこでは、研究方向を見定めるための挑戦的な研究についての理解を深めた。また、マシンアーキテクチャと実現技術のスケジュールを見定めた。さらに、並列アプリケーションとアルゴリズム

ムの性質と必要性を決定した。

◆1995年8月 PetaFLOPS Computing Summer School/Workshop : Applications and Algorithms Challenges for PetaFLOPS Computing が ARPA、DOE、NASA、NSA、NSF の後援で開催される。このミーティングの会議録は、Webで公開されており (<http://www.mcs.anl.gov/summer.study/index.html>)、各種アプリケーション側らのアプローチが詳述されている。

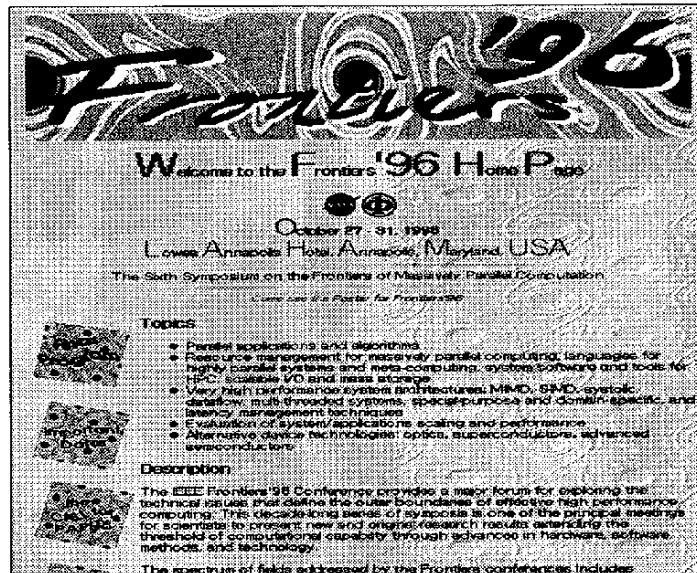
◆1996年4月 Petaflops Architecture ワークショップ開催 (Oxnard, CA)。後援はARPA、DOE、NASA、National Security Agency、NSFである。

◆1996年6月 Petaflops Summer Study on System Software開催 (BodegaBay, CA)。後援はARPA、DOE、NASA、National Security Agency、NSFである。

◆1996年10月 Petaflops Frontiers'96シンポジウムが、第6回超並列処理フロンティアシンポジウム (IEEE後援) をとして開催される (Anapolis, MD)。本シンポジウムに関する情報は、開催通知・論文募集については新規開設のホームページに掲載されているが、その結果報告等はまだ公となっていない。

シンポジウムではおおよそ次のことが議論された。

- (1) ペタフロップスマシンについて、マシンの持つイメージやアプリケーションの候補等が提案され、その中で特定目的マシンで、ペタフロップスを2000年に実現しようというGRAPE-6の計画が発表された。
- (2) システムに関しては、メモリ重視のアーキテクチャによる性能向上が提案された。また、階層的モデルが提案された。全般にアプリケーションに比べコンピュータビジョンの無さが指摘されたが、GRAPEはその中でも期待されていた。
- (3) ソフトウェア関連でのイノベーションが望まれている。
- (4) デバイス関連では、単一プロセッサの性能向上が強く望まれ、超伝導素子が期待されていた。集積度に問題があるが、現状で100GHzを越えるクロックで動作するのは超伝導素子しかなく、かつ超低消費電力であることが理由である。一方、メモリに関しては半導体メモリが唯一解に近い。超伝導素子以外で期待されているデバイスアーキテクチャはPIM (Processor In Memory) である。
- (5) NSFからのFundingを受けたPoint Design Study の発表があった。参加者からは、安定した資金供与に対する大きな要望がある。



<http://www.aero.hq.nasa.gov/hpcc/front96.html>

図2 Frontiers'96ホームページ

2.2.3.3 Point Design Study (NSF Funding)

1995年1月24日、NSFはHPCCの主要研究テーマの一つであるグランドチャレンジを進めるために、Fundingテーマの募集を行った。そこでは、次世代コンピュータのプロセッサ数、メモリ階層数、通信の遅れがいずれも10～100倍になることを想定した上で、高性能コンピュータの使い勝手の良さの向上が求められた。この中にはsolicitedの募集項目として、

- ・ New Technologies Program (ASC : Advanced Scientific Computing(NSF))
- ・ Microelectronics Systems Architecture Program (MIPS Technologies, Inc.)
- ・ Computer Systems Program (CCR : Computer and Computation Research(NSF)、ARPA、NASA)

が含まれていた。

少なくとも、100TeraOpsの性能を10年で実現するシステムを対象に、6つから8つの研究に対しそれぞれ10万ドルの賞金を与えるというアナウンスであった。スケジュールとして、プロポーザルは1996年4月1日までに提出されなければならず、4月中旬までに該当研究が決定され、それらの研究初期の結果は、1996年4月に開催されるPetaFlops architecture workshopと1996年7月に開催されるPetaFlops software summer studyで発表され、最終レポートは1996年11月1日までに書き終えるというものであった。

結果として次の8テーマが採択され、1996年10月27～31日に開催されたFrontiers '96のワー

クショップとテクニカルプログラムで発表がなされた。

これらの結果は、ペタフロップスイニシアティブが関与する研究成果の重要な部分を形成している。

- ・ MORPH : A Flexible Architecture for Executing Component Software at 100 Teraops
- ・ Pursuing A Petaflops : Point Designs for 100 TF Computers Using PIM Technologies
- ・ Architecture, Algorithms and Applications for Future Generation Supercomputers
- ・ Design Studies on Petaflops Special Purpose Hardware for Particle Simulations
- ・ Hybrid Technology Multithreaded Architecture
- ・ Hierarchical Processors-and-Memory Architecture for High Performance Computing
- ・ The Illinois Aggressive Cache-Only Memory Architecture Multiprocessor (I-ACOMA)
- ・ A Scalable/Feasible Parallel Computer Implementing Electronic and Optical Interconnections for 156 Teraops Minimum Performance

2.2.4 主な研究成果

ペタフロップスコンピューティング実現のための要素分野として、

- ・ デバイス技術
- ・ アーキテクチャ
- ・ システムソフトウェア
- ・ アルゴリズム
- ・ アプリケーション
- ・ ビジネスモデル

が挙げられている。

Webに載せられている成果を以下に記述する。

2.2.4.1 デバイス技術

- (1) SIA (Semiconductor Industry Association) の1992年版のロードマップの2007年までの技術トレンドをもとに、使用可能なCMOS性能の外挿を行っている。2013年にCMOSは0.05ミクロンルールが適用され、0.9 Vの電源電圧 (Vdd) で2,000MHzの内部クロックで動作すると予想されている。これはロジックチップでは8,000万ゲート、DRAMでは

256ギガビットに相当する規模のLSIである。1995年版SIAトレンドとの整合性は良く、内部クロック1,100MHzを除くと上記の傾向は同じである。表1にSIAによる1992年版ロードマップを示す。

表1 SIAロードマップ (1992年版)

Table 5.1
National Roadmap for Semiconductor Technology (1992 Version)

Year	1995	1998	2001	2004	2007
Feature size (microns)	0.35	0.25	0.18	0.12	0.1
Gates/chip	800k	2M	5M	10M	20M
Bits/chip DRAM	64M	265M	1G	4G	16G
Bits/chip SRAM	16M	64M	256M	1G	4G
Wafer Processing Cost (\$/cm)	3.90	3.80	3.70	3.60	3.50
Chip size (mm ²) logic per microprocessor	400	600	800	1000	1250
Chip size (mm ²) DRAM	200	320	500	700	1000
Wafer diameter (mm)	200	200-400	200-400	200-400	200-400
Defect Density (defects/cm ²)	0.05	0.03	0.01	0.004	0.002
Interconnect levels (logic)	4-5	5	5-6	6	6-7
Max power (W/die) high performance	15	30	40	40-120	40-200
Max power (W/die) portable	4	4	4	4	4
Power supply (V) desktop	3.3	2.2	2.2	1.5	1.5
Power supply (V) portable	2.2	2.2	1.5	1.5	1.5
Number of I/Os	750	1500	2000	3500	5000
Performance (MHz) off-chip	100	175	250	350	500
Performance (MHz) on-chip	200	350	500	700	1000

(<http://nhse.npac.syr.edu/roadmap/petaflops/peta.html>)

- (2) ペタフロップスレベルのコンピューティングを効率良く実現するには、レイテンシ管理が決定的な意味を持つ。レイテンシを低くするためには高速メモリが必須である。
- (3) ペタフロップスマシンのコストはDRAMを念頭に置くとメモリによって決まる。プロセッサ/メモリ比はシステム設計の重要なパラメータである。データセットを小さくし、ホログラフィを用いた光学技術とアウト・オブ・コアの技術を用いると、システムコストを大幅に削減でき、他の方法より早くペタフロップスマシンを達成できる。分散共有メモリ構造はプログラミングやシステムソフトウェアを簡略化できるが、レイテンシやバンド幅に問題がある。
- (4) 超伝導素子 (RSFQ: Rapid Single-Flux Quantum logic, K.K.Likharev) は100GHzを越えるクロックを超低消費電力で実現できる可能性があり、注目すべきである。

- (5) 光ネットワークはバンド幅を確保し、レイテンシを低く抑えるのに有効であり、アーキテクチャのいたるところに使用できる。また、自由空間光学を用いると、接続の複雑さを軽減できる。

2.2.4.2 アーキテクチャ

- (1) 今日、ペタフロップスマシンを構築するコストについての考察として、現在100メガフロップスのボードは大体1万ドルし、ペタフロップスの実現のためには、このボードが1千万枚必要で、約1,000億ドルが必要となると見積もられている。これは、このままでは実用的なものではないことを表している。
- (2) ペタフロップスマシンに必要なメモリ量の考察の結果として、3次元空間と時間を含む問題（大抵の科学技術計算）では「NギガフロップスはNの3/4乗ギガバイトのメモリを必要とする」という法則が適用できることがわかった。言い換えると、ペタフロップスシステムは32テラバイトのメインメモリがあればよいことになる。この結果は他の報告でも良く見かけられるため、この計画で得たかなり重要な知見であると思われる。
- (3) ペタフロップスレベルを視野に入れたアーキテクチャの本命として、より広範囲なアプリケーションに適用でき、コスト面で有利という点で、まず、400テラフロップスの個別性能を持つマシンを作り、それらを繋げるのが良いということが結論されている。

- (4) アーキテクチャのコンセプトを進めるのは、

- ・先進的プロセッサ設計
- ・ハード／ソフトに採用されるべき積極的なレイテンシ管理技術
- ・システム相互接続の構造と方法

であり、アーキテクチャの候補として次の4つが選ばれた。

- ・広大な市場の投資を開拓するCOTS (Commercial Off The Shelf) を基盤としたマルチプロセッサ
- ・COMA (Cache Only Memory Architecture) を実現する低電力かつ高メモリバンド幅のPIM
- ・超伝導プロセッサと半導体メモリの深い階層構造、および光学的メモリと光ネットワークを用いたハイブリッドなマルチスレッドNUMA (Non-Uniform Memory Access model)
- ・協調して再構成が可能な特殊用途デバイス

- (5) チップ面積が一定の場合、PIMのように同じダイにMPUとメモリを同時に集積し、メモリバンド幅を大きくできれば、低消費電力の小プロセッサを多数集めたほうが大消費電力の高性能プロセッサを少数使うより性能を上げられる。ただし、アムダールの法則は少数プロセッサ方式に有利に働く。
- (6) プロセッサ－メモリ間のバンド幅を非常に大きくしなければならないが、このため、PIMをはじめとする階層的な構造が有利である。また、接続技術、トポロジ、オン・ザ・フライ－データ圧縮などの技術が必要である。
- (7) 最初のペタフロップスマシンは、単一のアルゴリズムを実行するカーネルの要求を満足させるデータパスやロジックを持った特殊用途向けのものになる。これは、5年以内に1,000万ドル程度でできる。
- (8) COTSを用いず、特殊な方法を見つけることによって、ペタフロップスをより早く達成できると考えられている。

2.2.4.3 システムソフトウェア

ペタフロップスマシンを使うのに必要な基礎的なソフトウェアはあるが、オリジナリティがあって挑戦的なシステムソフトウェアが見当たらない。並列度やレイテンシのような重要なシステムパラメータのスケーリングの大きさが、システムソフトウェアの負荷を顕著に重くするため、コンパイル時間と並列実行、データセット、物理資源のランタイム管理の分野で革新的な新ソフトウェアが必要となっている。

2.2.4.4 アルゴリズム

ペタフロップスにおける並列度は100,000から1,000,000が予想できるが、これを実現するためにはアーキテクチャ、システムソフトウェアはもとより新しいアルゴリズムを開発する必要がある。

並列度を数桁上げ、レイテンシ保証した通信から情報処理を分離するために、新しいクラスのアルゴリズムが必要である。そのアルゴリズムは、区別された実行プロファイルを多数に分離したものからなるメタ－アプリケーションへの移行によって複雑化するであろう。

2.2.4.5 アプリケーション

以下に示すような広範囲の研究分野を応用として目標を明確化している。

(1) 高速処理技術

- ①画像処理と画像認識
- ②ニューラルネットワーク
- ③記号処理
- ④データベースと問い合わせ処理
- ⑤暗号法
- ⑥視覚と幾何学計算処理
- ⑦デジタル信号処理
- ⑧レンダリングとグラフィック
- ⑨電子図書館とマルチメディア

(2) 数学・論理学

- ①自動推論／証明
- ②代数学
- ③数論
- ④グラフ演算法
- ⑤符号理論
- ⑥ゲームプレーとゲーム検索
- ⑦最適化とスケジューリング問題
- ⑧多くの学問領域に及ぶ応用

(3) 物理・化学問題

- ①応力蓄積モデリング／地震
- ②天体物理学／一般相対性
- ③量子色力学と粒子物理学
- ④ゲノム情報学と系統発生学
- ⑤分子（動）力学、電子構造と材料物性
- ⑥散乱とプラズマ物理学
- ⑦反応性流体と燃焼、ポーラス材料内の流れ
- ⑧地球物理学上のモデリング
- ⑨生物圏のモデリング

(4) シミュレーション

- ①動的システムシミュレーション
- ②不連続事象シミュレーション
- ③電子デバイスシミュレーション
- ④回路シミュレーション
- ⑤制御システム
- ⑥経済モデル
- ⑦ストックオプション値付けモデル

(5) その他考えられる適用分野

- ①多人数参加型バーチャルリアリティ
- ②ジェスチャーの認識
- ③個人ベースの環境経済モデル
- ④画像ベースの検索と映像フローの分類
- ⑤消費者によるインタラクティブな消費者製品デザイン
- ⑥深層論法：1000命令／推論として、ベタフロップスならテラ回の推論
- ⑦回路設計、回路検証、推論補助
- ⑧証明の検証
- ⑨CADとビデオベースシミュレーションの融合
- ⑩コンピュータ合成による映画（ニュース）

- ⑪不特定多数の音声認識、手書き文字認識
- ⑫完全知識ベース
- ⑬地球全体の経済ならびにエネルギーモデルの入出力モデル置換
- ⑭ワールドワイド製品とサービスの、自由市場における値付け

2.2.4.6 ビジネスモデル

産業的および学問的研究開発やハード／ソフト生産の分野にわたり、かつその経済的基盤を支えるペタフロップスコミュニティを発展させ、支える方法は、有用なペタフロップスコンピューティングを実現することである。新しいビジネスモデルは全ての参加者の関係と役割を再定義する必要がある。

2.3 テラフロップスマシン開発状況

2.3.1 米国のASCI (Accelerated Strategic Computing Initiative) プログラム

(<http://www.sandia.gov/ASCI/>、<http://www.lanl.gov/Internal/projects/asci/>)



図3 ASCIの各ホームページにあるマーク

米国は地下核実験全面停止に際し、核弾頭の安全性、信頼性、爆発性能を評価する新方法を必要としていた。また、新たな核兵器製造は止めていたが、核兵器設計力、性能解析力、安全性と信頼性予測力は維持しなくてはならなかった。コンピュータ性能の著しい向上によって、核兵器の仮想実験や仮想作製が現実のものとなってきた。基本的なものは現在でも存在するが、実用とするために3次元空間中で全システムで起きる全物理現象を、より高分解能でシミュレートしなければならない。これは現在の技術レベルをはるかに超えるものである。

DOEは、ASCIプログラムを計画し、10年で10億ドルを投入する計画である。ここで、ASCIはDOEのSandia、Los Alamos、Lawrence Livermoreの3国立研究所の共同研究であり、図4に示すように、2004年までに100Tflopsを実現しようとしている。

(http://www.cray.com/PUBLIC/WHATS_NEW/COMPANY/OCT96/ASCI.html)

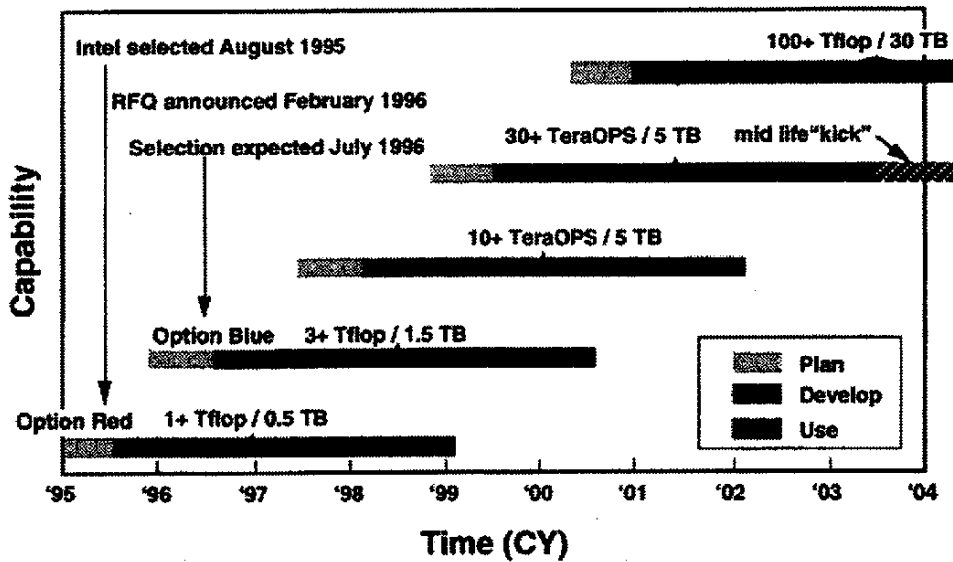
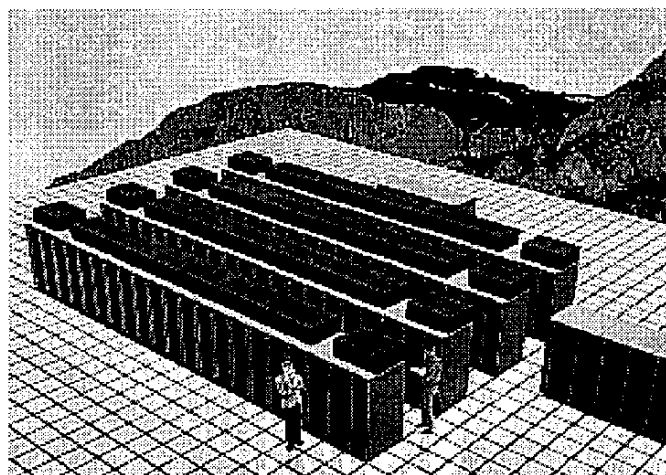


図4 ASCIプログラムロードマップ

2.3.1.1 Sandia国立研究所 (<http://www.acl.lanl.gov/~pfay/teraflop/>)

ここでは、1テラフロップスを実現するために、ASCI Red と呼ばれるマシンを1996年に導入する計画が立てられた。このマシンは最初の完全な商用部品 (commodity, commercial, off-the-shelf (C-COTS)) の大型スーパーコンピュータであり、部品は何百万ものデスクトップコンピュータやサーバと同じ部品が使われる。このマシンは Intel の Scalable Systems Division (Beaverton, OR)で開発され、MPUは9,000個以上の Intel Pentium Pro が使われる。



(<http://www.acl.lanl.gov/~pfay/teraflop/>)

図5 Intel ASCI Red

Intelの公開情報によると以下に示すような開発経過があった。

(<http://www.ssd.intel.com/press/asci1.html>)

1995年9月7日発表 核兵器仮想実験のため、DOEのDefense Programs (DP) laboratoriesの中にASCIが複数年度プログラムとして成立した。ASCIは独立したプロジェクトではなく、DOEのStockpile Stewardship Programの一部である。ASCIはDOEのAdvanced Design and Production Technologies (ADaPT) initiativeに対し、仮想製造シミュレーションを提供する役割を負った。

Intelは、ここで兵器以外に新薬開発などにテラフロップスコンピュータの効用があることを述べている。

1995年の8月以降発表 Intelは4,600万ドルでテラフロップスマシン開発を受注した。ASCIプログラム(複数年度研究)はDOEのHPCコンピューティング研究を加速するものであり、米国はテラフロップスに10億ドルを投資する(1995年6月23日)ことが示された。

1996年5月 ASCI Red、20 Gflops (64-node) を実現。

1996年11月 ASCI Red、213 Gflops (1,336 processor) を実現。

1996年12月 ASCI Red、1Tflopsを実現。(ここにきて\$55 million machineと記述している。)

さらに、次のようなアプリケーション(科学技術計算を除く)研究があることを主張するようになった。(Sandia 国立研究所の Massively Parallel Computing Research Lab. より)

- (1) 電子マネー(プライバシーの保護と犯罪防止の両立)
- (2) 仮想現実技術(ニューラルネットワークや人工知能は不使用)
- (3) 軍事(核実験)
- (4) Aspen(米国経済のミクロ的解析シミュレーションモデル)
- (5) データの可視化
- (6) 外科手術(大規模デジタルデータセットの処理)
- (7) 医用データのプライバシーと完全性の保護

Intel ASCI Redの仕様概要を表2に示す。

表2 ASCI Red の仕様

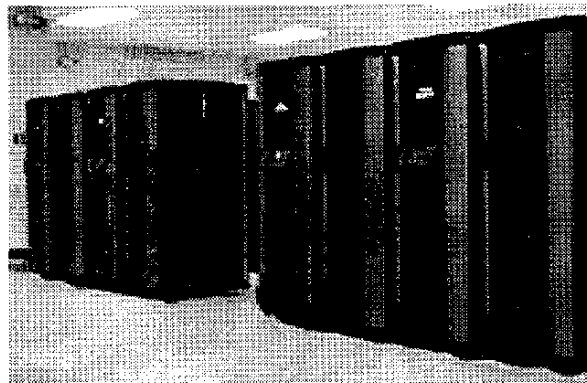
Compute Nodes	4,536
Service Nodes	32
Disk I/O Nodes	32
System Nodes (Boot and Node Station)	2
Network Nodes (Ethernet, ATM)	6
System Footprint	1,600 Square Feet
Number of Cabinets	85
System RAM	594 Mbytes
Topology	38x32x2
Node to Node bandwidth - Bidirectional	800 Mbytes/sec
Bidirectional - Cross section Bandwidth	51.6 Gbytes/sec
Total number of Pentium Pro Processors	9,216
Processor to Memory Bandwidth	533 Mbytes/sec
Compute Node Peak Performance	400 MFLOPS
System Peak Performance	1.8 TFLOPS
RAID I/O Bandwidth (per subsystem)	1.0 Gbytes/sec
RAID Storage (per subsystem)	1 Tbyte

(http://www.sandia.gov/ASCI/)

消費電力は 800 KW、OSはUNIX ベースの Paragon OS に高性能並列ファイルシステム (PFS) を付加した。

2.3.1.2 Los Alamos国立研究所 (http://www.lanl.gov/projects/asci/)

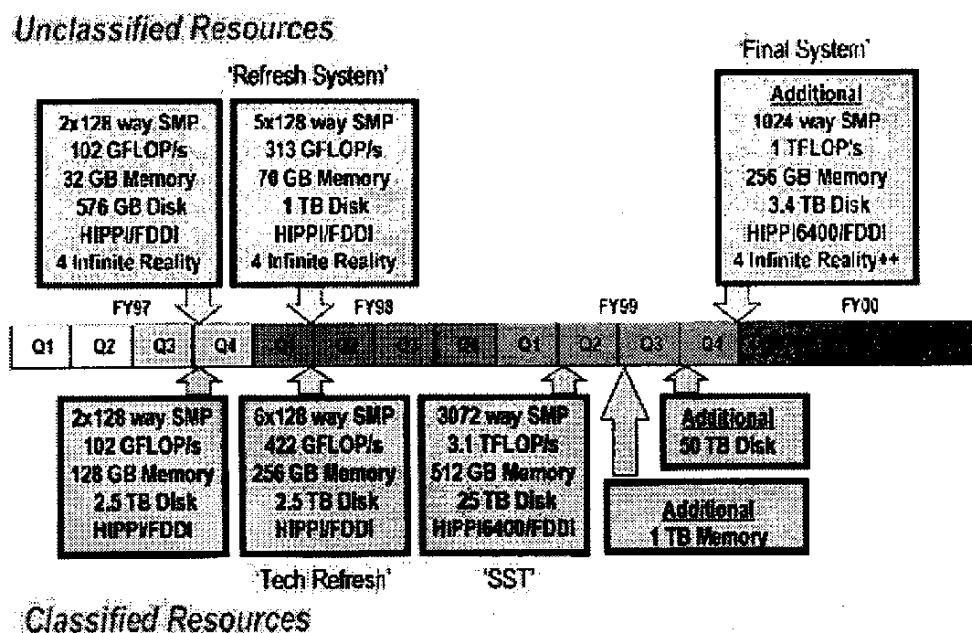
ここでは、ASCI Blue Mountain と呼ばれるシステムが開発されている。納入されたのは、Origin2000 (CPU : MIPS R10000、195MHz) と呼ばれるCRAY製のSMP型コンピュータである。OSは、Blue Mountain OSと呼ばれるCRAY製のCellular IRIXが用いられている。



(http://www.lanl.gov/projects/asci/)

図6 ASCI Blue Mountain

システムの導入計画を図7に示す。これから1996年11月に契約を結び、1997年2月に102 Gflopsのマシンを導入した後、1999年第一四半期に3.1 Tflopsを達成する予定である。

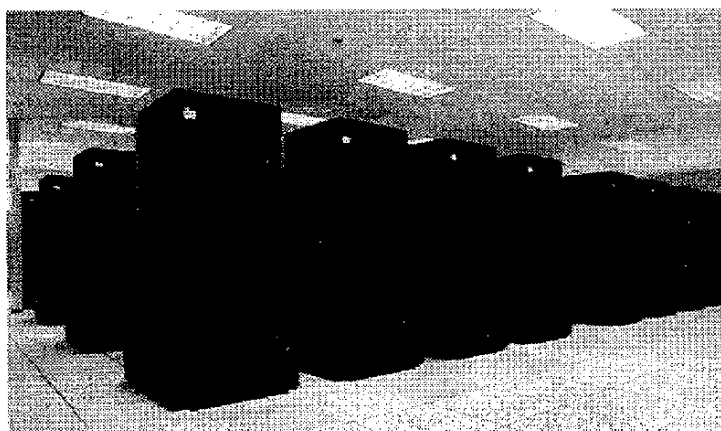


(<http://www.lanl.gov/Internal/projects/ascii/bluemtn/schedule.html>)

図7 Los Alamosの導入予定

2.3.1.3 Lawrence Livermore国立研究所 (<http://www.llnl.gov/sccd/lc/ascii/>)

ここでは、ASCI Blue Pacific と呼ばれるシステムが開発されている。開発契約は9,300万ドルでIBMと締結され、納入されるマシンはRS/6000SPである。IBMは1996年度のASCIプログラム予算8,500万ドルのうち2,700万ドルを獲得している。



(<http://www.llnl.gov/sccd/lc/ascii/llnl-ibmexcerpt.html>)

図8 ASCI Blue Pacific

1999年までの達成予定性能は、512-node, 8-way SMP configurationで、ピーク性能3.1Tflops、メモリ容量2.5TB（3GB/node、64nodeは16GBを有する）、ディスク容量75TBである。この構成により、中程度から非常に大きな三次元アプリケーションに対応でき、本研究所のコンピュータを利用する一般ユーザへのシステムの開放も速やかに移行できると主張されている。

この研究所では、広いスペクトルを持つ国家目標の達成のためには、国立研究所、大学、産業の新しい形の協力が必要であると強調されており、ASCIプログラムは、NSFと大学を協力に取りこめる方法の一部であると考えられている。

2.3.2 CSCC (Concurrent Supercomputing Consortium)

CSCC構想は、市販機より非常に高性能な分散メモリMIMD型コンピュータが浮上して来た1990年春に起草された。当時、同様なアーキテクチャのコンピュータで動作はしていたが、十分な結果を出すためには、より高速処理と多くのメモリを必要とするアプリケーションプログラム開発を進めるために、このコンソーシアムは、絶好の機会となった。また、テラフロップスシステムで用いられるアルゴリズムと実現技術を探索する絶好の場となった。

CSCCは次の12の機関により結成された。

- ・ ARPA
- ・ Argonne National Laboratory
- ・ California Institute of Technology
- ・ Center for Research on Parallel Computation
- ・ Intel Corporation's Scalable Systems Division
- ・ Jet Propulsion Laboratory
- ・ Caltech
- ・ NASA's High-Performance Computing and Communications Program
- ・ NASA's Office of Space Science
- ・ NSF
- ・ Pacific Northwest Laboratory
- ・ Purdue University

これらの機関は、このコンソーシアムを通じ、それぞれのリソースをプールし、相互に利用し、情報を交換することによって、高性能コンピューティングに関する諸問題について協力することになった。

CSCCでは、高性能コンピューティングを集中的に研究し、これまでにない性能を持ったコンピュータを用いることによって、これまでのコンピュータでは実行できないような規模

の重要な問題を解こうとした。そのため、CSCCのコンピュータリソースのほとんどを数少ない大規模の計算に関する研究プロジェクトに提供した。

これにより、科学技術分野での新しい成果を得るだけでなく、並列計算機のハードウェア／ソフトウェアを改善し、次世代高性能コンピュータ開発につなげようとした。

結果として、CSCCでは、これらの得られた知識をスケーラブルなソフトウェアとして残した。情報は、ワークショップ、ニュースレター、技術レポートを通して交換された。

研究者等を適切に訓練することは重要な資源を蓄積することになる。CSCCは、それに属する研究機関で行われる多種多様な計算関連の研究プログラムによって、次世代の計算関連の科学・技術者の教育・訓練に貢献することも目的とした。計算関連の科学プロジェクトには多くの訓練が自然に組み込まれた。大学での研究や研究所での一時任用、インターンシップ、短期訓練コース開催などで相当な数の研究者を教育・訓練した。

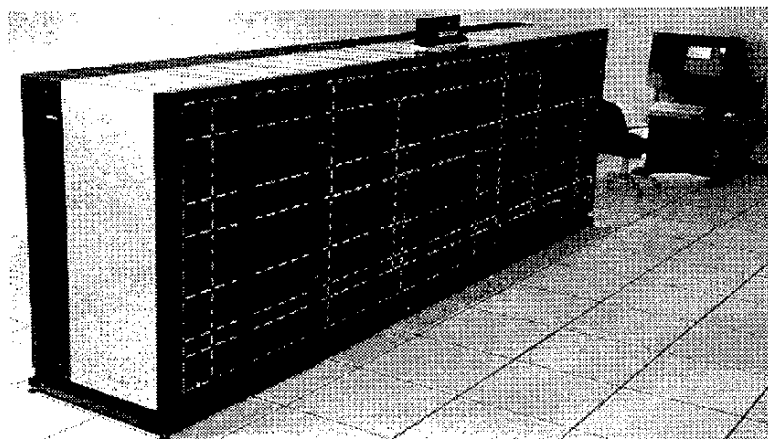
CSCCは、1990年夏に導入システムを決定し、それを受けてIntelはシステムの作製を9月下旬に始めた。7カ月後の1991年5月に568ノードの Touchstone Delta System と 64ノードの iPSC/860 hypercube がCaltechに納入された。(1992年CSCC年報)

1995年度CSCC年報によると次のようなハードウェアの更新を含むイベントがあった。

1990年11月 CSCC結成

1990年11月 Ajax (64-node Intel iPSC/860) 導入

1991年5月3日 Intel Touchstone Delta (over 30 peak Gflops) 導入



(http://www.cacr.caltech.edu/publications/annreps/annrep94/intro_1.html)

図9 The Intel Touchstone Delta System

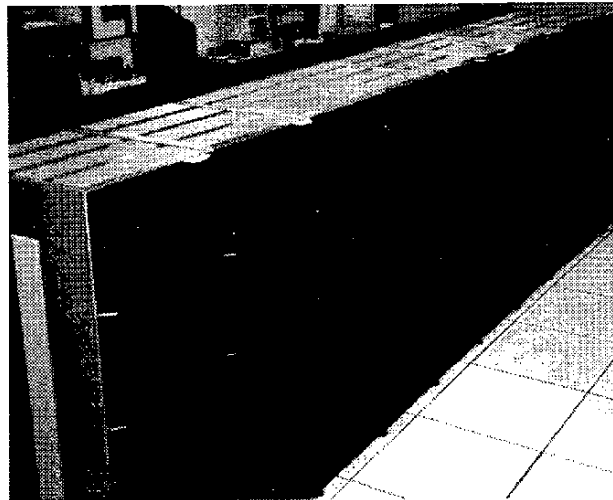
1992年10月19-20日 CSCC 政策局が2-3年の間の主要テーマとしてスケーラブルI/O イニシアティブを支持した。そこでは、I/O関連のシステムソフトウェア部分（コンパイラ、ランタイムライブラリ、並列ファイルシステム、高性能ネットワークインターフェース、OSサービス）の開発を進め、ARPA、DOE、NASA、NSFの出資により、超並列システムの試作機により、アプリケーションのI/O性能をフルスケールで評価することを目的とした。

1993年2月8日 Ajaxの次期機種としてIntel Paragon A4（Raptor：52ノード）導入

1993年3月25-26日 Second CSCC Delta Applications WorkshopがNorfolk, Virginiaで開催される。

1993年9月 Deltaの全ディスクシステム変更によりシステムの安定化達成

1993年12月10日 Intel Paragon L38（Trex：512 nodes、38.4 Gflops）導入。ParagonはDeltaとハードは非常に似ているが、OS（OSD/1 AD）は全く異なる。



(http://www.cacr.caltech.edu/publications/annreps/annrep94/intro_1.html)

図10 CSCC に納入されたParagon（Trex）

1993年12月 Ajaxがインテルに返還される。（Ajaxはその後、他ユーザに売却され、分割化されたようである。）

1995年 CSCCは1996年12月をもって6年間の活動が終わることが決定され、Caltechはその続きとしてCaltech's Center for Advanced Computing Research（CACR）を設立することになった。

CSCCが残した成果としてSandia国立研究所でのIntelの約10,000プロセッサシステム(ASCI Red)導入がある。このシステムはIntel Touchstone Deltaと非常に似たアーキテクチャを持っている。また、スケーラブルI/Oイニシアティブによって「スケーラブルI/O」という言葉が一般化された。ここでは、インテルのParagonに多数のディスク等を接続し、パラレルI/Oのハードウェア/ソフトウェアの開発を行った。ソフトウェアでは、並列チェックポイントツールがプリンストン大学により作成されている。

CACRは商用プロセッサのクラスタを用いる研究を始めた。これは、HPCCプログラムであるNASAのBeowulfパラレルワークステーションプロジェクトが元になっている。

新しい主力並列システムは、1997年6月までにHewlett-Packard Convex SPP 2000を256プロセッサに増設し、184 Gflopsにする。SPP 2000は、分散共有メモリアーキテクチャを有し、それを支える優れたハードウェア/ソフトウェアが導入される。メッセージパッシングプログラムパラダイムがサポートされる。

CACRはHewlett-PackardのConvex Divisionと協力し、OSやマルチプロセッサ上の共有メモリプログラムを開発する。HPのConvex Divisionは1999年、CaltechにSPP3000の一号機を納入する。この装置にはHPとインテルが共同開発している「Merced microprocessor」が使われる。

2.3.3 日米各社の(サブ)テラフロップス・マシン

巨大情報処理能力の必要性は、単に膨大な科学技術計算の需要だけでなく、データウェアハウス、データマイニングなどのビジネス応用にも出現している。この要求に応えるために、各種スーパーコンピュータに加え、汎用機の並列化や超並列サーバの登場、さらに、データベース処理対応として大型UNIX系並列サーバが登場している。

現在、本命視されているCMOSテクノロジーは、微細化によるスケーリング則が適用でき、性能向上が順調になされてきた。また、開発費を含めたコスト低減効果も大きく、従来のスーパーコンピュータに搭載されている高性能プロセッサを作製してきたバイポーラテクノロジーを駆逐しつつある。開発費は10分の1、性能もほぼ同等である。プロセッサ単体の性能向上の限界は、SIAのロードマップで示され、それを大きく越えた性能は実現が困難である。プロセッサのクロックタイムを例にとると、2010年で600MHz、2010年で1.1GHzと予想されている。

このように、高性能化、低コスト化の要求を満たすのはCMOSしかなく、その性能限界も見えている現在では、より高性能なコンピュータを作製するためには並列アーキテクチャを採用しなければならない。スーパーコンピュータはもとより、汎用大型機でさえも並列化が進んでおり、そこでは各社32~64プロセッサ機を投入している。

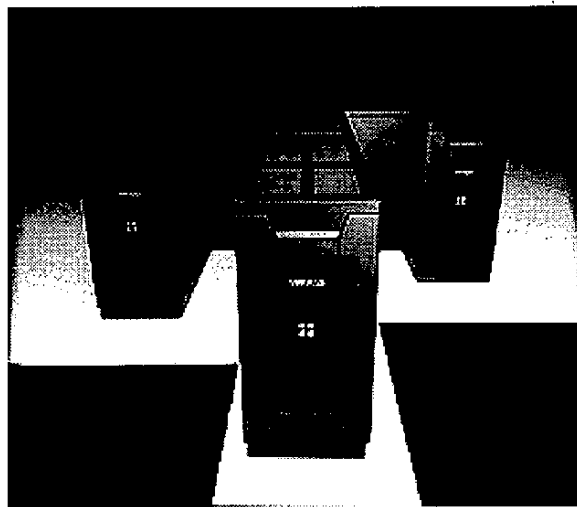
2.3.3.1 並列ベクトル型スーパーコンピュータ

この分野では、1976年に米国CRAYがCRAY-1を発表した後、日本のコンピュータメーカーが急追することになったが、現在ではCRAYはT90シリーズ（ピーク性能60Gflops（32CPU））を出荷したものの大型機は開発していない。現在、富士通、NECが大型機開発を行っており、各地で行われる入札では超並列型スーパーコンピュータと熾烈な競争が行われている。

(1) 富士通 VPP700

563.2Gflops（256Processor）

1.126Tflops（512Processor）



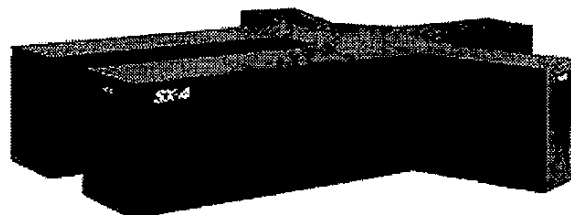
(<http://www.fujitsu.co.jp/hypertext/InfoPro/ipbo/globalit/vpp700.html>)

図11 富士通 VPP700

(2) NEC SX-4

1.02Tflops（2Gflops/processor × 512 CMOS processors）

OS：SUPER-UX



(<http://www.nec.co.jp/english/product/computer/sx/sx-p1.html>)

図12 NEC SX-4

2.3.3.2 超並列型スーパーコンピュータ

この分野においては、大学での特殊用途を除き、商用機を製造しているメーカーは米国に圧倒的に多いが、日本では日立のSR2201が高性能を実証している。

(1) Intel ASCI Red

予定性能1.8 Tflops (9,018 Pentium Pro)

0.35 μ m Bi-CMOS 200MHz 600MIPS

(2) 筑波大 CP-PACS

2048CPU 368 Gflops (実効)、614 Gflops (ピーク)

OS : HI-UX/MPP 02-00-/C + SUT

(3) CRAY T3E-900

1.8Tflops DEC Alfa (450MHz) 最大2048CPU



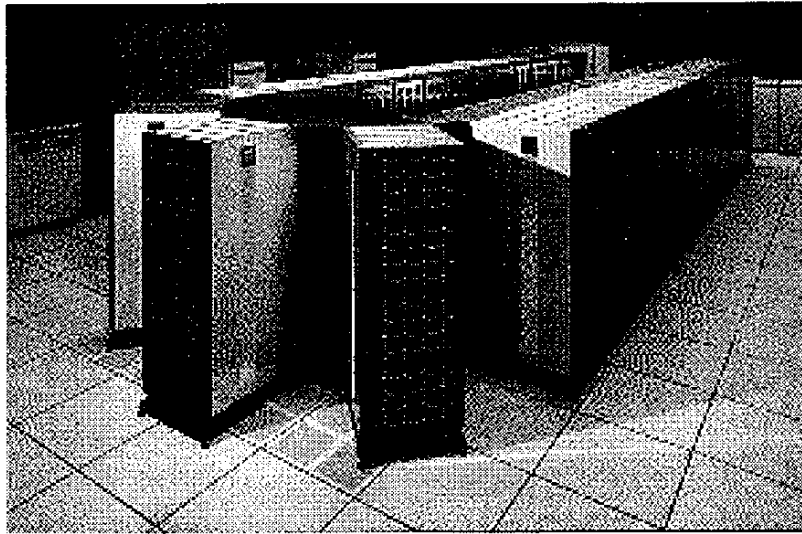
(http://www.cray.com/PUBLIC/product-info/T3E/CRAY_T3E-900.html)

図13 CRAY T3E-900

(4) 東大 数値風洞 230 Gflops (実効)

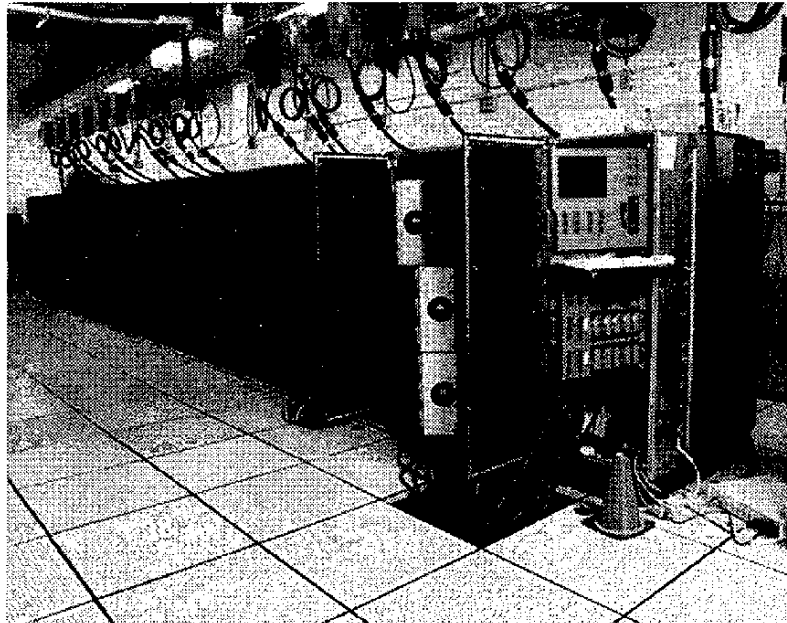
(5) 東大教養 GRAPE4 1.08 TFlops (天体物理学専用機)

(6) Intel Paragon 281 GFlops (6,798 i860XP)



(<http://www.cs.sandia.gov/mpcrl/index/iparagon>)

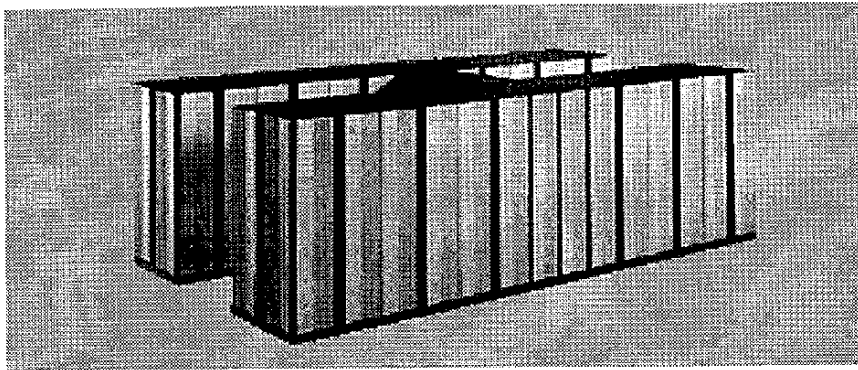
図14 Sandia 国立研究所のParagon



(<http://www.ssd.intel.com/press/temp/machine2.html>)

図15 Oak Ridge国立研究所のParagon、MP Linpackの記録達成に用いられた。

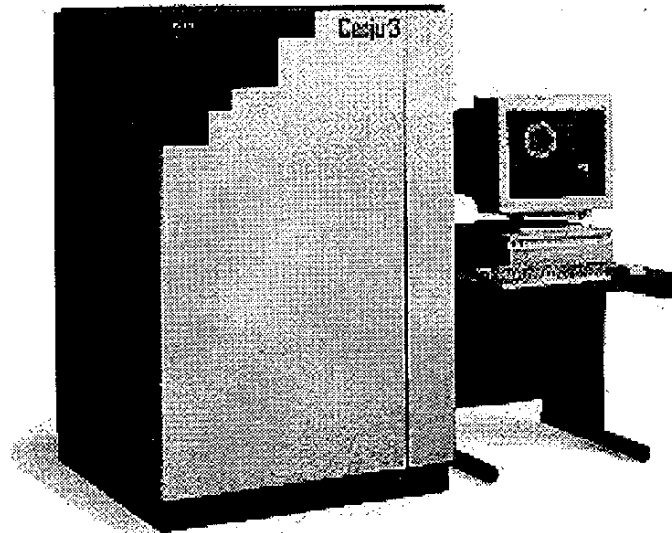
- (7) 日立 SR2201
2048processor 610 Gflops RISCチップ
疑似ベクトル処理機構
OS : HI-UX/MPP



(<http://www.hitachi.co.jp/Prod/comp/hpc/jpn/sr1.html>)

図16 日立SR2201ハイエンドモデル

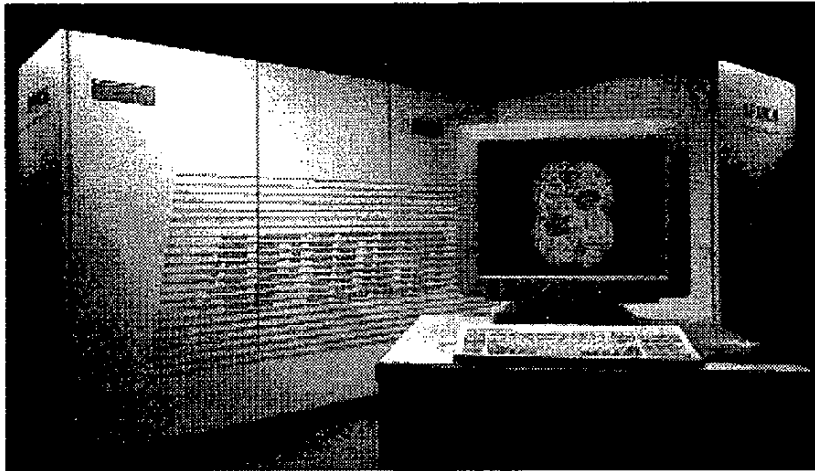
- (8) NEC Cenju-3
128processor 6.4Gflops



(<http://www.hpc.uh.edu/cenju/workshop97.html>)

図17 NEC Cenju-3

- (9) 富士通 AP1000
1024Processor 51.2Gflops SuperSPARC
(現在、UltraSPARC×1024processorを用いたAP3000を発表している)



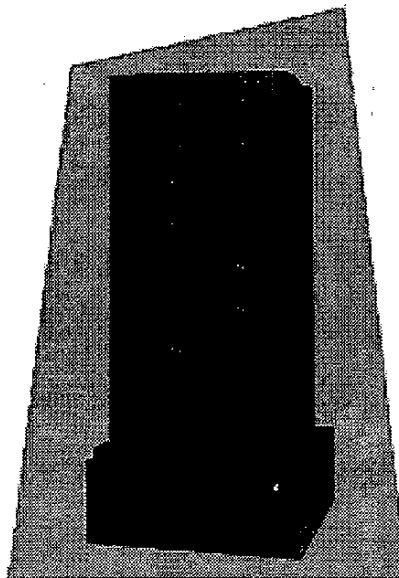
(<http://www.fujitsu.co.jp/hypertext/fpcrf/j/ap1000.html>)

図18 富士通 AP1000

(10) IBM RS/6000 (SP2)

8way 512ノード

3.1Tflops (予定：ASCI Blue Pacific参照)



(<http://www.rs6000.ibm.com/hardware/largescale/index.html#highlights>)

図19 IBM RS/6000 (SP2)

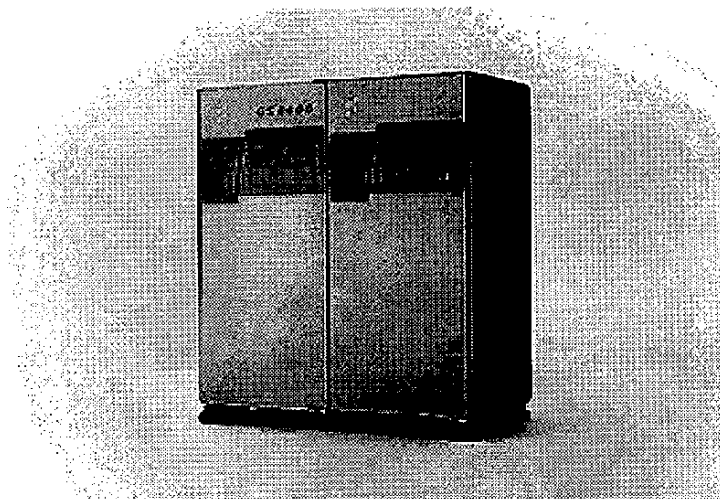
2.3.3.3 並列メインフレーム (日経ウォッチャーIBM版別冊 (95年11月) 参照)

従来の銀行オンラインシステムを代表とする業務処理を担当する大型汎用機も最近は並列化されている。CPU数で8-16プロセッサ/システムの並列度であり、処理能力も数百MIPSと

スーパーコンピュータに比較して格段に低い。データベース処理には超並列サーバを接続する場合が多い。

- (1) IBM システムコンセプト：パラレルSysplex
主要機種：9672-R2/R3 24MIPS×10PE
(データベースサーバとしてRS/6000SP (= SP2) 接続)
OS：MVS/ESA、UNIX
実績：花王の販売物流・夜間バッチ処に採用されている

- (2) 富士通 システムコンセプト：スケーラブルSCMP
(System storage Coupled MultiProcessor system)
主要機種：GS8600 0.35 μ mCMOS×64processor
DB検索用としてPD2000 (SPARC×256) 接続
OS：MSP-EX、XSP



(http://www.fujitsu.co.jp/hypertext/Products/Info_process/gs86.html)

図20 富士通 GS8600

- (3) 日立 システムコンセプト：Mパラレル・コンセプト
主要機種：MP5800 bi-CMOS×8
OS：VOS3/AS
実績：銀行口座管理オンライン、流通業DB、保険業DB、製造業生産管理
- (4) NEC システムコンセプト：パラレルACOS
主要機種：PX7800 200MIPS以上 (CMOS×32processor)

OS : DOIS

(5) ユニシス システムコンセプト : XTPA

(eXtended Transaction Processing Architecture)

主要機種 : 2200 (XPC) RISC×64CPU

DB用にOPUS2000 (Pentium×128)、Oracle 7採用

実績 : 航空座席予約システム、金融勘定系システム、札幌市役所 (XPC)

2.3.3.4 超並列サーバ (日経ウォッチャーIBM版別冊 (95年11月) 参照)

米国メーカーは、プロセッサ数、100以上の並列度を有する超並列機を業務システム等のいわゆる汎用用途に用いている。その代表的な例を示す。

(1) AT&T システムコンセプト : OCCコンセプト

(NCR) 主要機種 : 3600XP AP : P5×16、AMP : i486DX4×360

Oracle、Sybase、INFORMIX等の商用DBMSが稼働

OS : UNIX

実績 : 西友マーケティング分析・支援、A銀行情報系

(新型としてWoeldMark 5100M 4096CPU (Pentium))

(2) タンデム システムコンセプト : 疎結合非共有アーキテクチャ

主要機種 : Himaraya R4400×4032MPU

実績 : 金融、クレジット、通信、流通、製造、食品、運輸など

1.5年で63社・82システム、基幹業務系システム

(京セラ : COBOL500本を7カ月で移行)

(3) NEC 主要機種 : NX7000/PS200 1024RISC Chip (256ノード)

2.5 あとがき

米国におけるペタフロップスマシンの開発は、テラフロップスマシンの研究開発のあとをうけて1991年より、20年計画でスタートさせている。この計画は、アプリケーションからの

必要性を前面に出しており、軍事以外に、経済予測、核物理や天体物理学、生物学（遺伝子解析や蛋白質合成）、薬学、機械工学、自動推論、データベース／知識ベース処理など多数の応用例を掲げている。さらに、これらアプリケーション実現のためのソフトウェア構築や基本ソフトウェア構築も研究内容に含まれている。

また、ハードウェアの面でも、汎用のマイクロプロセッサなど商用部品（COTS）を用いる方式をとり、このプロセッサとメモリを一組とし、高速の転送能力を有するプロセッサ間ネットワークでこれらを多数結合したMIMD型のアーキテクチャを持つ超並列・分散マシン構築の新技术に挑戦している。

これらの技術は、従来の大型汎用マシンからコストパフォーマンスにまさる大規模な並列マシンやスケーラブルな分散マシンへの移行を加速し、市場における米国の競争力強化に直結するとみられている。また、その研究途上においても、有用な派生的技術を生み出すことが多く、特にソフトウェアに関しては、並列処理に関する有効な新アルゴリズムなど多くの新技术が生み出されると思われる。

第3章 技術的展望と課題

3.1 アーキテクチャ

3.1.1 結合網の技術からペタフロップスマシンを考える (天野英晴委員)

3.1.1.1 はじめに

ペタフロップスマシンが 10^{15} FLOPSのピーク性能を実現する計算機のこと、しかも5～7年後程度を目処に実現を狙っているとすれば、大規模並列計算機のプロセッサ数を増やす方向以外に手はない。これは、いかに高性能マルチプロセッサの性能が上がり続けているとはいえ、5～7年間程度では、10倍を越える性能向上を見込むことができないからである。現在、高性能な浮動小数点演算機能付きマイクロプロセッサの性能は、1 GFLOPSつまり 10^9 に達していない。したがって、ペタフロップスマシンを実現するためには、将来、性能が向上することにより数GFLOPSとなった高性能マイクロプロセッサを、10万から100万台接続する必要がある。これは、今まで超並列計算機として考えられてきた1000-10000プロセッサから成る規模をさらに越えている。つまり、簡単に言うと、近未来にペタフロップスマシンの実現を目指すならば、問題は単純にプロセッサ数の増加ということになる。

さて、ペタフロップスマシンがGFLOPSプロセッサを10万から100万集めた超々並列計算機であるとすれば、その鍵を握るのはプロセッサ間を接続する結合網であろう。ところがこの規模のマシンに関して結合網の技術から考えると、決して面白くない結論が速やかに導かれる。結論のみを書くともあまりにも簡単すぎるし、規定原稿枚数に達しないと困るため、まず、現在の大規模並列計算機用の結合網の現状を簡単にまとめ、それからペタフロップスマシンについて論ずる。

3.1.1.2 現在の大規模並列計算機の結合網の現状

結合網で問題になるのは、トポロジすなわちつなぎ方、パケット転送技術および実装技術である。

トポロジ 従来、結合網の研究ではトポロジが重視されてきた。1980年代に流行したハイパーキューブ結合は、その配線量の多さから、高性能のマイクロプロセッサによる大規模

並列システムに適合することができず、80年代後半から、実マシンには単純な2次元／3次元メッシュ（トーラス）またはクロスバの組み合わせから成るFat Tree等の間接網が用いられる一方、研究レベルではさまざまな超並列向きの結合網が提案された。しかし、これらの超並列向きの結合網が実際に利用された例はほとんどなく、90年代に入ってからトポロジの研究は下火になり、現在でもその傾向は続いている。この、最も大きな原因は、高性能のマイクロプロセッサに見合うだけの高速なリンクを長く伸ばすことが技術的に難しくなったためである。複雑なトポロジにより、プロセッサ間の距離を短くすることによるメリットよりも、リンクを伸ばしたことによる転送性能の悪化とコストの増加のデメリットが上回ったと見てよい。例外は、日本の超並列計算機プロジェクトで、CP-PACSのハイパークロスバ、RWC-1のhMDCE、JUMP-1のRDTなど新しい結合網を積極的に用いており、一つの特徴と見てよい。

パケット転送技術 トポロジに代わって80年代後半から技術革新が起きたのはパケット転送技術である。初期の大規模並列計算機はパケットの転送をソフトウェアで行ったため、パケットの転送はハンドシェイクを伴う単純なStore-and-Forwardであり、他に手の打ちようがなかった。ところがLSI技術の進展により、プロセッサと独立したルータというハードウェアが作られるようになってから、ルータのハードウェアが用いるさまざまなパケット転送技術が発達した。

DallyらによるWormholeルーティングと仮想チャネルの技術の確立により、デッドロックの心配をすることなく結合網の性能を最大限利用可能になった。90年代に入ってパケットの転送技術の最後のテーマは、適応型ルーティングの技術の確立であった。適応型ルーティングとは、パケットの転送経路を固定せず、結合網の混雑を判断して迂回する技術である。90年代の前半、さまざまな方法が提案された末、最適経路に関してはDuatoの方法、迂回を許す場合はTurnモデルなど実装可能な方法が登場した。現在、パケット転送技術の研究は一息ついて、今まで提案された方法の中で実地的なものに関して、実際のルータチップを作って実マシン上でテストする段階に入ったと言っているようだ。

実装技術 トポロジの研究、パケット転送技術の研究が共に一息ついた現在、それらをいかに生かしてルータチップ上に実装するかが重要になってきている。そこで、ここでは多少詳しく最近のルータチップに関して紹介する。はじめにルータの機能と構造について説明する。典型的なルータは、図1に示すように、パケットを格納するバッファと交換するスイッチ、転送と交換を制御するコントローラから成る。

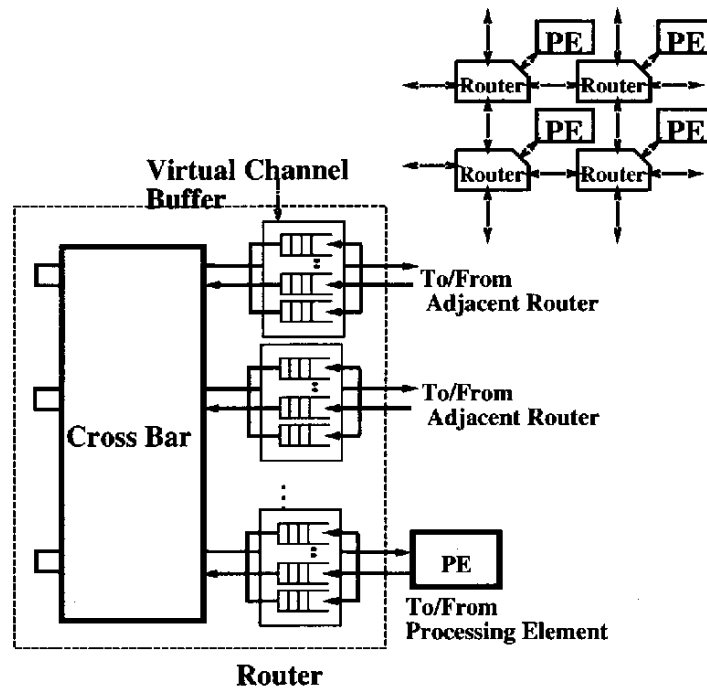


図1 ルータの構成

最近のルータは、パケットを効率良く転送するためWormholeルーティングまたはVirtual Cut throughと呼ばれる方法をとっている。また、リンクの利用効率を上げデッドロックを防ぐため、仮想チャネル(Virtual Channel:VC)を複数装備している。最近のルータのいくつかについて特徴をまとめて表1に示す。

表1 ルータチップの仕様

ルータ	T3E	Cavallino	RWC-1	JUMP-1
開発	Cray	Intel	RWCP	慶應大
トポロジ	3D Torus	2D Torus	hMDCE	RDT
ダイサイズ(mm)	16.7	13.5	13.6	
動作周波数(MHz)	75	200	50	60
転送レート(MHz)	375	200	100	60
リンクバンド幅(MB/s)	600	800	300	38
チャネル幅(bits)	14	16	24	18
使用ゲート数	81917		50000	90522
使用ピン数		350	409	299
テクノロジー(μ m)	0.35 CMOS	0.35 CMOS	0.7 CMOS	0.8 BiCMOS

T3E [1] はCray/SGIがT3Dに続いて開発した大規模並列型スーパーコンピュータ用ルータである。このルータの特徴は、簡単な故障迂回機構を内蔵していること、Duatoの方法を利用した最短経路上での適応型ルーティングを装備していることで、90年代のパケット転送技術研究の成果を積極的に利用している。一方で、 $0.35\mu\text{m}$ CMOS低電圧テクノロジーの利用により375MHzの転送周波数を実現しており、リンクの駆動も直接可能である。

Cavallinoルータ [2] は、T3Eのように凝った転送機能を持たない単純な構造を持つルータで、2次元メッシュの接続用の4本の物理リンクとプロセッサと接続するための2本のリンクを持っている。ルーティングアルゴリズムも単純なdimension-order routingである。

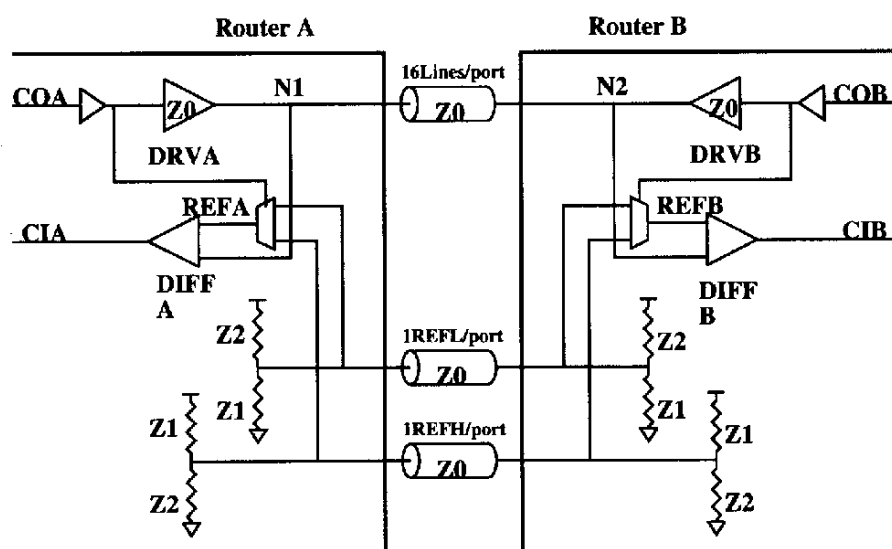


図2 双方向通信ブロック図

このルータの凄い所は、リンク間およびリンク、制御信号は、Bidirectional Signaling 法[3]により1本の線で全二重通信を実現している点である。図2のDRVA, DRV Bより送り出された信号は3レベルの電圧にエンコードされリンク上を伝わる。このリンク上の信号と自分が送り出した信号を用いてデコードすることにより相手が送り出した信号を取り出すことができる。両側から送られた信号は途中でぶつかるのだが、自分の出した信号は反転信号によりキャンセルされるので、相手が送った波形のみ取り出すことができる。

RWC-1 [4] のネットワークは、hMDCEと呼ばれる階層3次元直接網である。これは、x-y平面をc-Banyan(circular Banyan)、x-z平面をCCC(Cube-Connected Cycles)で結合したものである。hMDCEの各ノード部分はsub-networkノードと呼ばれ、4PEから構成されている。このルータは、RWC-1のアーキテクチャを反映した特殊な機能を持つ。まず、リアルタイム処理のマルチユーザ環境を実現するために、パーティション間で通信干渉の無い閉パーティション

を構成する事ができる。ルータは、パーティション制御のためのレジスタを持ち、パーティション外に出ようとするユーザパケットをエラー処理する(空間分割支援)。また、プロセス切り替えの際アプリケーションやOSの要請により、ルータ上に残っている未配送パケットをローカルメモリ内に保存、復活する機能も持つ(時間分割支援)。

JUMP-1 [5] ルータは、分散共有メモリ管理用にマルチキャスト機能と応答パケット収集機能を持つ点が特徴である。マルチキャストは、結合網RDT(Recursive Diagonal Torus)の階層構造を利用し、パケットヘッダ内のビットマップに応じて行われる。他のルータとの違いはBiCMOSテクノロジーを利用していることで、2m近いケーブルをECLレベルで直接駆動することができる。RDTはリンク数が多いことと、テクノロジーがやや古いため、転送性能は他のルータに比べて低い。

以上をまとめると、最近のルータは、

- 転送クロックは400MHz近くに達する。8-16bit幅なので、リンク当たりの転送容量は800MByte/secに達する。内部動作はそれより低く取ることが多い。
- 低電源CMOSが主流で、ケーブルを直接駆動する。
- それぞれのシステムの特徴に対応して、適応型ルーティング、OS支援機構、マルチキャスト機能などを持つ。

ということがわかる。

3.1.3.3 それではベタフロップスマシンは？

(1) ベタフロップスマシンはどんな構成になるか？

さて、結合網の点から現在の大規模並列計算機と、5-7年後の完成を目指したベタフロップスマシンを比較すると表2のようになる。

表2 現在の大規模並列計算機とベタフロップスマシン

技術項目	現在のマシン	ベタフロップスマシン
プロセッサ性能(GFLOPS)	0.1 - 1.0	1.0 - 10.0
プロセッサ数	100 - 1000	100000 - 1000000
リンク転送容量(MB/sec)	100 - 500	1000 - 5000

ここで、単体プロセッサ性能の性能向上に見あただけのリンク転送容量は、おそらく先に示した現在の最新鋭のルータ技術、すなわち低電源CMOSの延長線上で実現可能であろう。とはいえ転送クロック周波数は500MHzを越えることは実装が困難であるので、それから先はバンド幅、すなわち線の数を増やすことになる。Cavallinoルータが用いた双方向転送の技術はこのような状況では大変有効である。この周波数で転送を行うためには、一つの線上に連続していくつもパルスを送るWave Pipeliningの技術の利用が不可欠となる。

一方、100万プロセッサを実装する場合のスペースの問題はどうであろうか？1メートル四方の基板に100プロセッサを実装することは、MCM(Multi Chip Module)や1チップ上に複数のプロセッサを搭載する技術により可能になると思われる。その場合、この基板を $10 \times 10 \times 100$ の直方体状に配置すると、一辺が約10mとなり、高さは数m程度になる。この大きさは実現可能でないとはいえない。

ところが、この大きさになると、いかにWave Pipelining を効果的に用いたとしても、500 MHzでパケットが通るリンク（しかも1つのリンクの同軸ケーブル数は増やさなければならぬ）を、遠隔プロセッサに対して配線することはほとんど不可能といってよい。したがって、この規模になれば3次元Mesh/Torus以外、結合方式は考えられない。Torusのフィードバックループを心配する向きもあるが、これについては実装上の工夫 [6] や、マッピングの工夫 [7] で解決可能であるのでさほど心配は要らない。もちろん、効率良く動作するアプリケーションも隣接プロセッサ間でのみデータを交換する単純な陽解法的な数値計算に限定される。これは、PACSプロジェクトを指導した筑波大学の星野力教授、慶應大学の川合敏雄教授がかつて予想した世界である [8] [9]。すなわちペタフロップスマシンはスーパーPACSにならざるを得ない。

(2) ペタフロップスマシンは存在価値があるか？

ところで、このような巨大なマシンが一つのシステムとして実装されている意味はあるのだろうか？例えば、ペタフロップスマシン1台よりも、10TFLOPSマシン100台を巨大なバンド幅を持つ情報スーパーハイウェイで接続した方がいいのではないか？

10TFLOPSマシンを例えば $10 \times 10 \times 100$ の3次元Torusで実現した場合、境界プロセッサ数は、 $100 \times 4 + 1000 \times 2 = 2400$ となる。ペタフロップスマシンの一つのリンクは1000MB/secの転送容量を持っているので、これと同じプロセッサ間転送バンド幅を実現するためには、10TFLOPSマシン間を接続するスーパーハイウェイは、2400GByte/secの転送容量を必要とすることになる。これは、いかに光転送技術と大容量の交換技術が発達しても実現は容易ではなく、まして既存のネットワークを利用するということは現実的ではない。このように、ペタフロップスマシンは、技術的には一つのシステムとしてまとまっている価値が十分存在するのである。

3.1.1.4 ではなにをすれば良いか？

(1) 次に何を調査すべきか？

今まで検討したシナリオに基づけば、ペタフロップスマシンはできる限りシンプルな構造が良い。したがって、計算機アーキテクチャ屋のやるべきことは、耐故障性の検討以外全くないといってよい。ちなみに、OS屋のやるべきこともほとんどない。2次元/3次元Torusの耐故障性に関する検討は、すでに多くの研究が行われており、この中から現実的な方法を見つければよい。したがって、以降、この委員会の調査活動は、アプリケーション、記述法、実装技術、耐故障性の検討に専念し、とりあえず計算機アーキテクトには全員お引き取りを願うのがよい。筆者も含め計算機アーキテクトは、その研究が飯の種であるため、マシンを複雑にする傾向にあり、ペタフロップスマシンの研究開発には害の方が大きい。PACSやGRAPEの開発が計算機の専門家によらなかった点を考慮すべきである。

(2) わが国として取り組むべき課題

上記の議論から、結論として、わが国としては軍事予算を大幅に削って、ペタフロップスマシン＝スーパーPACSの開発を迷うことなく即座に開始すべきである。軍事予算で開発するのは、核兵器の実験が禁止される以上、実験に代わり得る計算を行うことのできるペタフロップスマシンこそが最大の軍事的抑止力になるという考え方に基づいている。（この考えを筆者自身は信じていない。実験せずに計算だけで創り出せるものなどあるわけがない。しかしもはや現在の日本には他に資金のあてはないし、ペタフロップスマシン＝軍事抑止力という考え方は案外真面目に信じられているようなので利用しない手はない）。PACS型の計算機は日本独自の考え方に基づいており、Originalityという点で他国からクレームが出る心配は全くない。将来にわたっても商売にならないから、貿易摩擦も心配ない。これを日本が始めれば、米国をはじめとして各国も競って軍事費を削って（他国だってお金はないので）ペタフロップスマシンの開発を始めざるを得ないであろう。このことにより、破壊と殺人以外使い道のない兵器を開発するお金を使って、地球の温暖化対策をはじめとして、我々がこれから地球上に生き残っていくために必要な研究に使う道具を得ることができるのである。

<参考文献>

- [1] Steven L. Scott and Gregory M. Thorson: "The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus" Hot Interconnect Symposium IV, p.147-156, 1996
- [2] Joseph Carbonaro, Frank Verhoorn: "Cavallino: The Teraflops Router and NIC" Hot Interconnect Symposium IV, p.157-160, 1996
- [3] Randy Mooney, Charles Dike, Shekhar Borkar: "A 900Mb/s Bidirectional Signaling Scheme"

IEEE International Solid-State Circuits Conference, P38-39, 1995

- [4] Takashi Yokota, Hiroshi Matsuoka, Kazuaki Okamoto, Hideo Hirono, and Shuichi Sakai: "A High-Performance Router Design for the Massively Parallel Computer RWC-1" Hot Interconnect Symposium IV, p.1-12, 1996
- [5] Hiroaki Nishi, Katsunobu Nishimura, Ken-ichiro Anjo, Tomohiro Kudoh, and Hideharu Amano: "The JUMP-1 Router Chip: A versatile router for supporting a distributed shared memory" International phoenix Conference on Computers and Communications, 1996.
- [6] 田辺、小柳、"三次元実装に基づくマルチパラダイム超並列テラフロップスマシンのアーキテクチャ," 信学報 CPSY92-24, 1992.
- [7] 村田、天野、"数値計算における疎密のある物理空間に対応するマッピング：ローリングマッピング," 情処論文誌 Vol.35, NO.2, pp.332-341, 1994.
- [8] 川合敏雄、"スーパーコンピュータへの挑戦" 岩波書店、1985.
- [9] 星野力、"PAXコンピュータ -高並列処理と科学計算," オーム社、1988.

3.1.2 ペタフロップスマシン実現に立ちはだかるもの (稲上泰弘委員)

1980年代にベクトルプロセッサ型スーパーコンピュータが普及し、超高性能計算需要を喚起し加速させた。日本のスーパーコンピュータメーカーは、汎用計算機で培った超高速バイポーラLSI技術を活用し、世界最高性能のシステムを市場投入し続け、ギガフロップスマシンを実現した。ハードウェア技術、アーキテクチャ技術、ソフトウェア技術の全般にわたり世界をリードしてきた。

1990年代に入り並列プロセッサへの期待が高まり、1990年代半ばを過ぎた現在、性能が1～2 GFLOPSのベクトルプロセッサを数百台規模で結合した並列ベクトルプロセッサや、300～500MFLOPSのスーパースカラプロセッサを1000台規模で結合した超並列プロセッサが実用化され、テラフロップスマシンの時代に突入した。使用されるハードウェア技術はCMOSテクノロジーが主流となり、ここにおいても世界をリードするポジションにある。[1]

超高速計算が必要な最先端の計算科学分野ではさらに高速なスーパーコンピュータが必要とされ、2000年代にはペタフロップスマシンに向けた新たな技術開発が望まれている。一方、スーパーコンピュータ市場は頭打ち（横ばい）との予測が一般的で、これまで性能最優先で進められてきた技術開発のあり方に疑問を投げかける議論も出ている。ペタフロップスマシンの実現に必要な技術が、最先端の計算科学研究者にとどまらず、広い範囲の研究者・技術者にとって役に立つ計算機に広く応用されることが必要なことを示唆している。

本節では、ペタフロップスマシンを実現するための重要課題について考察する。

ペタフロップスマシンの実現に並列プロセッサ技術は不可欠であるが、一つの問題を並列処理した場合の処理時間の内訳は次のようになる。

逐次処理	データ転送（並列処理可能）	計算処理（並列処理可能）
------	---------------	--------------

すなわち、並列処理ができない逐次処理部分の実行時間、多くの場合並列処理が可能なプロセッサ間データ転送時間およびプロセッサでの計算処理時間である。高い並列処理性能を実現するには、アルゴリズム／プログラミング手法の工夫、アーキテクチャ／論理方式の工夫などハード、ソフト両面の高い技術が必要である。アルゴリズム／プログラミング手法の工夫としては、逐次処理時間をほとんどゼロにすること、計算処理とデータ転送処理をできるだけオーバーラップさせ実効的なデータ転送時間を少なくすること、計算処理の並列度を高めることがある。アーキテクチャ／論理方式面では、単体プロセッサ性能の向上、低レーテンシ・高スループットのプロセッサ間データ転送機構の実現が鍵となる。ここでは、アーキテクチャ／論理方式に関連する課題について考察する。

1 ペタフロップスマシンの実現を考える場合、どの位の性能を持つプロセッサを何台程度

結合するかが第一の課題となり、実現可能かつ広い応用で実用可能な範囲で決定する必要がある。表1に、1ペタフロップスマシンの実現を考えた場合の、プロセッサ台数とプロセッサ単体性能との組み合わせ例を示す。以下、表に示した3組のアプローチの実現性について考える。

表1 1ペタフロップスマシン実現のアプローチ

項番	プロセッサ台数	プロセッサの単体演算性能	プロセッサ性能実現のために必要な動作周波数、命令/演算実行並列度	
1	1,000	1 TFLOPS	2 GHz	500
2	10,000	100 GFLOPS	2 GHz	50
3	100,000	10 GFLOPS	2 GHz	5

(1) プロセッサ台数

現在実運用されている最も性能の高いシステムの並列度は1000台規模である。1000台規模のシステムを高い稼働率で使うことは決して簡単なことではなく、かなりの工夫が必要とされている。今後、並列アルゴリズム、並列プログラミング技術の発展を期待するとしても、実用レベルで使用できる並列度は現状の一桁程度上が限界と考えるのが妥当である。したがって、実用的なペタフロップスマシンのプロセッサ並列度は高々10,000台規模である。表1に示した3つの組み合わせの中で、項番3は特定の応用を除けば採用できない方式と考えざるを得ない。

(2) プロセッサ単体演算性能

プロセッサ単体の演算性能は、プロセッサの動作周波数および命令/演算の実行並列度で決まる。最近の高速プロセッサの動作周波数は500MHz前後まで向上し、今後も高速化が図られると予想される。今後10年の間に現状の4倍の2GHz程度は実現されると期待したい。命令/演算の実行方式はスーパスカラが主流であり現在2ないし4の並列度が一般的である。VLIWなどの新しいアーキテクチャの導入により、今後10年の間に少なくとも16ないし32の命令/演算の並列実行方式が実用化されと考えられる。しかしながら、1プロセッサにおける数百レベルの命令/演算の並列実行は、プロセッサチップ構成方式、コンパイラ方式いずれの観点からも実用性が低いと言わざるを得ない。したがって、表1・項番1に示したプロセッサ単体性能1TFLOPSの実現は、用途を限定しない限り実現性は低いといえる。

(3) ペタフロップスマシン構成のガイドライン

以上のことより、今後10年以内にペタフロップス級のマシンを実現する場合、プロセッサ

並列度は数千から1万台規模、プロセッサ単体の性能は高々100GFLOPS程度とする構成方式が妥当である。

(4) メモリ構成方式

プロセッサ単体の性能は、命令／演算の並列実行機構だけでなく、メモリとレジスタ／命令実行機構との間のデータ供給能力(レーテンシ、スループット)にも影響される。レーテンシはキャッシュ機構を導入することにより解決してきており、今後も技術開発が進められる。スループットについては、キャッシュをプロセッサチップに外付けする場合はLSIピン数の制約を受けるが、今後はLSIの高集積化に支えられたロジック・メモリ混載LSI技術の進展により、チップ内で高スループットのキャッシュの実現が可能と考えられる。

超高性能計算機を必要とする応用の中には、プログラムのワーキングセットが大きく、キャッシュが有効に働かないものがある。ベクトルプロセッサなどでは、キャッシュ機構を使わず、レジスタと主記憶との間で直接データのやり取りを行い、かつアクセスリクエストの先行制御、データ転送のパイプライン処理により性能を高めている。ベクトルプロセッサの場合、1FLOPS当たり4～16B/sのスループットを実現して高い実効性能を維持してきているが、ペタフロップスマシンのようにプロセッサ単体性能が100GFLOPSオーダになると、数百GB/s～TB/sのスループットをプロセッサチップとメモリチップ間で実現する必要がある、実現性・コスト面で無理が生じる。ここにおいてもロジック・メモリ混載LSIの活用によるチップ内高スループット主記憶を基本とする必要があるが、容量面で制約が生じ、プロセッサ性能、メモリ性能、メモリ容量のバランスを考慮した並列アーキテクチャの追求が重要となる。しかしながら、このような方式を広い範囲の応用に適用することは難しいと考えられ、基本的には、ロジック・メモリ混載LSI技術を念頭においたキャッシュ方式、階層記憶方式を追求する必要がある。

(5) プロセッサ間結合方式

並列プロセッサの稼働率を高めるには、全実行時間に占めるプロセッサ間データ転送時間の割合を少なくする必要がある、アーキテクチャ／論理方式面では、低レーテンシ・高スループットのプロセッサ間結合方式の実現が重要である。

現状システムのプロセッサ間データ転送レーテンシは、通信ライブラリMPIのレベルで考えると10～50 μ s程度である。一方、プロセッサの演算性能は、数百MFLOPSないし数GFLOPSである。テラフロップスマシンにおいても、演算性能とプロセッサ間データ転送性能のバランスが相似的に拡大できれば、現在使われているアルゴリズムや並列化手法をそのまま拡大するだけでテラフロップスに到達することができる。レーテンシは、データ転送に伴うプロトコル処理などを改善していくと、最終的には装置の規模に依存する。ペタフロッ

プスを実現するシステムのプロセッサ台数は数千から10,000台規模となり、現在実用になっているシステムより1桁多い台数である。ハードウェア技術の進展を勘案しても装置規模が劇的に小さくなることは考えにくい。したがって、レーテンシの大きな削減は期待できず、演算性能とのバランスにおいては、現状より1～2桁悪くなる。すなわち、ユーザにとって並列化が難しくなる方向にシステムバランスが移り、全体処理時間に占める実効的なデータ転送時間の割合を現状より1～2桁小さくする並列化アルゴリズム／手法の開発が不可欠となる。

スループットについては、現在の代表的なシステムのシステムバランスは、富士通VP700の場合、プロセッサ単体性能が2.4GFLOPSに対してプロセッサ間データ転送スループット約1GB/s、CRAY T3Eが0.6GFLOPSに対して0.6GB/s、筑波大学のCP-PACSが0.3GFLOPSに対して0.3GB/sである。前述のように、テラフロップスマシンのプロセッサ単体性能はおよそ100GFLOPSであるから、現在の並列化アルゴリズム／プログラミング手法の継承を考えるなら、100GB/s程度のプロセッサ間データ転送スループットが必要になる。プロセッサ間結合ネットワークの動作周波数をプロセッサ動作周波数の約半分の1GHzと仮定すると、100GB/sを実現するには、数千から10,000台規模のプロセッサ間に100バイト程度のデータ線を張り巡らせる必要があり、非現実的である。プロセッサ間のデータ線幅は高々2～8バイトと考えられるので、演算量に占めるデータ転送量の割合を、現状に比べ1～2桁押さえる技術が必要となる。

以上をまとめると、ペタフロップスマシンを実用的なシステムとして実現するには、10,000台規模のプロセッサ並列を有効に使う並列処理技術のほか、演算性能とメモリ性能のギャップを埋める手段としてロジック・メモリ混載LSI技術の積極的活用、さらに、演算量に占めるデータ転送量の比率を現状より1～2桁程度小さくする並列アルゴリズム／並列化プログラミング技術の開発が必要である。このうち、並列アルゴリズム／並列化プログラミング技術の開発が特に難しい課題と考えられ、並列処理ソフトウェア研究の強化がペタフロップスマシン実現にまず必要な施策と考えられる。

<参考文献>

- [1] J. Dongarra Performance of Various Computers Using Standard Linear Equations Software. March 15, 1997.

3.1.3 ペタフロップスマシンの展望 (久門耕一委員)

3.1.3.1 ペタフロップスマシンに対する日米の差

日米でのペタフロップスマシンに対する大きな差は、実現に対するニーズの差である。米国では、核実験の計算機シミュレーションの必要性からテラフロップスマシンが実現された。米国としては、計算機上での核実験シミュレーションが可能になれば、費用削減、世論対策などの多くのメリットがある。また、そこで作られた技術を産業界へ転用することができるので、かつての軍事主導型の技術開発がうまく機能するものと考えられる。

一方、日本においては技術革新は主に産業界において進められてきたが、現時点ではバブル崩壊後の投資の冷え込みにより巨大システム構築のための余力がない。また、大規模システム開発は半導体産業ですら企業共同体でなければ開発が行えない状況になっており、また、多くの場合、日本企業同士の連合体を作りやすい風土はない。また、テラフロップスマシンをいくらで実現でき、結局いくら儲かるのかという費用対利益の構図がはっきりしないと企業連合が作成されることはないだろう。

3.1.3.2 将来予想と解決策

ペタ (P) フロップス (flops) マシンの実現性の検討を行う際、コストの検討を避けて通ることはできない。もちろん、コスト以前の問題としてどのような応用に使用し、どのような利益があるのかを定義し、そのために必要な費用・構造を検討することが第一に必要なことはあるが、ここではあえてそれを避け、現在作成可能と思われるアーキテクチャとその問題点について、実現面からの検討を行う。

(1) ペタフロップスマシンのハードウェア

単純に、単体性能に台数を掛けてペタフロップスを達成できるマシンを作ることさえ現在使える技術では困難が多い。通常は、実性能を求める立場から、単体性能を極限まであげ、それを必要最小限に組み合わせると言う解がもっともらしいが、ここではそれに因われず、無闇と単体性能を向上させる方法と無闇に数で稼ぐ方法の2つを考えてみる。

単体性能を可能な限り向上させる方法 Peta (10^{15}) Flopsマシンを今後10年程度で作製するためには、

表1 P flopsを達成するために必要な単体性能と台数

単体性能	台数
100M Flops	千万CPU
1G Flops	百万CPU
10G Flops	十万CPU

が必要になる。現在、マイクロプロセッサによる単体 CPU の性能はすでに 500Mflops を越えており、10G flops 以上は見込めるとしよう。つまり、台数として10万台以下を考えれば良い。

10万プロセッサを接続する場合に、実装面から見て使用可能なネットワークポロジの選択肢は、それほど多いわけではなく、プロセッサの配置に素直に従う3次元メッシュやその亜流しか選べないように思える。

10万はおよそ50×50×50なので、縦横高さ方向に50ノード繋げると考えれば、費用さえ厭わなければ実現可能だろう。

接続CPUを可能な限り増加させる方法 現在のLSIの規模は1チップで1,000万Tr (10M Tr) 以上が可能である。複雑な制御を行わないのであれば、1CPUは1MTrもあれば100 Mflops級のプロセッサが作製可能だと信じることにする。つまり、現状でも単純計算で言えば1チップあたり10 CPU 程度集積することは全く可能である。しかし、このような構成のCPUを作成しても、メモリとの通信、IOとの通信がピン数、速度共にネックとなり、CPUチップが外部と通信するためのスループットが向上せず、結果的に性能が得られない。

(2) ペタフロップス・マシンの可能な構成法

従来は、計算機能をCPUに集中させ、大容量の記憶装置を周辺に配置する形でのシステムが多く作られてきた。並列計算機であっても、現段階で商用化されているものは、性能の高い単体ユニットをネットワークにより複数接続し並列化することで構成されている。このアーキテクチャのコスト的欠点とでも言うべきものは、CPUの稼働率が高いがメモリの稼働率はかなり低いということである。ここで言う稼働率は、あるビットやチップがアクセスされる率である。現実にある大規模な計算機システムにおいては、CPUよりもメモリやI/Oのコストがシステムコストの大半を占めている。特に、CPUとして汎用マイクロプロセッサを用いる場合はその傾向が大きい。すなわち、計算機システムの大半のコストはあまり利用されない部分にかかっていると言える。

このことを現在ありふれていそうなシステムを基準に検証してみる。

ここでは、ハイコストパフォーマンスなシステムとしてインテルPentium-Proを使用したシステムを考える。

表2 1997年におけるPCシステムのユニットあたりのコスト

(コストは1997年初頭の市中価格による)

ユニット	速度	コスト(円)
Pentium-PRO	600M inst./s 200M load or store /s 256K Cache	70k/CPU
P6-Bus	530MB/s 66MHz 8Byte	
DRAM	60ns/access	500k/1GB

現在のPentium-PROは、200MHzのチップ内クロックで動作するSuper scalarプロセッサで、連続的に3命令ずつ実行する事ができる。このことから、プロセッサ自身は最大で600M命令/秒の実行が可能である。load/storeに限ってみても、毎クロック実行が可能であるから、200Mアクセス/秒のメモリアクセスが可能である。一方Pentium-PROが採用するP6バスは最大スループット約530MB/sである。これは、8byte幅で66MHzのクロック速度により実現されている。この速度がPentium-PROと外部とのアクセス速度の上限となっている。また、現在使用されているメモリのアクセス速度は60ns程度のものであり、この段階で66MHzのバスクロックとの間に4倍程度の速度比が存在する。

これらから、CPUが最高の性能を発揮するためのバス速度を達成するためには、メモリを4wayインターリーブしなければならない。これを、64MbitDRAM(16Mbit×4)のチップで構成する場合には8byteバスにするため、16chip、4wayインターリーブのために4倍で合計64chip、すなわち、512MBのメモリが搭載されることになる。

表3 本仮想計算機システムのユニットあたりコスト

ユニット	コスト(円)
Pentium-PRO	70k
P6-Bus	--
DRAM	256k

この単純な検討からもCPUのコストよりもメモリのコストが3.5倍も高価であることがわかる。それぞれの機能を実現するために必要なチップ面積を考えてもメモリが占めるチップ面

積の方がはるかに大きいことがわかる。

次に、このすべてのメモリをCPUがアクセスするために必要な時間を算定する。512MB/(520MB/s) からCPUが全メモリにアクセスするには少なくとも約1秒が必要である。すなわち、最高速度ですべてのメモリを順番にアクセスするようなプログラムでさえ、あるビットを平均1秒1回しかアクセスすることができない。メモリは大変に暇であるにも関わらずシステムコストの大きな部分が占められていることがわかる。システム性能に対するメモリの遅さを隠蔽するために用いられるキャッシュメモリを大きくすれば、実効性能は向上するがCPUのコストにそのまま影響してくる。

計算機システムがこのような構成を取るのは、

- 並列プログラミングが困難であり、一般にCPU台数に見合う性能向上を得ることができない。
- 単一のCPUの性能向上を計ることがシステム性能向上への近道、

であるため、単一のCPUの性能を最大限に生かす周辺構造を持たせることが必要だからだと考えられる。

一方、Peta flopsマシンを考えると並列プログラミングの必要性は疑う余地はない。しかも、ノード数は少なく見ても10万程度は必要であるから、単体性能を向上して数を減らすというアプローチができるレベルではない。つまり、単体性能の無闇な向上はあきらめ、とにかく数を増やすことにより速度の向上を図ることが残された手段である。

さらに、単体nodeのメモリ価格を抑えながら実質的なメモリ量を増やすために、図1の構造を取ると考えよう。

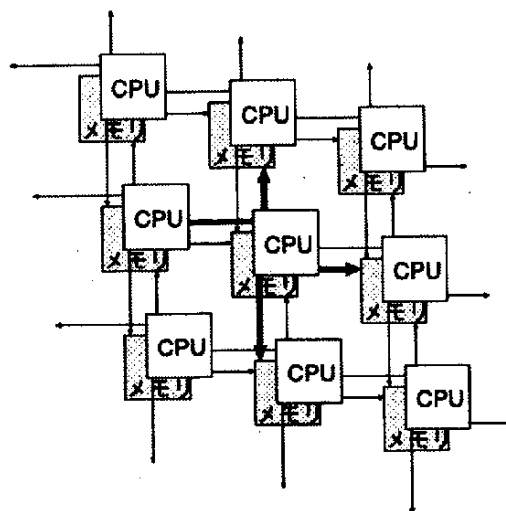


図1 二次元メッシュ部分共有メモリ

ここでは二次元メッシュ構造を示した。

この構造では各CPUはローカルにメモリを持つと同時に周囲のCPUのメモリへのアクセスも可能で、図1では、各メモリは計5ノードからのアクセスが可能である。この構造は、現実の物理現象が近接作用に基づくという前提に基づいている。実際には、三次元現象のシミュレーションへの適合性から三次元メッシュ構造が必要となる。現在の計算機システムではメモリのチップ面積とCPUのチップ面積の比率を見ると圧倒的にメモリの面積が大きい。しかし、上記の構造を取ってCPUの面積をメモリのチップ面積と互角にしても所詮はメモリのチップ面積が倍に増えるに過ぎない。

このようにして、1チップで1 G bit RAMと1 G flops CPUのペアを16個実装することが可能になるとして、このチップを65536チップ実装することで、計算上は1 Pflopsが達成できる。アプリケーションをどのように実現するのが最大の問題として残る。

実装上、このような構造で三次元メッシュを作製した場合、実世界の三次元構造をCPUのメッシュにマッピングするのが自然である。ここで、立方体の体積は辺の3乗、表面積は辺の2乗に比例するが、計算量は体積に比例し、通信量は表面積に比例すると考えられるため、辺を大きくするほど通信量が相対的に減少する。すなわち、通信路に対する負担が減るはずである。このことは、1つのnodeで大きな体積分の演算を行えば、辺の長さに逆比例して必要な通信速度が低下することを示している。

このチップ16個程度Multi Chip Module (MCM) により実装すれば、そのMCMを4096使用することで計65536CPUとなる。この段階まではチップ内と同じ通信速度が達成できると仮定する。熱設計の問題もあり、MCM間の結合は物理的に遠くせざるを得ない。MCMには256個の1 G flopsプロセッサ、つまり計256G flopsが搭載できる。

このように構成されたMCMを通信路により結合するが、結合路として利用できそうなバンド幅は恐らく10G~100G B/sだろうと考えられる。これほど高速な通信は、シリアル通信で行うよりもパラレル通信を使う方が実現が容易であるが、当然線間のスキューの補償が必要である。

ここで問題となるのは、このような技術がアナログ的に行われるが、最近の傾向として大学などでもアナログハードウェアに対する比重が少なく、また企業内においてもアナログの技術者が欠けているということである。日本は、高速通信分野で世界のトップに位置しているが、このような高速通信のハードウェアは単品生産的に製造されるものが多い。しかし、ペタフロップスマシン実現のためには、このような技術をLSIあるいはMCMの中に搭載し大量に安定して製造する技術が求められている。すなわち、超高速アナログ技術を含んだLSI技術が必要である。同様に、LSIのチップ内の配線においても数百MHzを越える場合には、デジタルとはみなせないため、高速アナログ回路の技術が必要になる。残念ながら、最近の技術者でアナログ技術を習得している人はほとんどいない。シリコンバレーでも、論理設計

だけでなく、Trレベルでの回路設計をこなして、Spiceシミュレーションにより自作のマクロの検証まで行う設計技術者の給与は、その雇主の社長の給料をも凌ぐと言われている。だから、日本では、技術者がいないのは仕方ないのかも知れない。このような技術に対するニーズは増えるが技術者が減っているのが現状であるから、大学教育の段階で何とか手を打っておく必要がある。

(3) マシンを何に使う

一般に、計算機の利用形態として以下の2つが考えられる。

高速電卓 すなわち技術者が考えたアイデアをシミュレーションにより確認するために計算機を用いる。各種のシミュレーションがこの目的である。

高機能電卓 計算機の膨大な探索能力を用いることで、最適解を求めるものである。

高機能電卓実現のためには高速電卓を用いたアイデア検証の積み重ねが必要であるため、高速→高機能の順に应用が開発されるものと考えられる。

高速電卓として使用する場合、何を入力として与え何が出来として得たいのかが重要なポイントとなる。マシンのメモリをすべて埋めるようなデータを入力できるわけではなく、出力も同様にできない。入力も、出力もある種の統計情報（例えば温度とか密度など）として与えることが現実的な解だろう。そういう意味では、入出力に高速性が求められるのは、中間結果の保存の場合だと考えられる。それなら、各ノードの結果を独立に保存すればよいので、大量のディスクをいかに結合・実装するかが焦点であり、並列ファイルシステムなどを考えなくてもよい。

今すぐにでもペタフロップスマシンを必要とする分野としては、材料分野でのデバイスシミュレーションが挙げられる。しかし、汎用高速計算機として使用することは困難で、アーキテクチャに適合できるシミュレーションアルゴリズムのみが計算の対象になる。(2)で述べた構造は、現実の三次元構造を各CPUにマッピングすることを念頭においているため、現実世界のシミュレーションを行うにあたってアルゴリズムを考えることが可能だと思われる。

3.1.3.3 国としてとり組むべき課題

3.1.3.1でも述べたように、現状では産業界が独自にペタフロップスマシンを開発することは考えにくく、作るとすれば国が音頭を取る以外にないだろう。例えば、材料関係などの研究機関からの要望をまとめ、ニーズを実現するためのペタフロップスマシンを共同利用施設

として実現することが必要だろう。その場合、ペタフロップスマシンそのものが課題となることはあり得ず、ペタフロップスマシンによって解かなければならない問題の方にこそ課題がある。

繰り返しになるが、このような巨大計算機システムのアーキテクチャはそれほどの自由度があるとも思えない。実現できるアーキテクチャ間での性能の差は高々数倍以内に留まるのではないだろうか。だとすれば、アーキテクチャの検討よりは、マシン実現により何が解決可能な問題なのかを分離し、それぞれの分野でどのように問題を定式化すれば狭い選択肢のアーキテクチャで並列処理が可能になるのかを検討することがマシン応用への最短経路だと考える。

3.1.4 メモリ・アーキテクチャの展望と課題 (中島浩委員)

本節ではPeta-Flops Machine (あるいはそれに相当する超大規模並列計算機) のアーキテクチャについて、メモリ機能を中心に議論する。また、超大規模並列計算機の研究開発に関する議論も行う。

3.1.4.1 メモリ・アーキテクチャ

まず以下の3点を議論の前提とする。

(1) 超大容量メモリの必要性

想像を絶するようなパラダイム・シフトがない限り、Peta-FLOPS Machineにおいてもメモリは存在するものと予想される。また仮に「MIPS=MBの法則」が引続き成り立つとすると、その総容量はPeta-Byteのオーダーでなければならない。

(2) 演算／メモリ・スループットのバランス確保

Peta-FLOPSの性能を実際に得るためには、それに見合うだけのメモリ・スループットを確保する必要があるだろう。したがってPeta-FLOPS Machineは、同時にPeta-MOPS (Memory Operation Per Second) Machineでなければならない。

(3) von Neumann bottleneckの存在

Peta-FLOPS Machineが現状のようなMulti von Neumann Machineであるかは不明であるが、広義のvon Neumann bottleneck は解消できないものと予想される。すなわち、素子レベルのメモリ技術に革命的なブレイク・スルーがない限り、Peta-Byteオーダーのメモリと、それをアクセスするプロセッサ (あるいは演算器) との間に、物理的な隘路が生じるのは避けられないだろう。また、現状の技術の単純な延長を考えると、このbottleneckは相対的により細いものとなるのは明らかである。

上記の前提に基づけば、使い古された言葉ではあるが「von Neumann bottleneckの解消／緩和」が、Peta-FLOPS Machineのメモリ・アーキテクチャの鍵となると考えられる。すなわち、現状ではこのbottleneckをアドレスやデータが通過しているが、これをより通過しやすい形に変質させることを考える。

以下、「bottleneck通過物の変質」の候補として、プロセッサからメモリへの機能移動を取り上げて議論する。

3.1.4.2 Visible Memory System

メモリがプログラムから可視であることは当然であるが、メモリ・システムのすべての要素が可視であるとは限らない。その代表例は現状のキャッシュであり、その挙動をプログラムが制御するのは極めて困難である。

例えばキャッシュが前提としている時間的／空間的局所性が成り立たないようなプログラムでは、不要なキャッシングやキャッシュの汚染が生じるが、これを避けるのは困難である場合が多い。また、共有メモリ型の並列計算機で行われるキャッシュの一貫性管理は、局所性の前提に加えて共有データのアクセス特性に関する前提があるため、キャッシュが無効あるいは有害となる可能性はさらに高くなる。

このような問題を解決する方法として、キャッシュをはじめとするメモリ・システムの要素を可視化して、何らかの制御を可能とするようなアーキテクチャがいくつか提案されている。例えばJUMP-1 [1] (図1)では、さまざまな一貫性管理や高度なメモリ機能を用意し、ソフトウェアが適切なものを選択できるようにしている。またソフトウェアによる一貫性管理に関するさまざまな研究も行われており [2]、我々の研究でも適切なハードウェア・サポートがあればハードウェアのみによる一貫性管理よりもよい性能が得られることを示している [3]。

この他、アクセスするデータの種類によって、時間的局所性を重視した通常のキャッシュと、空間的局所性や（データではなく）操作の時間的局所性を利用するバッファ的なキャッシュとを併用する試みもあり、例えば我々が提案しているVirtual Queueもその一つである (図2) [4]。またHewlett Packard社のPA-RISCの一部に採用されているAssist Cacheも、ある意味ではアクセス特性によってキャッシュを使い分ける方式である。

これらの、キャッシュをはじめとするメモリ・システムの要素の可視化における重要な問題点として、ソフトウェアとのインタフェースの設定があげられる。単純な方法としてはメモリ・システムを制御するための命令の追加があり、すでにプリフェッチなどを命令セットに加えているプロセッサは多い。別の方法としては、JUMP-1のようにアドレスの一部を制御に使うというものもある。いずれにせよ、均一であることが大きな特徴であったメモリ・システムを非均一にし、しかも相当な期間にわたって安定したインタフェースを設定するのは、必ずしも容易ではない。したがって、可視であるような要素を含む非均一なメモリ・システムの統一的ビューを設定し、ソフトウェアが安定的に利用できるようにすることが重要な課題である。

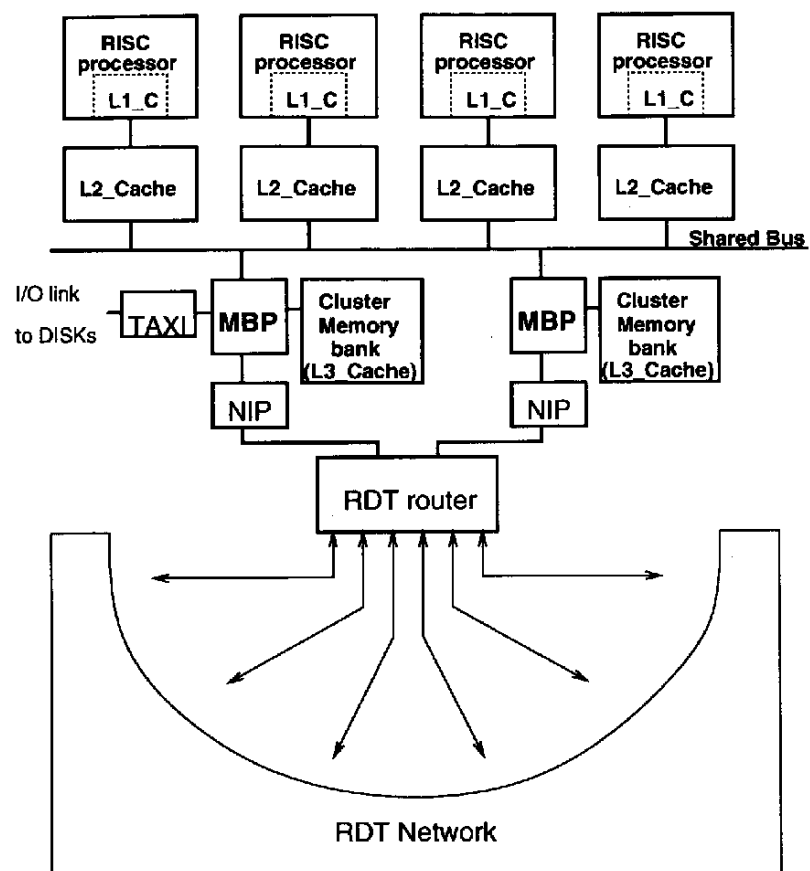


図1 JUMP-1のクラスター [1]

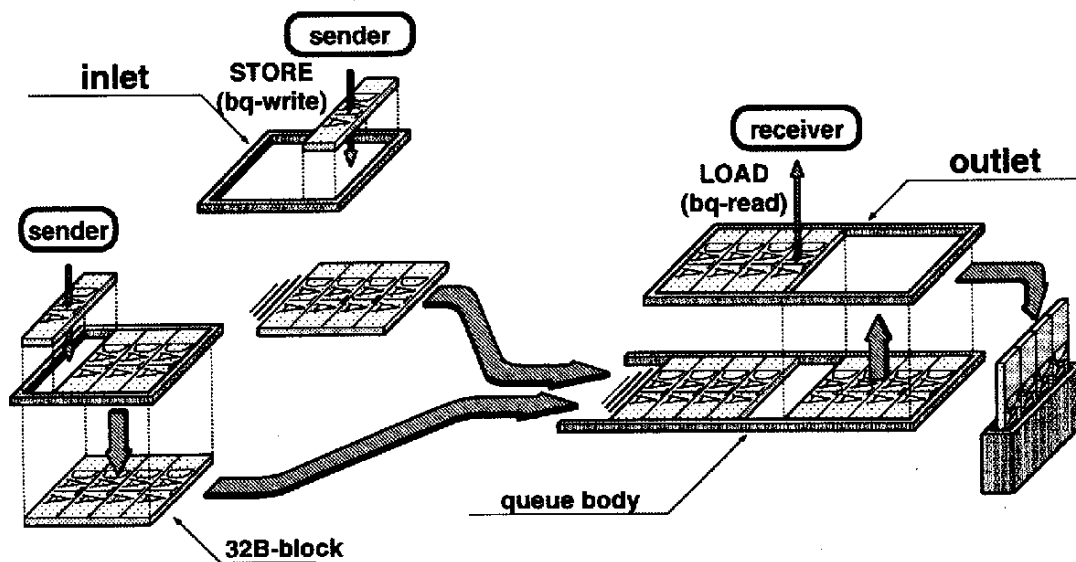


図2 Virtual Queue [4]

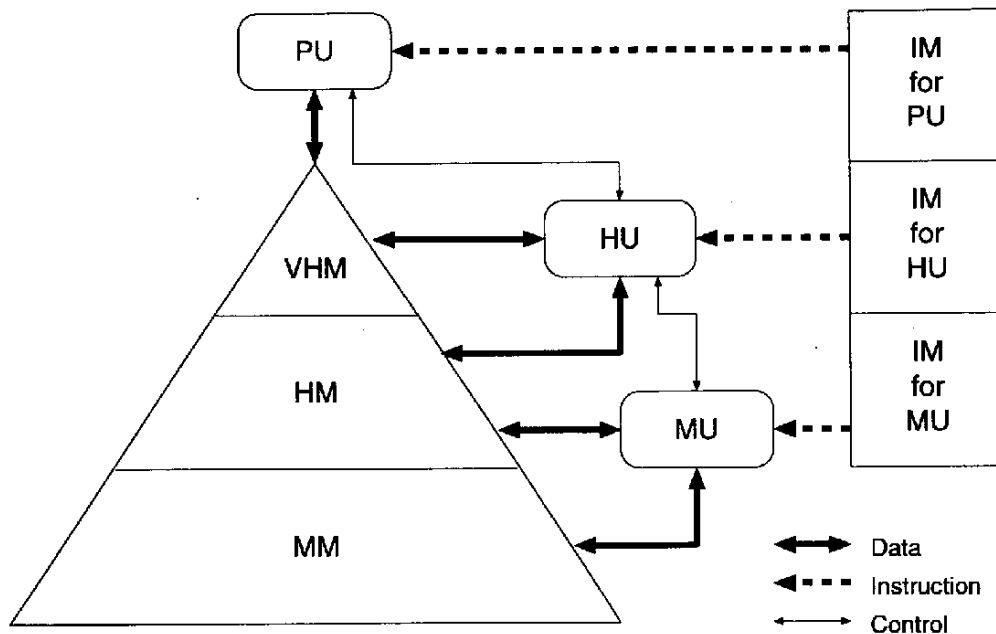


図3 UPCHMS [7]

3.1.4.3 Memory with Program

前項の可視化は、あくまでプロセッサが制御の主体となるものであるが、制御をメモリに移管してプログラムの実行をメモリが分担する試みもなされている。例えば、JUMP-1にも採用されている松本らのMBP [5] や、KuskinとGuptaらのFLASH [6] はその例であり、メモリに密接に結合されたプロセッサにより一定のメモリ機能をプログラマブルに実現するものである。

さらに先鋭的なアプローチとしては、牧らによるUPCHMS [7] がある(図3)。これはメモリ階層のそれぞれに独立して動作するプロセッサを配し、階層間のデータ転送を全てプログラム制御しようというアーキテクチャである。この方式には、階層間の制御情報を適切なタイミングで受け渡す方法や、階層ごとのプログラムをいかに生成するなど、さまざまな問題点があるが、メモリ・アクセスの新たなパラダイムを見出す試みの一つとして興味を持たれるアプローチである。

これらの方法にはいずれも、適切なタイミングでメモリをアクセスするための情報が、アドレスの列よりも十分に小さくできる、という共通の基盤があり、この基盤から新たなメモリ・アーキテクチャを生み出せる可能性がある。

3.1.4.4 Code Flow Model

前項の制御／機能のメモリへの移管をさらに押し進めると、以下のような新たなアーキテクチャを想起することができる。

- (1) メモリ上の全てのオブジェクトは固有のプロセッサを持つ。
- (2) あるオブジェクトをアクセスするプログラム・コードと、その実行に必要な（比較的小さな）コンテキストが、メモリ上のオブジェクト間を移動し、オブジェクトに付随するプロセッサによって実行される。

このようなモデルを仮にCode Flow Modelと名付けると、ある意味でData Flow Modelの双対として考えることができる。もちろん、上記のようなモデルを直接ハードウェアにマッピングすることは(Data Flow以上に)困難であり、実現のためには何らかの変換が必要となる。例えば、以下のような変形が考えられる。

- (1') メモリ上のオブジェクトは一定の集合を形成し、各々のオブジェクト集合ごとに固有のプロセッサを持ち、全てのプロセッサはコードの完全なコピーを持つ。
- (2') オブジェクト集合の要素をアクセスするプログラム・コードは、プログラム・カウンタを含む（比較的小さな）コンテキストによって表現され、このコンテキストがメモリ上のオブジェクト間を移動し、オブジェクトが所属する集合に付随するプロセッサによって実行される。

このモデルを仮にPC Flow Modelと名付けると、第0近似としては以下のようなものが考えられる。

- 例えばRWC-1 [8] などのように、細粒度メッセージ処理に適合したアーキテクチャを持つプロセッサを、メモリ・オブジェクト集合に付随するプロセッサとする。
- 同じオブジェクト集合に属するメモリ・アクセスをスレッド化し、コンテキストの移動や切替のオーバヘッドを削減する。
- オブジェクトのコピーをいくつかの集合に分散配置することにより、メモリ・アクセスと計算の負荷を分散する。

このようなアーキテクチャがはたして現実的でかつ有効であるかには、もちろん疑問点が多々あるが、広義のvon Neumann bottleneckを通過するものを大きく変質させていることは明らかである。また、このような根源的なパラダイム・シフトを行うか否かに関わらず、並列計算を構成する要素がどのような時間的／空間的な分布をしているべきかを、原点から見直す必要があると考える。

3.1.4.5 研究開発支援体制の課題

Peta-FLOPS Machineのような超大規模の並列計算機に関する研究開発を行う上で、最も問題になるのは研究プラットフォームである。特にソフトウェアに関する研究開発には、ある程度大規模なプラットフォームが必要であるが、現状ではそのような環境を有する研究者はごく限られている。もちろん大規模な並列計算機を有する大学や公的研究機関は少なくないが、それらのほとんどは実際に大規模計算を行うユーザのためのものであり、コンパイラやOSの研究者のプラットフォームとはなり得ない。

このようなプラットフォームの存在が研究開発を飛躍的に進めることの好例は、数年前に富士通が行ったAP-1000の無償使用と無償貸与であり、AP-1000を利用して非常に多数の研究論文が生み出されたことは記憶に新しい。このようなプラットフォーム整備を一民間企業に頼るべきではなく、本来は公的組織が主導的に進めることが強く望まれる。

またハードウェアのみならず、研究開発用のソフトウェア・プラットフォーム、例えばシミュレータ、コンパイラ・ツール、仮想並列マシンなどの整備も必要である。このようなソフトウェア資産をIPAのような枠組を利用して育成することも、重要な課題として検討すべきである。

<参考文献>

- [1] Shinji Tomita, Kei Hiraki, Hiroshi Nakashima, Shin-ichiro Mori, Takashi Matsumoto, Hideharu Amano, Hironori Nakajo, and Toshinori Sueyoshi. Hardware Architecture. In Hidehiko Tanaka, Yoichi Muraoka, Masato Amamiya, Nobuo Saito, and Shinji Tomita, editors, *The Massively Parallel Processing System. JUMP-1*, chapter 5, pp. 169--224. OHMSHA Ltd., February 1996.
- [2] L. Choi, H. B. Lim and P. C. Yew, Techniques for Compiler-Directed Cache Coherence, *IEEE Journal of Parallel & Distributed Technology*, pp. 23--34, Winter 1996.
- [3] 森 眞一郎、福島 直人、五島 正裕、中島 浩、富田 眞治：Self-Cleanup Cacheを採用したNCC-NUMAアーキテクチャの提案、情報処理学会研究報告、Vol. 97, No. 22, 97-ARC-123-8, pp. 43--48, March 1997.

- [4] 五島正裕、森眞一郎、中島浩、富田眞治：Virtual Queue: 超並列計算機向きメッセージ通信機構、情報処理学会論文誌, Vol. 37, No. 7, pp. 1399--1408, July 1996.
- [5] 松本 尚：Memory-Based Processorを使用した汎用超並列計算機の基本アーキテクチャ、並列シンポジウムJSPP'94, pp. 245--252, May 1994.
- [6] Jeffrey Kuskin, et al., : The Stanford FLASH Multiprocessor, *Proc. 21st Intl. Symp. on Computer Architecture*, pp. 302--313, April 1994.
- [7] 牧 晋広、岡本 秀輔、曾和 将容：ユーザプログラム制御階層メモリシステム、情報処理学会論文誌, Vol. 37, No. 10, pp. 1512--1526, October 1996.
- [8] S. Sakai, et al., RICA: Reduced Interprocessor-Communication Architecture, *Proc. 5th IEEE SPDP'93*, pp. 122-125, 1993.

3.2 ソフトウェア

3.2.1 並列システムソフトウェアの課題 (石川 裕委員)

3.2.1.1 プログラミングパラダイム

並列プログラミングパラダイムをデータ授受の観点から大別すると、メッセージパッシング型と共有メモリ型になる。

メッセージパッシング型の欠点は、システムソフトウェアによる通信データのバッファ管理に伴うメモリコピーのオーバーヘッドが問題になる。これにより通信性能が高く通信路における遅延が小さくても、メモリコピーの性能で頭打ちになるケースが多い。

例えば、MPICHと呼ばれるMPIを実装したPDSに基づくIntel ParagonおよびPCクラスタ(表1)上での我々の経験 [1] では、MPIのバンド幅およびレイテンシはParagon上で57 Mbytes/s、69 μ sec、PCクラスタ上で26 Mbytes/s、35 μ secである。表と対比すると分かる通り、ほぼメモリコピーのバンド幅となっている。PCクラスタ上の我々が開発した通信ドライバPMでは、ユーザのメモリ空間に対して116 Mbytes/sのバンド幅を実現しているが、これを利用したMPIではユーザ空間にある通信バッファからアプリケーション側で指定している領域へのコピーのためにメモリコピーのバンド幅以下になっている。

表1

	PC Cluster with myrinet network	Paragon
Processor	Pentium	i860XP
Clock	166 MHz	50 MHz
Network Bandwidth	160 MB/s	200 MB/s
Memory Bandwidth(memcopy)	32 MB/s	68 MB/s

MPIのように同期通信がある場合には、システムは無駄なメモリコピーを削ることが可能となる。この場合、例えば、我々の環境では、ほぼ通信ドライバで実現されている116 Mbytes/sの性能をMPIレベルで出すことが可能となる。これがシステムソフトウェア側としては本質であるが、プログラミングをするユーザにとっては負担を強いることになる。通信遅延を隠蔽するために、あるいは非同期処理を実現するためには、同期通信+マルチスレッ

ドとすることがある。

共有メモリ型プログラミングの利点は逐次処理プログラムからの移行がメッセージパッシング型に比べて楽な点である。しかし、例えば、ローカル計算＋斉通信というパターンでは、通信時におけるネットワークのトータル転送容量の大きさが性能に効いてくる。ここにSMPのような並列計算機におけるバスの限界によるスケーラビリティの悪さが生じてくる。ランダムにメモリをアクセスをするような並列プログラムは共有メモリ型プログラミングが向いている。しかし、共有メモリ型のシステムでは、従来から言われている通りfalse sharingによる性能劣化の問題がある。

3.2.1.2 プログラミング言語システム

並列計算機用に使用されるプログラミング言語システムとしては、i) データ並列が可能な逐次プログラムを自動並列化するコンパイラを提供する、ii) コントロール並列記述を含めた並列処理記述構文を提供する並列プログラミング言語を提供する、の2種類がある。ここでは、ii) に関して議論する。

従来より並列プログラミングのために、より高機能な並列処理記述を提供する試みがなされてきている。研究成果物である新しい並列処理言語は、特定用途には、普及している例を幾つか見ることができるが、FortranやC、C++のように広く一般に普及し利用されている並列プログラミング言語はない。現在でも、新しいプログラミング言語の提案が多数発表されているのが現状である。これは、計算機利用の多様化に伴い、アプリケーションの性質に合致したプログラミング支援を必要としているからであろう。

このような状況では、ある特定用途のプログラミング言語システムを開発するよりも、ユーザのニーズに合わせたプログラミング構文を作成できるプログラミング言語システム、すなわちメタレベルアーキテクチャを持つプログラミング言語が必要である。

メタレベルアーキテクチャの中でコンパイル時メタレベルアーキテクチャでは、ユーザがコンパイル時に、コンパイラのパーザ、コード生成を変更することが可能となる。本機能を利用して、ユーザのニーズに合致する構文をユーザ自身が定義していくことが可能となる。

さらに、行列ライブラリを提供する人は、ライブラリを「組み込みライブラリ」の如く提供することができる。すなわち、ライブラリが呼び出されている環境をコンパイル時に知ることができ、その環境に適したコード生成を記述することができる。非常に簡単な例として並列基本行列計算ライブラリを考えてみる。

$Y = A * X + B$ は次のようなライブラリコールになるであろう。

MatMul(A, X, TMP)

MatAdd(TMP, B, Y)

従来の処理系では上記のライブラリコールは単なるライブラリコールの列としてしか生成できない。しかし、メタレベルアーキテクチャでは、上記のプログラムを展開し、一時変数をなくして最適化することが可能となる [2]。

3.2.1.3 デザインパターン

ソフトウェア工学の分野では、再利用技術として、クラス、クラスライブラリ、フレームワーク、デザインパターン等と研究のシフトが進んでいる。デザインパターンとは、プログラミングにおける定石あるいは手筋をまとめたものであると言える [3]。ソフトウェア工学においては、ウインドウプログラミングやビジネスアプリケーション等においてデザインパターンの応用事例が多い。並列プログラミングにおいても、デザインパターンが重要であり今後の研究の発展が望まれるところである。

3.2.1.4 オペレーティングシステム

並列計算機を気楽に使える環境を提供するシステムソフトウェアが必要である。利用者はワークステーションを利用しているかのごとくシームレスに並列計算機が利用できる必要がある。ワークステーションで主流のUnix系を採用している現状の並列計算機上のOSは、このような環境を満たしているように見えるが問題点が多い。各プロセッサ上にローカルなUnix OS が独立して稼働しているために、グローバルな資源管理に基づく効率よい並列実行環境が提供されていない。そこで、いくつかの商用計算機では、UnixによるTSS環境とバッチ環境の2つを提供し、システム管理者は、ユーザのニーズに合わせ、TSSおよびバッチ環境のサービスをスケジュールすることをしている。

一方、Unix擬きのOSを開発し、ギャングスケジューリング等を採用することにより、並列マルチタスク環境を実現しているシステムも存在する。しかし、このようなシステムは、ワークステーション上ほどの豊富なプログラミング支援ツールが整備されず廃れていくことが多い。

コモディティOS環境を並列計算機上で効率良い実行環境を提供するシステムが必要とされている。現在、我々は、PCクラスタ上で、そのような環境を実現すべく研究開発を進めている [3]。並列OS、通信ドライバ、プログラミング言語を同時に設計し、ギャングスケジューリングを実現している。並列OSであるSCoreはUnix系の上に構築している。プログラミング言語はC++を拡張したMPC++と呼ぶ言語を開発している。

<参考文献>

- [1] F. O'Carroll, "Performance of MPI on Workstation/PC Clusters using Myrinet, "Cluster Computing Conference'97, Atlanta, 1997.
- [2] Y. Ishikawa, et.al, " Design and Implementation of Metalevel Architecture in C++ -- MPC++ Approach --," Reflection '96, pp. 141 -- 154, 1996.
- [3] RWC Massively Parallel Software Laboratoy Home Page, <http://www.rwcp.or.jp/lab/mpslab>

3.2.2 並列化コンパイラから見たペタフロップスコンピューティング

(笠原博徳委員)

5～10年後のペタフロップスマシン研究開発方針として重要なのは、ピーク性能だけではなく、アプリケーションを実行した時の実際の性能（実効性能）の向上を図るということである。これは、従来分散メモリ型マルチプロセッサを用いた超並列システムが高いピーク性能が得られるため注目されていたが、実効性能がピーク性能の数%から30%程度の低い値にしか達せず、より低いピーク性能をもつ共有メモリ型マルチベクトルプロセッサシステムの方が多くのアプリケーションに対して高い実効性能を与える [1] ことが知られるようになってから、米国を中心に科学技術計算に超並列型のマシンは適さないと理解され、超並列マシンを開発していた企業が事業から撤退していったことを見ると明らかである。

このように実効性能の高いペタフロップスマシン開発を行うためのキーとなるのは、コンパイラを始めとしたシステムソフトウェアの高度化 [7、8] と、ソフトウェアにおける並列化技法の性能を最大限に引き出すことのできるアーキテクチャ [14] の開発すなわちソフトウェア/ハードウェア協調設計型マルチプロセッサシステムの開発である。もちろん、システムソフトウェアにおける並列処理方式の開発においては、各種のアプリケーションプログラムの並列性を十分に活かせるようにしなければならない。

また、高実効性能ペタフロップスマシンにおいて、多くのプロセッサを利用率高く動作させ、プロセッサ間データ転送オーバーヘッドおよび同期オーバーヘッドを低くおさえるために、ユーザが高度なチューニングを余儀なくされるというのでは、商業的成功は望めない。すなわち、ペタフロップスマシンといえども、誰にでも簡単に使いこなせるようなユーザフレンドリネスが重要と考えられる。このためには、ユーザが並列処理を意識することなく従来の使い慣れたCあるいはFortranなどの言語でプログラミングを行うと自動的に並列化してくれるコンパイラ（このようなコンパイラでは原子力コードのようなFortranで記述された既存のプログラムの自動的な実行も可能） [2-5]、あるいはユーザはプログラミングを行わず、ユーザが使い慣れた問題記述法、例えば電力システムの解析であれば電力系統図、制御系であれば伝達関数、ロボットであれば関節およびリンクの構成などをグラフィカルインタフェースを用いて入力すると、問題の求解プログラム（例えば電力系統過渡安定度計算のための非線形微分方程式求解プログラム）を内部で自動的に作成し並列化する専用目的コンパイラの開発が重要と思われる [2、12]。

以下では、ペタフロップスマシンの開発に向けて今後ますますその重要度が高まると考えられる自動並列化コンパイラ技術の現状と課題、またそれらの対処策の例について簡単に述べる。

(1) 自動並列化コンパイラの現状

並列化コンパイラは逐次型言語あるいはその拡張言語で書かれたプログラムに、

- ・ プログラム中のデータの流れ及び制御の流れの解析による並列実行可能な部分の検出
- ・ 使用する並列処理システムの性能を最大限に引き出すプログラムのタスク（並列処理単位）への分割（タスク粒度の決定）
- ・ プロセッサ間データ転送オーバーヘッドを最小化するデータの分割
- ・ 並列処理できる形へのプログラムの自動変換（restructuring）
- ・ 並列実行時間を最小とするようなタスクのプロセッサへのスケジューリングおよびデータのローカルメモリあるいはグローバルメモリへの配置
- ・ 並列化マシンコードの生成

等の処理を適用し、並列化マシンコード（あるいは並列化拡張言語）を生成するものである。

現在までに実用化されているマルチプロセッサシステム用自動並列化コンパイラの並列性抽出部では、プログラム中で最も多くの時間が費やされるDoループの並列化を対象にしておき、ループ内のデータ依存解析 [2、3、5] およびDoall, Doacrossループへの変換（ループ並列化） [4] が行われる。

ここで、データ依存解析とは、並列性抽出のために、命令間、ステートメント間、サブルーチン間、ループ間のデータの定義・使用関係により生じる一種の実行順序関係の制約を解析することであり、具体的にはフロー依存、出力依存、逆依存、入力依存、の4種類があるが、並列性解析ではフロー依存、出力依存、逆依存が解析され、入力依存はデータの分割配置を行う際に考慮される。

次に、制御依存解析とは、ステートメント等が並んでいる順序および条件分岐等を解析しながら、ある条件分岐の分岐方向が決定するとどのステートメントが実行可能になるかを解析することで、データ依存を考慮しない場合のステートメント間の最大の並列性を調べることができる [2]。しかし、実際にはデータ依存を考慮しながら並列性を解析することが必要であり、制御依存およびデータ依存を考慮して最大の並列性を解析する方法として最早実行開始条件解析 [2、11、15] が提案されている。

またプログラム・リストラクチャリングでは、データ依存解析等の結果を利用して、Doall, Doacrossループに変換（ループ並列化）するほか、そのままではベクトル化あるいはループ並列化できないループを、

(a)ステートメントの実行順序の変更

(b)ループ分散（ループディストリビューション）

(c)テンポラリ（配列）変数への代入を行うステートメントの挿入（ノードスプリッティ

- ング、スカラエクスパンション)
- (d)多重ループにおける内側ループと外側ループの入れ換え (ループインターチェンジ、ループパーミュテーション)
- (e)ループの展開 (ループアンローリング)
- (f)1重ループの多重ループへの変換 (ストリップマイニング)
- (g)各プロセッサ用プライベート (配列) 変数の生成 (アレイプライベートイゼーション) [22]
- (h)ユニモジュラー変換 (ループリバーサル、パーミュテーション、スキューイング) [4]

等各種の手法を用いて並列化を可能にする。

このループ並列化のためのプログラムリストラクチャリング手法に関しては、非常に多くの研究がイリノイ大学が中心に行われ [4、5、20]、現在は成熟期に入っている。

しかしこのようなループ並列化によるイタレーションレベル中粒度並列処理技術だけでは、アムダールの法則からも分かるように、プロセッサが数千以上のペタフロップスマシンの場合には並列化できないループやループ以外の処理部分の影響で、実効性能の向上を図ることは困難になる [18]。

(2) ペタフロップスマシン用自動並列処理化コンパイラの研究課題

上述のようなループ並列化技術の限界を考慮し、ペタフロップスマシン用自動並列処理化コンパイラでは、米国におけるペタフロップスワークショップでも一部議論されているように、以下のような並列化技術の開発が重要になると考えられる。

- (a)従来のように一部のループだけではなく、プログラム全域にわたり全ての並列性を抽出・利用するために、サブルーティン／ループ間の並列性を利用する粗粒度並列処理 (マクロタスクデータフロー処理) [9] [15]、従来のループ並列化 (中粒度並列処理)、逐次ループあるいはループ外の基本ブロックの内の処理を並列化するための (近) 細粒度並列処理 [13、17、18] を階層的に適用するマルチグレイン並列処理技術 [9、11、15]
- (b)各プロセッサ上のローカルメモリあるいは分散共有メモリを有効に利用し、プロセッサ間のデータ転送を最小化するために、複数ループ間のグローバルなデータ依存解析、ループの移動を伴うデータ自動分割・配置技術 [16、19、23、24]

(c)データ分割配置によるデータ転送の最小化によっても除去できなかったプロセッサ間データ転送のオーバーヘッドを隠蔽するための、データブロードあるいはポストストアを用いた処理とデータ転送のオーバーラッピング[10]を可能とするスケジューリング技術 [2] [6]

(a) のマルチグレイン並列処理は、ペタフロップスマシンにおけるプロセッサ間だけではなく、5年後から7年後にマイクロプロセッサの主流になるとと思われるシングルチップマルチプロセッサでも非常に重要となる技術である。

また (b) のデータの自動分割配置と (c) の処理のデータ転送のオーバーラッピング技術は、今後アーキテクチャ的にもコンパイラ的にも非常に重要な研究テーマになると考えられているソフトウェア (プログラム) 制御キャッシュの開発のための基礎技術としても重要である。

(3) まとめ

本節では、商業的に成功するペタフロップスマシンの開発においては、実効性能および使い易さの向上が重要であり、このためには自動並列化コンパイラの高度化、コンパイラによる並列化技術の性能を最大限引き出すことのできるアーキテクチャの研究が重要であることを述べた。さらに、自動並列化コンパイラの現状と問題点を指摘し、今後マルチグレイン並列化、データ分割配置、処理とデータ転送のオーバーラップスケジューリングなどのコンパイラ技術を開発実用化して行くことが重要であることを説明した。

また、これらのコンパイル技術は、ペタフロップスマシンのみでなく、現在日本が米国に大きく遅れているRISC系プロセッサの開発において、5年後以降に世界をリードするプロセッサを開発するチャンスとなると考えられるシングルチップマルチプロセッサのアーキテクチャに大きな指針を与えると共に、そのようなプロセッサを効率良く動作させるためにも極めて重要な技術となると考えられる。

しかし現在のところわが国では、このような最先端の並列化コンパイラの研究を行う研究者の数が大学、企業、国研を通じて少なく、産官学協力し早急に人材を育成することが必要である。さらに、このコンパイラ技術は、将来のマイクロプロセッサからスーパーコンピュータまでの性能を左右する最重要課題の一つと考えられるので、国としても研究の推進を積極的に支援していく必要があると思われる。

<参考文献>

- [1] G.Cybenco, D.J.Kuck, "Teraflops Galore: Supercomputing /Reinventing the Machine - Revolution or Evolution?" IEEE Spectrum, pp. 39-41, Sep. 1992.

- [2] 笠原, 並列処理技術, コロナ社, (1991)
- [3] U.Banerjee, Dependence Analysis for Supercomputing, Kluwer Academic, 1988
- [4] U.Banerjee, Loop Parallelization, Kluwer Academic Pub., 1994
- [5] M.Wolfe, High Performance Compilers for Parallel Computing, Addison Wesley, 1996.
- [6] H.Kasahara and S.Narita, "Practical multiprocessor scheduling algorithms for efficient parallel processing", IEEE Trans. Comput., Vol.C-33, No.11, Nov.1984
- [7] 笠原, "並列処理ソフトウェア", 電気学会論文誌C「並列処理技術特集」, Vol.113, No.11, pp.919-927, Nov. 1993.
- [8] 解説特集"並列処理のためのシステムソフトウェア", 情処, Vol.34, No. 9, Sep. 1993.
- [9] 笠原, 吉田, 合田, 岡本, 本多, "Fortranマクロデータフロー処理のマクロタスク生成手法", 信学誌, Vol.J75-D1, No.8 pp.511-525, 1992.
- [10] 藤原, 鈴木, 白鳥, 笠原, "データプレロードポストストアを考慮したマルチプロセッサスケジューリングアルゴリズム", 信学論, Vol.J75-D1, No.8, pp.495-503, 1992
- [11] H.Kasahara, H.Honda, S.Narita, "A Multi-Grain Parallelizing compilation scheme for OSCAR," Proc.4th Workshop on Languages and Compilers for Parallel Computing, 1991
- [12] 笠原, 藤井, 本多, 成田, "スタティック・マルチプロセッサ・スケジューリング・アルゴリズムを用いた常微分方程式求解の並列処理, 情処論, Vol.28, No.10, pp.1060-1069, Oct. 1987.
- [13] H.Kasahara, H.Honda and S.Narita, "Parallel Processing of Near Fine Grain Tasks Using Static Scheduling on OSCAR", Proc. IEEE ACM Supercomputing'90, Nov. 1990.
- [14] 笠原, 成田, 橋本, "OSCARのアーキテクチャ", 信学論D, Vol.J-71D, No.8, Aug.1988.
- [15] 笠原, 岡本, 合田, 宮沢, 本多, 笠原, "OSCARマルチグレインコンパイラにおける階層型マクロデータフロー処理", 情処論, Vol. 35, No.4, pp.513-521, Apr. 1994.
- [16] 吉田, 前田, 尾形, 笠原, "Fortranマクロデータフロー処理におけるデータローカライゼーション手法", 情処論, Vol. 35, No.11, 1994
- [17] 尾形, 吉田, 合田, 岡本, 笠原, "スタティックスケジューリングを用いたマルチプロセッサ上の無同期近細粒度並列処理", 情処論, Vol.35, No.4, Apr. 1994.
- [18] 笠原, "マルチプロセッサシステム上での近細粒度並列処理", 情処, Vol.37, No.7, 1996.
- [19] 吉田, 前田, 尾形, 笠原, "Fortran粗粒度並列処理におけるDoall/シーケンシャルループ間データローカライゼーション手法, 信学論, Vol. J78-D-I, No.2, pp.162-169, Feb. 1995.
- [20] W.Blume, R.Eigenmann, J.Hoeflinger, P.Petersen, L.Rauchwerger and Peng Tu, "Automatic Detection of Parallelism, IEEE Parallel & Distributed Technology, Vol.2, No.3, pp.37-47, Fall

- 1994.
- [21] D.J.Lilja, "Exploiting the Parallelism Available in loops,"IEEE Computer, pp.13-26, Vol.27, No.2, Feb.1994.
 - [22] P.Tu and D.Padua, "Automatic Array Privatization," 6th Annual Workshop on Languages and Compilers for Parallel Computing, 1993
 - [23] M.Gupta and P.Banerjee, "Demonstration of Automatic Data Partitioning Techniques for Parallelizing Compilers on Multicomputers," IEEE Trans.on Parallel and Distributed System, Vol.3, No.2, pp.179-193, 1992.
 - [24] J.M.Anderson and M.S.Lam, "Global Optimizations for Parallelism and Locality on Scalable Parallel Machines," Proc. of the SIGPLAN '93 Conference on Programming Language Design and Implementation, pp.112-125, 1993.

3.2.3 ペタフロップスマシンにおけるプログラミングインタフェース

(妹尾義樹委員)

科学技術計算分野におけるペタフロップスマシンの実現を考えると、ユーザに使いやすく、しかも高いシステムの性能を引き出せるためのプログラミングインタフェースを提供することは不可欠である。このための要素技術、技術課題、将来の予想、技術調査のポイントについて述べる。

3.2.3.1 日米における技術的現状の比較と我が国での問題点

現在の科学技術計算の分野では、最高速のシステムのピーク性能が、数百ギガFLOPSから1テラFLOPSである。これらのシステムはすべて、分散メモリタイプの並列マシンであり、プログラミングインタフェースとしては、大きく分けて、メッセージパッシングによりSPMD (Single Program Multiple Data Stream) 形式のプログラムをユーザが明示的に記述する枠組みと、データパラレルプログラミングの2種類がある。前者はMPI (Message Passing Interface) [1]、後者はHPF (High Performance Fortran) [2] [3] が現在の代表的な標準インタフェースとして認知されている。いずれの場合でも、システムの高い性能を引き出すための鍵は、データアクセスローカリティの抽出である。つまり、これらのシステムでは、演算性能に比べて、通信性能が計算のボトルネックになる場合がほとんどであり、データアクセスの局所性をうまく引き出して、必要となる通信量を最小にする必要がある。

MPIはメッセージパッシング、つまり、プロセッサ間のデータ転送を、ユーザがライブラリを用いて明示的に記述する枠組である。SPMD形式に基づいて、複数の制御の流れを意識したプログラミングが必要になる。これに対してHPFは従来のFortran言語に最小限の付加的指示で並列化を可能とすることを狙ったデータパラレル言語である。現在はまだ仕様そのものや処理系(コンパイラ)の完成度の面で発展途上であるが、大きな可能性を秘めるとともに、今後、分散メモリマシンを並列処理の専門家だけの研究ツールから実用ツールにするためには必須のプログラミングインタフェースである。このMPIとHPFについて、日米の技術的現状について比較を行う。

(1) MPI

MPIは、分散メモリ型のプログラミングインタフェースとして、現在最も標準的に用いられているものである。MPIは、米国のJ. Dongarra、D. Walkerや欧州のR. Hempelらによって標準化が推進された。標準化はMPIF(MPI Forum)と呼ばれるプライベートなフォーラムで行われた。このインタフェースは、米国のPVM、P4や欧州のPARAMACSなどのメッセージパッ

シングインタフェースをベースに設計されており、現在では、並列I/Oやプロセスの動的生成・消滅などの機能をさらに追加したMPI2の言語仕様が検討されている。また、最近では、並列計算機用の標準ベンチマークとしてMPIで書かれたプログラムが開発されている。日本と欧米では、言語インタフェースに関する研究のみならず、その利用経験の蓄積において、まだかなりの開きがあると思われる。

(2) HPF

HPF言語の標準化活動もMPIと同様に、Rice大学CRPCのKen Kennedy 教授を中心とするプライベートな組織HPFF(HPF Forum)で行われた。ANSIを中心に行われたFortran90の仕様制定に10年以上かかった反省からである。活動は1991年に開始され、1993年にHPF1.0が制定された[2]。その後もHPFF2において、仕様の曖昧性のチェックや、HPF1.0で不十分であった機能拡張が検討され、昨年末にHPF2.0 [3] の仕様が定められた。HPFの標準化活動も、主に米国を中心として行われ、商用のコンパイラについてもPGIやAPRといった米国ベンチャーが日本のベンダに比べて先行している。HPF関連の技術についても、特にその利用技術の蓄積について欧米が日本に比べて格段に進んでいると思われる。特に米国では、TMC社(Thinking Machine Corporation)のConnection Machine上でCM-Fortranを用いたプログラミング経験が長く蓄積されており、現在のHPFで書かれた実用コードの多くはCM-Fortranプログラムを書き換えたものである。後で、さらに詳しく触れるが、日本の並列処理技術の発展のためには、コンパイラとユーザの緊密な共同作業による、利用経験の蓄積、利用環境の整備が急務である。

3.2.3.2 ペタフロップスに向けた将来予想と解決策

現在のデバイステクノロジーの延長線上で、15年から20年先のスパンでペタフロップスマシンを実現するとすると、これは並列システム、それも少なくとも1000台以上の、かなり多数の要素プロセッサが結合された超並列システムとならざるを得ないと考えられる。また、これら多数のプロセッサを効率よく動作させるためには、物理的には、現在のテラフロップスマシンと同様に、分散メモリ構成にならざるを得ない。共有メモリでは、メモリアクセスがボトルネックとなり、システムの実効性能を制限するからである。

物理的に分散メモリで構成された並列プロセッサで性能の高いプログラムを記述する際のポイントは、データの分散メモリへのレイアウトである。結合された各プロセッサを可能な限り独立に並列動作させるためには、データアクセスのローカリティを最大限に抽出する必要があるからである。

ペタフロップスシステムの実現を可能にするための言語処理技術について、そのニーズ、

要素技術、パラダイムシフトについて述べる。

(1) ニーズ

科学技術計算の分野における、計算機システムに対するニーズについて、言語インタフェースに関連するものは以下のものである。

- (a) 高速性：さらに分類すると、計算のターンアラウンド時間とスループットに分けられる。
- (b) 適用性：高い性能が得られる応用プログラムのレンジ。適用性を決める大きな要因としては、非定型問題の対処や使えるメモリ容量などがある。
- (c) ポータビリティ：ある高速システムを対象として記述したプログラムが、そのままアーキテクチャの異なる他のシステムで高速に実行できるかどうか。対象システムとしてはプログラムの開発環境という意味で、WSやパソコンも考慮する必要がある。
- (d) 記述性、保守性：プログラム開発や、その保守の容易性が重要である。これは言語インタフェースやコンパイラの機能だけではなく、デバッガやチューニングツールなどの整備が大きな要因となる。数値計算ライブラリの整備なども記述性に大きく影響する。

これらの各項目は、一般にはトレードオフの関係にある。すなわち、高速性だけを追求して、他を無視してよいならば、ユーザがアセンブラでプログラムを開発すればよく、複雑な言語処理系を開発する必要はない。逆に高速性を犠牲にして、他の項目を実現するのも容易である。実際には、要求される高速性を維持しながら、他の項目を高めていく必要があり、これには、一つ一つの言語処理系に関する要素技術を着実に完成させていく地道な努力が必要である。

(2) 要素技術

上記で述べたニーズを実現するために必要となる要素技術について、特にHPFなどのデータパラレル言語の枠組みに話を絞って述べる。

HPF言語の現在の問題点は、並列化/通信最適化についての記述力の不足と、これを補うだけの処理系による自動最適化が現状の技術では困難なため、並列化の適用範囲が制限されることである。今後、HPFがベタフロップスマシンの言語インタフェースとなりうるためには、以下の要素技術を確立する必要があるだろう。

(a) プログラム記述能力の向上

- ・不規則問題の記述： 有限要素法など、配列の間接参照や不規則データ構造の問題をコンパイラが効率よく処理できる形で記述する枠組みの研究が必要。

- ・タスク並列性： HPFで記述できる並列性の指定は、INDEPENDENT指示行で記述できるDOALL型のループ処理とFORALL型のループ処理および配列演算だけである。これ以外のDO ACROSS型の並列化や、制御並列を含むようなタスク並列性の指定ができない。これは、HPF言語がシングルストリームのセマンティックスを基本としているためである。何らかの形でHPFの中にタスク並列性／オブジェクト並列性を取り込む必要がある。
- ・並列入出力： 大規模演算においては、大規模データ(ファイル)を扱うことが多く、入出力が並列化の妨げになることが多い。これを解決するための並列I/Oを記述する方法の研究が重要である。

(b) コンパイラ解析能力／最適化能力の向上

- ・アルゴリズムレベルの最適化： 現在のコンパイラは、記述されたプログラムのセマンティックスを保存する形の最適化しか行うことができない。最適化はシンプルなパターンマッチングにかざられている。与えられたプログラムのアルゴリズムを認識し、等価なアルゴリズム変換を行うなどの最適化手法の研究が重要である。Vienna大学など、このような研究に着手したところもある [4] 。
- ・不規則処理の最適化： 非線型添え字を伴う処理の並列化には、通常は、INSPECTOR／EXECUTOR [5] といって、添え字の内容を先に解析し、この解析結果を用いて効率の良い通信を生成する方法が用いられる。しかしながら、現状では、まだまだ性能の高いプログラムが生成できるのは、非常に限られた場合だけである。ペタフロップスシステムを汎用とするためには、不規則処理を避けて通るわけにはいかない。
- ・通信最適化： 複雑なケース、特に複数の手続きにまたがった通信最適化は、まだまだ非常に困難である。また、プログラムは同一でも、対象とするシステムや入力データによっては採用すべき通信最適化の手法が異なることが多い。
- ・手続き間解析： 分散メモリ上の並列化は、グローバルな情報が必要であるが、これには手続き間解析が不可欠である。大学などで研究用に開発された処理系や一部の商用システムでは、すでに手続き間解析が採用されているが、まだまだ課題は多い。例えば、これらのシステムでは、大規模ソフトウェアを開発する際に、1つのファイルを更新すると、全ファイルを再コンパイルする必要がある。また、詳細で正確な手続き間解析には、長時間の解析時間と大量のメモリが必要になる。
- ・自動データレイアウト [6]： 現在のHPFでは、データのレイアウトは、すべてユーザが明示的に記述するのが基本である。しかし、この作業を最適に行うには、対象となるシステムの詳細に理解する必要がある。並列処理の専門家以外が使える汎用の高速システムの実現には、データレイアウトの自動化または、すくなくともレイアウトを支援するツールの研究が重要である。

(c) 実行時最適化技術

並列化においては、最適化の方法や、最適化のパラメータがデータサイズやループ長に依存することが多い。選択肢が2、3個の場合には、マルチバージョン方式といって、複数の最適化コードを生成しておき、実行時にこれらを選択実行する手法がとられるが、選択肢が多数あったり、複数のパラメータが相互にからみあう場合には、使えない。実行時にプログラムの最適化方法を決定できる枠組みの研究が必要である。

(d) 階層構造の克服

ベタフロップスシステムが、仮に1万台以上の要素プロセッサから構成されるとすると、これらプロセッサが一様に1階層で構成されるとは考えにくい。クラスタなど、何らかの階層構造が採用されるのが普通である。この場合、並列化およびデータレイアウトについて、ユーザに階層を意識させない枠組みの研究が必要である。

(e) ツールの整備

ベタフロップスマシンになると、今より一層デバッグや性能チューニングが困難になると考えられる。これらの作業をシステムティックに支援する統合ツールの研究開発が重要である。

(3) パラダイムシフト

以上まで、ベタフロップスマシンが、現在の延長上で実現され、利用されることを前提に、今後の課題、検討すべき要素技術について述べた。以下ではシステムの利用形態も含めて、起こりうるパラダイムシフトの可能性について大胆に論じてみる。

(a) 異機種分散処理

LAN、WANなどのネットワーク技術の進展により、ベタフロップスマシンは、異機種分散システムとして実現される可能性がある。現在の大規模数値シミュレーションにおいても、主要計算部は計算サーバで実行し、結果の可視化やデータベース処理に専用システムを用いる場合がある。また、計算部分についても、マルチdisciplinaryコードなど、複数の計算スキームを扱うようなシミュレーションにおいて、適材適所のプロセッサを用いた異機種分散並列処理による高速処理が考えられる。このような場合には、異機種間での負荷分散、スケジューリングや、通信最適化技術の研究が重要であるとともに、異機種システム上での分散処理をユーザに意識させずに効率良く処理を行うフレームワークの確立が重要である。また、オブジェクト並列処理とデータ並列処理を融合するアプローチの研究も重要となるであろう。

(b) Reduced Programming Interface

ベタフロップスマシンにおいては、性能が出るパターンが非常に限られる可能性がある。このようなシステムの上でFortranやCのような、汎用の何でも記述できる言語を提供するのは实际的でないかもしれない。このような場合には、言語仕様として、性能の出るパターンだけをユーザに書かせられる枠組みの研究が大切である。言語の記述性は制限されるが、ユーザは、この言語に従う限り性能が保証されるのであれば、安心してプログラミングができる。この方向としては、Fortranのサブセットとして定義された、プログラミング言語F [7] という研究がある。

(c) ライブラリ指向プログラミング

上記と同じように、ベタフロップスマシンでは、高性能が得られる計算パターンが、かなり限られる可能性がある。この場合、性能の出るパターンをライブラリとして提供しておき、ユーザはこれの組み合わせにより、すべてのプログラミングを行うかもしれない。並列化や性能上プログラミングの困難な部分は、すべてライブラリの中に隠蔽できれば、ユーザはこれらに頭を煩わされることなくプログラミングが可能になる。ライブラリだけですべての問題を解決できる可能性は低いが、少なくとも現在のScaLapack [<http://www.netlib.org/scalapack/>]のような並列ライブラリインタフェースとその実現に関する研究は今後ますます重要になると考えられる。

3.2.3.3 わが国として取り組むべき課題（特に、国策として実行すべき課題）

これまで、並列処理における言語インタフェースの研究は、主にコンパイラ研究開発者を中心に行われてきた。しかしながら、今後の高性能計算機における言語処理技術を発展させるためには、コンパイラ開発者と計算機ユーザとが緊密に連携した研究活動が重要である。米国でMPIの標準化活動を推進したのは、主にユーザが主体であったし、最近では欧州でも

PHAROS[<http://www.vcpc.univie.ac.at/activities/projects/PHAROS/>]や

HPF+[<http://www.par.univie.ac.at/hpf/>]

といった、ユーザと言語研究者が一体となったECのESPRITプロジェクトが発足している。今後は前節 [2] で述べたような研究課題を両者が一体となって進めるプロジェクトを国策として実行すべきであると考ええる。

3.2.3.4 さらにどのような技術調査をすべきか

今後重要であると思われる技術調査項目について列挙する。

- ◆タスク並列性/オブジェクト並列性とデータ並列性の融合
- ◆データ自動レイアウト技術
- ◆静的性能予測技術
- ◆アルゴリズム認識+チューニング技術
- ◆プログラミングツール
- ◆実用コードの並列化

個人的には、この中で、実用コードの並列化例を可能な限り洗い出し、この中からベタフロップスマシンで性能が出るとされる計算パターンを分類する作業が重要であると考ええる。

<参考文献>

- [1] Message Passing Interface Forum, MPI: A Message-Passing Interface Standard, The International Journal of Supercomputing Applications and High Performance Computing, Vol.8, No. 3/4, pp. 165-416 (1994).
- [2] High Performance Fortran Forum: High Performance Fortran language specification, version 1.0, Technical Report CRPC-TR92225, Center for Research on Parallel Computation, Rice University (1992).
- [3] High Performance Fortran Forum: High Performance Fortran language specification, version 2.0 (1996).
(<http://www.crpc.rice.edu/HPFF/hpf2/index.html>)
- [4] B. D. Martino, Design of a Tool for Automatic Recognition of Parallelizable Algorithmic Patterns and Description of its Prototype Implementation, TR 95-3, Institute for Software Technology and Parallel Systems, University of Vienna (1995).
- [5] R. Das, J. Saltz, R. von Hanxleden, Slicing Analysis and Indirect Access to Distributed Arrays, Proceedings of the 6th Workshop on Languages and Compilers for Parallel Computing, Portland, OR, August (1993).
- [6] U. Kremer, Automatic Data Layout for High Performance Fortran, Proc. Workshop on Automatic Data Layout and Performance Prediction, Rice University, April (1995).
- [7] W. Brainerd, C. Goldberg, and J. Adams, Programmer's Guide to F, Unicomp, ISBN 0-9640135-1-7 (1996).

3.3 アプリケーションとアルゴリズム

3.3.1 ペタフロップスコンピューティングにおけるアプリケーション技術

(関口智嗣委員)

3.3.1.1 ペタフロップスにおける課題

21世紀の高度情報化時代のキーテクノロジーである基盤ソフトウェアにおいて我が国は大きく立ち遅れている。とりわけ、産業界における製品設計開発、製造、研究開発などの目的でさまざまに用いられている共通の知的資産としてのソフトウェア技術（以下、産業基盤ソフトウェア技術と呼ぶ）の大部分は海外の特定の製品に大きく依存している。具体的には、情報処理産業のみならず、新機能材料、半導体設計、新エネルギー開発、地球環境対策、資源探索、機械設計、防災計画などの分野で用いられている産業基盤ソフトウェアはその源流が欧米に発するものが多い。このような現状は、欧米各国から「科学技術タダ乗り」との非難を浴びると同時に、将来、わが国の産業競争力の著しい低下を招くものであり、早急に対策を講じる必要がある。このような産業基盤ソフトウェアや米国のHPCC計画グランドチャレンジに代表されるような大規模科学技術計算問題は今後ますます超高速スーパーコンピュータ、大規模データベース、広帯域高速ネットワークを最大限に活用して解く必要となってくる。したがって、こうしたソフトウェア技術がペタフロップスコンピューティングの発展を強く推進するニーズとなる。計算能力の観点からだけでも大雑把に1,000ギガフロップスから100,000ギガフロップス以上の能力がアプリケーションをより高精度に、より大規模に、より高速性が要求される。ペタフロップス（1,000,000ギガフロップス）の演算能力はインターネットに接続されているPCの総演算能力の数倍程度であり、また最高速の超並列計算機の1000倍程度に相当する。このように現状の技術の単なる延長上ではペタフロップスコンピューティングは実現されない。ここでは、アプリケーションを実行する観点からペタフロップスを実現する構成を 1) 単体性能 2) 並列度 3) ネットワークの3点から仮定を導入する。さらに、アプリケーションの実例として日本で計画中のVRMS (Virtual Microscope) を例にとり、ペタフロップスコンピューティングでの実現方法と展望について述べる。

3.3.1.2 ペタ・フロップスシステム構成の仮定

単体プロセッサはすでにその能力の限界が来ていると言われる。しかし、半導体製造技術の革新により21世紀には1GIPSのマイクロプロセッサが実現すると言われている（本委員会ヒアリング資料を参照）。これをペタフロップスシステムの基本コンポーネントとする。基

本コンポーネントに付加する専用ハードウェア加速装置はアプリケーションをある程度特定することで、実現することが可能である。従来の高速計算システム発展の歴史を見ると、汎用機に付加した IAP（アレイプロセッサ）、これを一般化したベクトルパイプライン、FPS社などから提供されたミニコンピュータ用の加速装置、CM-5等で採用されたマイクロベクトルプロセッサ、最近ではGRAPEなどに代表される超高速な専用加速演算装置がある。これらは従来の汎用プロセッサと比較して10倍以上の能力があり、ペタフロップスを達成するための基本コンポーネントに不可欠なものである。ここでは、10ギガフロップスの専用ハードウェアを付加した1 GIPSの汎用プロセッサという2ユニットを基本として単体の演算装置を構成する。

並列度についてはコネクタ、ケーブル等に起因する物理的な制約、相互結合ネットワークトポロジの制約、容易なプログラミング環境の提供等の条件から現状に対してさほど今後の技術発展が期待できない。もちろん、相互結合ネットワークに要求されるネットワークスループットは現状の10倍程度以上は必要となる。ここでは、Hitachi SR2201等ですでに実績のある1,024台程度で3次元クロスバという構成の相互接続が可能であると仮定する。

上記2つの条件は単体プロセッサならびに相互結合ネットワークから構成される超並列システムに関するものである。すなわち単体のシステムの限界が10ギガフロップスを1,024台で並列システムとして構成することにより約10テラフロップスを実現するというものである。しかし、これではまだペタフロップスには到達しない。そこで、この不足分を補うためには10テラフロップス級規模のシステムを100台規模高速ネットワークで接続したネットワークコンピューティングで解決する。現時点では32台から64台規模の構内ネットワークコンピューティングが現実的であるが、100台規模の広域分散コンピューティング環境を実現するためには

- (1) 高速ネットワーク技術の開発
- (2) 構内から広域へのネットワーク利用基盤技術の開発
- (3) 分散アプリケーション技術の開発
- (4) 超大規模分散データベース技術

といった点の技術開発が必要であるが不可能な数字ではない。

以上をまとめて、スカラプロセッサ1 GIPS+専用ハードウェア加速装置10ギガフロップス、これを1,024台並列のシステム（10テラフロップス）を100台程度のネットワークコンピューティングにより1ペタフロップスを実現することをこの章での仮定とする。

3.3.1.3 ネットワークコンピューティング技術

上記仮定においてネットワークコンピューティング技術開発がますます重要となるであろう。国内における高速ネットワーク技術の応用は従来マルチメディア利用技術として開発が実施されてきたため、HPCC技術に発展するようなネットワーク技術は、特に日本の技術が後れている。ネットワークコンピューティングとは高速ネットワークにより接続された計算機資源や計算資源を駆使することにより、それぞれの資源の物理的な位置を意識することなく従来単独では実行不可能であった情報処理を実現する技術である。

ネットワークコンピューティングの対象としては超高速数値処理を目指した技術と超大規模データ処理を目指した技術がある。すなわち、前者はスーパーコンピュータ、並列計算機システムといった高速計算機資源を有機的に連携して利用し、大規模なシミュレーションを実現する。後者はそれぞれが100GB程度のデータベースがシステム全体で1TB以上級のデータに対して処理を実施するものである。超高速数値処理アプリケーション技術としては例えば地球環境のシミュレーションでは海洋循環モデル、大気循環モデル、極地モデル、大陸モデル、化学反応モデル等を個別に高速計算機に分配し、これらを統合するためのカプリングモデルを導入しシミュレーションを実施する。現状の技術ではスーパーコンピュータを用いてもモデル単独を処理する程度が実用的である。精度の高い計算、長い時間軸に添った計算を行うにはネットワークで接続された高性能計算機の利用技術が必須である。また次節で述べるバーチャルマイクロスコプにおいても同様である。超大規模データ処理アプリケーション技術としてはDigital Sky Surveyとして天体観測データをデータベースとして保持し、世界中の研究者が各地のデータにアクセスすることで処理を行うものである。また、合成開口レーダの実時間データ処理などがあり、今後データ量はまさに天文学的数字となる。ネットワーク利用基盤技術としてはギガビットネットワークにおけるプロトコルとサーバの構築、ユーザが利用するクライアントライブラリ開発、ネットワークコンピューティングを記述するための言語開発、遠隔データベースへのアクセス技術開発、負荷分散と可用性、信頼性向上技術開発などがあげられる。

3.3.1.4 ペタフロップスコンピューティングのアプリケーション

専用加速装置、超並列システム、ネットワークコンピューティングから構成されるペタフロップスコンピューティングシステムにおける想定されるアプリケーションとしてバーチャルマイクロスコプを実例として述べる。

(1) バーチャルマイクロスコープ全体概要

バーチャルマイクロスコープとは次のようなシステムである。分子軌道法と分子動力学法によれば、2つの方法を結合することにより、実験情報を全く使用せずに分子や原子の化学的振る舞いをシミュレーションすることが原理的に可能である。このような手法を用い化学プロセスをコンピュータ上で実現し、コンピュータグラフィックスで視覚化することにより、実験では観察不可能な化学現象をあたかも“顕微鏡”で見ているように観察することが可能である。このような、分子化学総合システムをバーチャルマイクロスコープと呼ぶこととする。このようなシステムは世界的に研究段階にあり、その実現に必要な技術は確立していない。さらに、本システムの特徴としては並列処理、分散処理、オブジェクト指向ソフトウェアなどの計算機技術の向上を十分に取り込んで開発を行うため、ライフタイムの長いソフトウェアとなることが期待されている。このバーチャルマイクロスコープ・システムは、バーチャルマイクロスコープ制御部、ヒューマンインタフェース部、解析部より構成される(図1)。

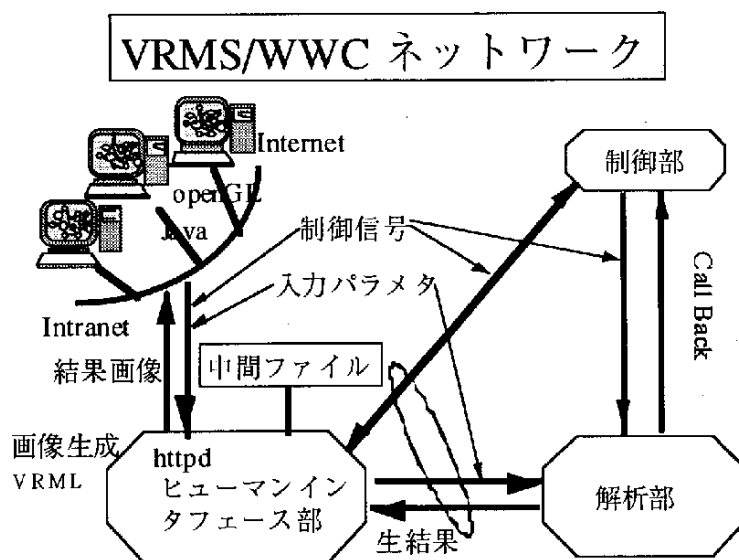


図1

(2) バーチャルマイクロスコープ制御部

バーチャルマイクロスコープ制御部は、システム全体をバーチャルマイクロスコープとして機能させるための制御部であり、ヒューマンインタフェース部と解析部との動作を整合・制御する機能、ヒューマンインタフェース部と解析部とのデータの流れを制御する機能、計算結果をリアルタイムで表示するための解析部を制御する機能を有する。これらの機能を実現するためには、それぞれ、イベントの制御技術、データの流れの制御技術、人間の認知時間での表示技術が不可欠である。現象を追跡しながら、ある時点で現象をゆるやかに進行させたり、時間を戻したりするなどのロールバックが必要となったり、ある時点で解像度を変

えて再計算を行ったり、観測・認識したい現象の呈示手法を即時に変更したり、さまざまな要求が解析部とヒューマンインタフェース部の相互を往復する。

(3) ヒューマンインタフェース部

ヒューマンインタフェース部は、解析条件の入力、分子構造の作成および分子構造と計算結果の物理量（電子密度分布の時間変化等）の表示を行う。分子構造の作成や分子構造の表示を対話的に行い、必要な情報（計算結果、分子構造等）を、必要な形態（3次元グラフィックス、グラフ、2次元グラフィックス、数値等）で、必要なときに表示する。これにより、あたかも実際の顕微鏡を操作する感覚で、実際には観察不可能な物理量を観察可能とする。

(4) バーチャルマイクロスコープ解析部

解析部は、第一原理分子動力学法を中心とした計算システムである。この計算システムは、分子軌道法の中で、分子の安定構造の予測など基底状態の計算に有効であるSCF法と密度汎関数法から構成される。いずれも電子の波動関数を求めるHartree-Fock法と原子核の安定位置を求めるエネルギー勾配法からなる。第一原理分子動力学法は、エネルギー勾配法により計算された原子核に働く力を用い、分子動力学により、化学反応における電子分布および分子骨格の経時的変化を計算する。原子核に働く力を第一原理計算で求め、分子動力学法により時間変化を計算する技術は世界的にも研究段階にあり、実現していない。

(5) ペタフロップスシステムでの実現

バーチャルマイクロスコープの各部はそれぞれに大規模計算を必要とする。特に解析部は現在のスーパーコンピュータ（SX-4）においても1,000軌道程度の計算を実現するのに数時間を要する。実用的には数万軌道の計算を数秒の単位で実現することが可能でないとあたかも顕微鏡を見るようにバーチャルマイクロスコープを実現することはできない。2電子積分と呼ばれる主たる計算部分の計算時間は基底数の4乗のオーダーに比例する。ところが逆にこの部分は専用ハードウェア化が可能でもある。したがって、計算能力で約100,000倍以上が求められるので100ギガフロップスが要求される。これはネットワークコンピューティングにおいて10ギガフロップス級のシステムを10台程度接続することを意味する。各構成部は高速ネットワークにより接続され、制御データ、解析部からの生データ、入力パラメータ等がやりとりされる。さらに出力として生成されたグラフは現状技術で言うところのOpenGLやVRML、Javaのようネットワークコンピューティングと密接に関連した技術を通じて共有可能となる。こうしたことからバーチャルマイクロスコープ技術の延長上に先に述べたペタフロップスシステムが適合することが期待される。

3.3.2 ペタフロップスマシンとその応用 (中島克人委員)

3.3.2.1 はじめに

進歩の早い情報処理関係の技術および市場の予測をするのに、20年後の議論は困難であり、徒労に帰する確率が高いので、ここではおよそ10年後ぐらいのレンジで、ペタフロップスマシン（またはペタフロップススケールのマシン、以後同様）の未来を予想することにする。このレンジで議論する限り、ペタフロップスマシンは少なくとも「高価で稀少なマシン」という位置付けで捉えられ、例えば「身近なPCが全てペタフロップス」という想定は除外される。というのは、ペタフロップスマシンの実現は専用のものでも5年以上[1]、汎用では20年かかる[2]と予想されているからである。

3.3.2.2 ペタフロップスマシンとは何か

さて、10年後でさえも「高価で稀少なマシン」であるペタフロップスマシンの開発について、米国は早くも検討を開始しているのであるが、これを我々はどう見れば良いであろうか？

ピラミッド説

エジプトの巨大ピラミッドの建造理由は王権の強化のためと言われている。圧倒的な国王の威厳の象徴であるばかりでなく、当時の最新の建造技術とその建造を整然と遂行させる強固な中央集権組織のデモンストレーションでもあったと想像されている。しかし、建造後のピラミッド自身は（当時許されたかどうかは定かではないが）物見櫓程度にしか使えず、主に「王権のシンボル」という無形の精神作用のためだけのものとなる。国家財政を破綻させかねないような莫大な建造費用と年月をかけるピラミッド建造が何百年も続けられたということは、その精神作用にそれだけの価値があったわけである。

ペタフロップスマシンとは米国科学技術の優位性を象徴するためだけのピラミッドなのであるか？

宇宙開発説

1960年代にソ連と米国による宇宙開発競争が繰り広げられた。莫大な国家予算をつぎ込むことができた理由は幾つかあろう。冷戦時代における兵器開発競争が第1の理由に違いない。しかし、ピラミッドと同様、国家威信をかけた面も大きかったと見られる。ピラミッドの場合と（恐らく）異なるのは、民衆が宇宙開発の結果に夢を持てたことで

あろう。宇宙旅行や地球外生命との遭遇等のロマン(?)の実現以外に、実利として、地球外資源の獲得、新材料の発見・開発、地球外植民地の開発などの夢があったはずである。

しかし、1969年に月への有人飛行が実現して30年近く経たというのに、宇宙開発は当初想像されたようなスピードでは進んでいない。1960年代のペースで開発が進めば、今日では火星への有人旅行ぐらいいは(着陸できるかどうかは別問題として)できているような気がする。開発ペースが落ちたのは、もちろん冷戦の終結という理由もあったが、それ以前に、その膨大な予算を支えるだけの経済力と世論の支持が続かなかったからと言える。しかし、大きな夢やロマンの早期実現を諦めたかに見える傍ら、スペースシャトルを活用した実利を目指した宇宙開発はそれなりに進展した。また、人工衛星打ち上げが商用ベースに乗り、通信衛星の静止軌道上での場所争いに発展しそうなほど、宇宙利用は急速に伸びている。ペタフロップスマシンは米国科学技術の裾野を広げるための戦略的先陣部隊なのであろうか?

米国でのペタフロップスマシン開発の動きを、筆者は後者の宇宙開発を小規模にしたものとするべきだと考える。ペタフロップスマシンの開発費は宇宙開発に比べれば恐らく「無視できるぐらいに小さい」であろうから、米国にとっては大々的に世論の支持を取りつける必要はなかろう。ただ、スポンサー(例えば米国DOEなどの省庁)がその投資に対して周囲の理解を得られるだけの理由を提示できればよい。冒険を許容し、欠点よりも長所を重視する米国の国民性ゆえ、ペタフロップスマシンの社会的貢献に期待をかけて、意外とスムーズに開発が進む可能性がある。

日本国民は技術開発に関して、かなり保守的であるといえよう。日本の宇宙開発の立ち遅れをペタフロップスマシンに当てはめると、日本が米国のペタフロップスマシン開発を愚行であると嘲笑している間に、その陰でテラフロップス級のマシンの利用とその技術を米国が蓄積し、わが国はますます米国と地力の差をつけられてしまうというシナリオも考えられなくもない。

以上、ペタフロップスマシン開発の政治的もしくは社会的な目的や位置付けを考察したが、どのような開発も現在もしくは将来のニーズ(=応用問題)の無い所に成果はあり得ない。以降では、ペタフロップスマシンが活かされるであろう応用問題に関する分析を試みる。

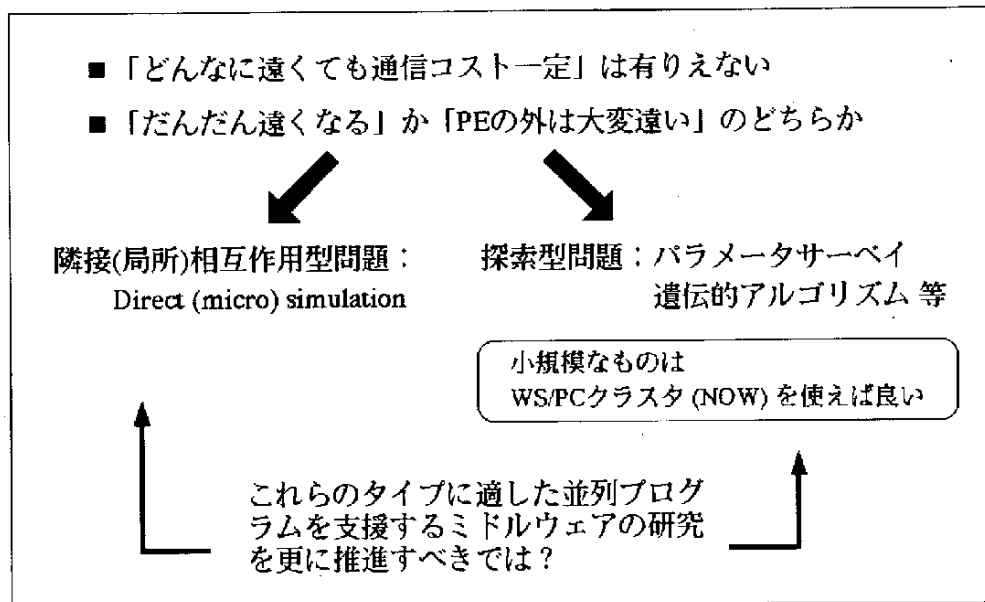


図1 PE間通信による応用問題の分類

3.3.2.3 PE間通信による応用問題の分類

まずはアーキテクチャ、具体的にはPE (Processing Element) 間通信コスト (レイテンシおよびバンド幅) が、ペタフロップス級の並列処理への適不適という意味で応用問題を選別するのではないかという仮説に基づき考察する。というのは、プロセッサ間がどんなに離れていても、「通信コストは一定である」ということは物理的に考えられないからである。

システムの直径が数十を超える並列マシンのPE間通信に関して、大きく分けて次の2つのどちらかに分類されると筆者は考える。

●「だんだん遠くなる」

システム内の隣接PE間では通信は速いが、遠方のPEとはその距離に応じて通信コストが大きくなると考える。多段バス結合に代表されるように、小システムを束ねて階層的に大きなシステムにまとめられたものはこの範疇と見なす。

このようなマシンモデルに適する応用問題の代表的な『タイプ』としてDirect (micro) simulation のような隣接 (局所) 相互作用を扱うものが挙げられる。プロセッサの隣接性と問題空間の隣接性を対応させるべきであることはいうまでもない。なお、N体問題に対する有名なBarnes-Hutアルゴリズムは、本来は大域的な通信と計算が頻繁に必要な所を、それらを隣接相互作用問題に近い計算オーダに下げたという意義があり、このタイプに属するものとして考えることができる。

●「PEの外は大変遠い」

PE内にも厳密には1次キャッシュ、2次キャッシュ、主記憶などの多段のメモリ階層があるが、PE内のデータアクセスに比べてPE間の通信コストがはっきりとした段差を感じさせる程大きい場合である。ノード内を共有メモリバス結合、ノード間をメッセージ通信ネットワークで接続した場合のノード内外の通信コストの違いはこの範疇と見なす。

こちらの場合は、計算間の同期頻度が非常に少なく済む代表的な『タイプ』として、探索型問題が挙げられる。工学的応用としては最適設計のためのパラメータの組合せを多数回のシミュレーションから求めるパラメータサーベイが代表例である。パラメータの組合せだけのデータ並列性があり、総当り、遺伝的アルゴリズム（例えば島モデル並列）、シミュレーテッド・アニーリング（例えば温度並列）、ランダムウォーク（例えば方向並列）等、種々の並列化可能なアルゴリズムが存在する。

なお、このタイプの問題においては、その問題規模が小さい場合にはワークステーション(WS)やPCをLANで接続したようなWSクラスタやPCクラスタでも十分であることが多く、専用並列マシンに対するコストパフォーマンス上の強敵となる。

3.3.2.3 PE間通信のタイプに即した並列処理ミドルウェアに関する提言

さて、これらのペタフロップス・スケールのマシンに期待がかかる問題の『タイプ』に対して、その並列プログラミングや並列実行を適切にサポートする基盤的、共通的ソフトウェア技術が健全に発展しているであろうか？

「PEの外は大変遠い」場合に関しては、WS/PCクラスタを1つの並列システムと見なし、バッチジョブモニタに相当するツールの研究[3]やその製品化[4][5]などがすでに相当数見受けられる。探索型の用途に限定しなければ、LANまたはWAN上に分散された科学技術計算ライブラリを使い分けるシステム[6][7]なども現れつつあり、実用化も進みつつある。しかし、ユーザから見た場合の標準化や情報共有といった面ではそれぞれがバラバラの状態と言えよう。米国においては、旧称DIS (Distributed Interactive Simulation) Workshopと呼ばれていたSimulation Interoperability Workshop[8]では分散環境下での異種シミュレータ統合の仕組みの標準化に関して議論を始めているが、これもつい最近の話である。

「だんだん遠くなる」の場合は種々のアルゴリズムは個別に提案されているものの、Direct (micro) simulation を直接的にサポートするような研究[9][10]はまだわずかなように見受けられる。

並列言語の研究は比較的盛んであるが、上記のような応用問題への橋渡しを行うもう一つのレイヤー、すなわち、並列処理用ミドルウェア（開発支援ツール、実行管理モニタ/スケ

ジューラ、実行時ライブラリ等)の研究開発とその利用普及が相対的に遅れていると見られ、この分野に組織的に取り組むことは、技術発展だけでなく、新市場創出の可能性という意味でも大変有意義だと考える。

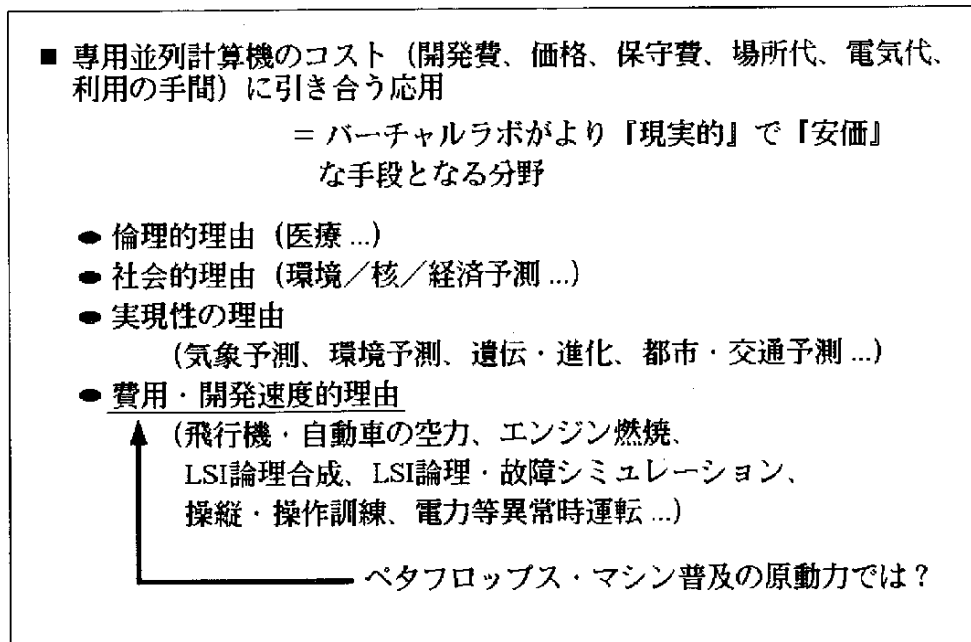


図2 必要理由（動機）による応用問題の分類

3.3.2.5 必要理由（動機）による応用問題の分類

応用サイドに立つ時、高価で恐らく手間のかかる並列マシンをわざわざ利用する動機となるのは、多くの人が指摘しているように、『コスト』的に引き合う場合である。マシンを利用するための『コスト』としては、購入／レンタル価格（場合によっては開発費）、保守費、場所代、冷却も含めた電気代などの直接の費用の他に、プログラムの書き換えの手間や試行錯誤を繰り返す手間などが含まれる。

この『コスト』に引き合う理由をブレークダウンすることにより、どのような応用がペタフロップスマシン開発／利用のドライビングフォースとなるかが見えて来よう。そして、例えば並列処理ミドルウェアの研究開発を考える場合は、その方向が定まってくるのが期待できる。

(1) 倫理的理由

主に医療分野である。人体実験はもちろん、最近では動物愛護運動などの影響で動物実験さ

え自由にならない場合がある。倫理的に許されない人体実験に相当するような医療行為で、計算機でのシミュレーションで代行できるものとして、仮想手術、仮想医薬品（複合／過）投与、配偶関係による遺伝的障害の予測などが挙げられよう。ただし、これらは人体モデル自身が十分に解明されていないため、10年後程度ではそれ程の進展はないかも知れない。なお、訓練としての手術シミュレーションだけは現在の技術でも十分可能と思われる。ただし、ペタフロップススケールの計算量は必要ないかも知れない。

（2）社会的理由

実験しようと思えばできないことはないが、それが社会的に許されないために、シミュレーションが望まれる一群の問題がある。例えば、環境汚染の進行予測、核実験、核兵器の保存条件と放射能漏れの危険性との関係、金融政策と株価変動などの経済問題などが挙げられよう。

（3）実現性の理由

実際に実験することができないために計算機上でシミュレーションを行わざるを得ないものもある。地球規模の環境変化（含む、汚染）予測、気象予測などが物理的な大きさの点で実験不可能である。時間スケールの実験不可能な生命進化予測なども挙げられよう。なお、都市計画や大規模道路網における交通予測も実験は現実的とは言えず、シミュレーションが唯一の手段といえよう。

（4）費用・開発速度的理由

実験は技術的に可能であるし、昔は実際に実験を行っていたが、現在は高速な計算機のお陰で、費用的にも、時間的にもシミュレーションが効率的であるという応用が沢山現れてきている。これらは産業界における開発競争という強い動機が作用しているために、今後ますます進展するのは確実である。

この範疇に入る代表例は航空機や自動車の空力特性解析、エンジン燃焼解析、LSI設計における論理合成、論理／故障シミュレーション、機械・構造の強度解析、熱伝導／放熱解析、などなど枚挙にいとまがない。訓練用シミュレータとしてはパイロット用のフライトシミュレータが老舗であるが、原子力発電所等の管制センタの操作員訓練シミュレータも、異常時に備えての訓練も可能なために一般的になってきているようである。

3.3.2.6 全く新しい並列応用としてのデータマイニング

前節での考察は言わば仮想実験室（バーチャルラボ）としての並列マシン利用を想定して

来たが、計算機にしかできないことを行う応用も出てきている。しかもビジネス分野での応用である。それは米国でこの3～4年で急速に発展しつつあるデータマイニングである。

ディスクの価格低下と産業界の情報化の進展を背景に膨大なデータが蓄積されつつあり、その活用を動機としてデータマイニング技術とその応用が米国を中心として注目を集めている[11]。大量のデータの中から項目（または属性）間に隠された規則性を見つけ出し、それを経営・販売や製造プロセス改善に活かすことが目的であるが、データは今やテラバイト級に及び、ビジネス分野での大規模並列応用の本命の一つと目されている[12]。

従来の統計的手法による解析では、オペレータの勘や経験を元に、注目すべき複数項目を選択し、それらの相関などを求めたり、グラフ表示したりするということが行われてきた。知識発見を標榜するデータマイニングでは、項目の絞り込みを予め行わないことがポイントである。多数の項目間のそれぞれの組合せの検査はシステム側の責任であり、その組合せを旨く刈り込んで行くことがデータマイニングアルゴリズムの特徴となる[13]。

しかしながら、レコード数の増大により計算量が線形以上に増大するのはやむを得ず、レコード数を N とすると、良くて $O(N \log N)$ 程度の計算量が必要となる[12]。したがって、この部分がペタフロップスマシンの威力の発揮しどころとなる。多額の金銭に影響する経営方針の意志決定支援を用途とする場合は、例えばペタフロップスマシンの使用に数億円以上かかろうとも、その利用は促進されるであろう。

3.3.2.7 まとめ

ペタフロップス（あるいはそのスケールの）マシンの将来について、そのシーズとニーズの画面から予測と分析を試みた。

シーズとしてはもちろん半導体デバイスの技術の進展もあるが、特に米国の世界一たらん政策的動機、フロンティア精神等の社会的風土が開発の促進剤として作用することが考えられる。良い面を評価する米国と異なり、平均点を判定基準としがちで、時には減点主義に陥る国民性を持つ日本は、ペタフロップスマシンだけでなく、その開発の陰で進展するであろう重要な技術さえも見逃す危険性を持っている。冷静な分析と多様性の確保を良しとする風土の醸成が望まれるところである。

ニーズとしては、バーチャルラボ的な応用が本命となり、特に費用・開発速度的な面でペタフロップスマシン利用が有利となるものが強力な原動力となるであろう。ただし、PE間通信が軽い探索型もしくはパラメータサーベイ的な応用でかつ規模がそれほど大きくないものについてはWS/PCクラスタがもっぱら利用されるため、ペタフロップスマシンの優位性はより大規模か、即時性の要求が強い応用に限られよう。また、大容量・高速マシン利用なら

ではのデータマイニング等の新しい応用も生まれつつあり、ペタフロップスマシンの陰に隠れて、例えば1世代前の枯れた技術となるであろうテラフロップスマシンの利用がどんどん広がっていく可能性も見逃せない。

仮想実験室の活用を思い浮かべれば分かるように、計算機を使いこなす力が技術力の差に現れる。実験精度を考えれば、単一のマイクロプロセッサではできなくて、並列マシンでこそできる実験が常に存在する。並列マシンを使いこなせるかどうかの一つのバリアになっているだけに、これを越えた者と越えない者には明確な差が現れよう。わが国は今回も米国が越えた後にロープを垂らして貰うのであろうか？ それとも、組織力を動員して自力で、もしくは、国際的に協力しつつ這い上がるのであろうか？

<参考文献>

- [1] Petaflops Architecture Workshop, DRAFT Findings and Recommendations, 1996.
(<http://www.aero.hq.nasa.gov/hpcc/petaflops/petasoft96/paws.outcome/findings.html>)
- [2] Applications and Algorithm, Challenges for PetaFLOPS Computing, 1995.
(<http://www.mcs.anl.gov/summer.study/index.html>)
- [3] 白石 将、佐藤 裕幸、中島 克人：分散型並列処理支援ツールParaJET、電子情報通信学会技術研究報告 CPSY96-60、 pp.23 -- 30, Aug. 1996.
- [4] <http://www.hp.com:80/wsg/ssa/task.html>
- [5] <http://www.platform.com/>, <http://www.comtec.daikin.co.jp/lsf/LSF-1.html>
- [6] S. Sekiguchi, M. Sato, H. Nakada, S. Matsuoka and U. Nagashima : - Ninf -: Network based Information Library for Globally High Performance Computing, In Proceedings of Parallel Object-Oriented Methods and Applications (POOMA) , Santa Fe, Feb. 1996.
- [7] <http://phase.etl.go.jp/ninf/TechnicalNote.html#sekiguchi96>
- [8] <http://siso.sc.ist.ucf.edu/>
- [9] 古市 昌一、他 : A Space-time Object :An Object Oriented Model for Parallel Simulation, Object Oriented Simulation Conference, pp.86--91, 1996.
- [10] 阿部 一裕、他：超並列オブジェクト指向シミュレーション環境 OSim, In Proc. of JSSST Workshop on Object Oriented Computing, S3-2, ISSN 1341-870X, 1996.
- [11] 森末 真一：データマイニング最前線. ACM SIGMOD日本支部 第7回大会講演論文集, pp.39 -- 48, Nov. 1996.
- [12] V. Kumar, A. Sameh, A. Grama and G. Karypis : Architecture, Algorithms and Applications for Future Generation Supercomputers. In Proceedings of the Sixth Symposium on the Frontiers of Massively Parallel Computing, pp.346 -- 354, Oct. 1996.
- [13] R.Agrawal and R.Srikant : Fast algorithms for mining association rules. In Proceedings of the 20th VLDB Conference, Sept. 1996.

3.3.3 PetaFlops計算機の適用分野とアプリケーションからの要望 (福井義成委員)

3.3.3.1 PetaFlops計算機のニーズ

PetaFlops計算機が必要な分野には、

- (1) ミクロな原理からマクロな現象を解明する場合
- (2) リアルタイム性が要求される分野
- (3) 多くのケースの計算が必要な場合

のようなものがある。以下で詳細について説明する。

(1) ミクロな原理からマクロな現象を解明する場合

例えば、原子を支配する基本法則（第一原理）だけから物質の特性を計算して求める場合などがある。原子の基本方程式を対象とする原子数に対して計算し、そこから物質としての性質を求める分野である。

現在では原子レベルからの計算も計算機的能力からいろいろな仮定をおいて計算を進めることが多いが、それでは、仮定条件以外の答が出てくる可能性はない。なにも仮定をおかない第一原理からの計算ではこれまで人間が考えつかなかった答が出てくる可能性がある。この場合、扱える原子数がどれくらいかで扱える現象の範囲が決定する。例えば、1原子しか扱えない場合は、その原子が純粋な塊（多くの場合、結晶）でどの方向へも無限に続いている場合しか表現できていない。多くの有用な現象は、同じ原子が無限に続いている場合ではない（現実的にもそのようなことはあり得ない）。多くの場合、有用な現象は物質の表面や2つの性質の異なる物質の境界面でおきる。触媒の場合は物質の表面であり、半導体では2つの性質の異なる物質の境界面である。このような場合、小さな原子数しか扱えないと、出てくる現象は小さなものだけである。

現在の計算機的能力では、関係する原子の数が10～100のオーダー程度が無理なく計算できる範囲である。1モルの分子数は 10^{23} のオーダーの原子数であり、その差は大きい。 10^{23} 個の原子を扱うのは、現在は不可能である。すべての欲しい物質の性質を出すのには、 10^{23} のオーダーの原子数は必要はないが、現在の計算機の少なくとも1,000倍以上の能力は必要である。

(2) リアルタイム性が重要な分野のシミュレーション

シミュレーション結果のリアルタイム性が重要な分野では実時間よりも速くシミュレーション結果を得て、対応策を検討・適用するため、高速な計算機が必要になる。計算結果を得ることは現在の計算機でも可能であっても、リアルタイム性を実現するためには、PetaFlops計算機が必要になる場合であり、気象予測、地震予測、原子力発電所の制御などが考えられる。また、化学反応、溶鉱炉、半導体製造などで、製造段階の情報からシミュレーションを行い、結果を予測し、その後の工程を制御することにより、品質向上などが可能になる。

(3) 多くのケースの計算が必要な場合

1 ケースの計算にはペタフロップス計算機は必要ではないが、シミュレーションのケース数が数千から数万ケースの場合がある。数多くの設計パラメータの組み合わせに対して性能の検証が必要な場合である。この場合、個々の計算ではPetaFlops計算機は必要ないが、数多くの計算結果をまとめるため、統合化されたシステムでないと不便である。

上記の用途以外でも、PetaFlops計算機は必要となると予測する。かつて、IBM7090やCRAY-1が出現した時、世界中で数台あれば、すべての計算が可能と言われてことがあったが、実際にはその予想をはるかに超える台数が使用された。これは高性能計算機が実現したことにより、当初想定していた用途以外にも新たな用途が出現したことを意味している。PetaFlops計算機でも同じことが起こると考えられる。そのためには、多くの人に利用可能であり、アプリケーションから見て有用なものであることが重要である。

3.3.3.2 モデル記述

計算機のユーザが問題を解決するために計算機を使う場合、モデル（プログラム）記述が容易なことが大切である。ユーザにとって、演算が速いことではなく、解きたい問題が速く解けることが大切である。PetaFlops計算機でのモデル（プログラム）記述はどのようにするのが、望ましいのか？

ここでは、PetaFlops計算機は並列計算機と仮定する。現在、普通に使われている手続き型言語（Fortran、C 等）はノイマン型計算機を前提にしている。プログラム（モデル）はどのデータをどのように加工するかの手順を計算機に指示することにより構成されている。現在のアプリケーションも多くのものはアルゴリズムもノイマン型計算機を前提にしている。手続き型言語での並列プログラムの作成はユーザにとって負担が大きい。並列計算機では、

単独プロセッサでは問題とならない複数プロセッサ間のタイミングの問題も生じる（図1）。タイミングにからむ問題は再現性が容易でないことが多く、これが手続き型言語での並列プログラミングの困難さの一因である。さらに、現在の並列計算機は各計算機ごとに性能を発揮させるための手法が異なっていることが多い。ユーザにとって、そのための負荷も大きい。

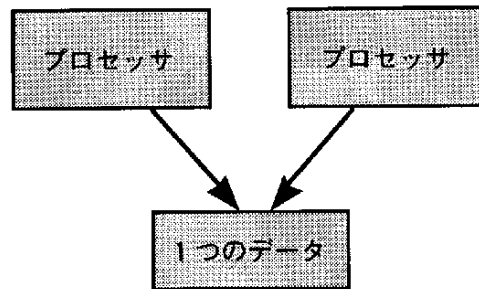


図1 複数プロセッサでのデータの競合

並列計算を使い易くするためには、並列計算機向きのモデル記述・アルゴリズムがあるべきである。ユーザにとって、手続き型言語で並列プログラムを書くには負荷が大きく、それが可能な人口は限られてしまう。その点からも並列処理のユーザを拡大するための障害になってしまう。データ並列はベクトル化と本質的に同じであり、問題解決にはならない。

ユーザにとって、モデル記述が容易で、モデルの変更も容易で、なおかつ、並列性の抽出が計算機システムにとって容易であることが望ましい。

並列処理には非手続き型でモデル（プログラム）を記述する方法も解決のための1つの方策である。非手続き型言語とは、プログラムを手続きではなく、関数型ないしはオブジェクト指向的に因果関係だけを任意の順序で記述することである（図2）。関数型ないしはオブジェクト指向的な非手続き型言語で、並列プログラム（モデル）を記述することは人間の思考にあっており、非常に容易である。制御系でよく使用されるブロックダイアグラムが良い例である（図3）。

手続き型言語では

$$\begin{aligned} x &= y + z \\ q &= x + r \end{aligned}$$

と

$$\begin{aligned} q &= x + r \\ x &= y + z \end{aligned}$$

では結果が異なるが、
非手続き型では
同じ結果になる。

手続き型言語では

$$a = a + 1$$

は意味があるが、
非手続き型言語では
エラーとなる。

図2 手続き型言語と非手続き型言語

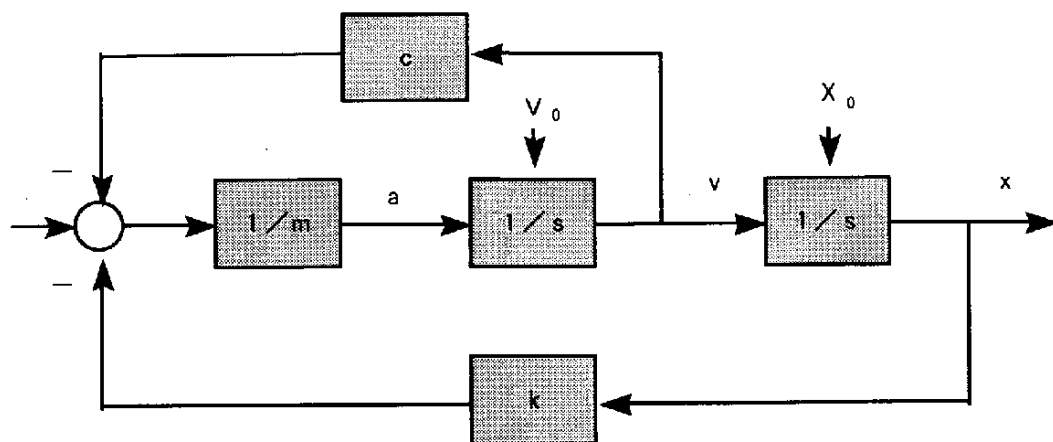


図3 振動子のブロックダイアログ

実例として、制御系でよく使用されている連続系シミュレーション言語の例を示す。連続系シミュレーション言語はブロックダイアグラムをそのまま計算機で記述することを目的としている。連続系シミュレーション言語はアナログ計算機と同じ発想である。

連続系シミュレーション言語はユーザが記述したモデルの演算順序を自動的に決定してくれる。連続系シミュレーション言語はノイマン計算機で、データフロー計算機をシミュレーションしていると考えることができる。さらに、連続形シミュレーション言語では、各問題によって内容が異なるモデルの部分は、上記のような方法で容易に記述できる。さらに、常微分方程式の初期値問題を解く部分は共通化でき、共通化できる部分は各計算機の性能を十分発揮できるようにすることは困難ではない。(連続系シミュレーションは必ずしもPetaFlops計算機を必要としないことが多いが、非手続き型言語の例として使用した。)

例として、重りとバネから構成される振動子を考える。 m を質量、 k をバネ定数、 c を減衰係数、 x を物体 m の変位、 t を時間とすると、振動子は以下のような常微分方程式の初期値問題として記述できる。

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0$$

これを標準形式に変換する。 v を速度、 a を加速度とする。

$$\frac{dx}{dt} = v$$

$$\frac{dv}{dt} = a$$

$$m \cdot a + c \cdot v + k \cdot x = 0$$

さらに積分形式に変換する。 x_0 は x の、 v_0 は v の初期値とする。

$$x = \int_0^t v \, dt + x_0$$

$$v = \int_0^t a \, dt + v_0$$

$$a = -(c \cdot v + k \cdot x) / m$$

これを計算機に入力しやすいように、文法を定め、以下のように表現する。

$$X = \text{INTGRL} (X_0, V)$$

$$V = \text{INTGRL} (V_0, A)$$

$$A = F/M$$

$$F = -C \cdot V - K \cdot X$$

この表現を連続系シミュレータの前処理を行うと以下のようなFORTRANに変換される。

$$ZZ0002 = V$$

$$F = -K \cdot X - C \cdot V$$

$$A = F/M$$

$$C \quad V = \text{INTGRL} (ZZ0001, A)$$

$$C \quad X = \text{INTGRL} (X_0, ZZ0002)$$

振動子の例は小さなものであるが、この例のようにモデル（プログラム）の記述が任意のところからできる言語であると、モデルの記述性が非常に向上する。手続き型言語ではモデルの計算順序を使用者が考慮しなければならないが、小さなモデルでは可能であるが、大きなモデルでは不可能に近い。計算順序を間違えると位相ずれとなって現れる。

3.3.3.4 上位レベルでの問題記述

上記の非手続き型言語でのモデル記述も同じことであるが、並列化（高速化）のためには、ミクロなレベルではなく、よりマクロなレベルで記述できることが重要である。例えば、行列と行列の積を考える。手続き型言語（FORTRAN、C等）では

```
do 300 i=1,n
  do 200 j=1,n
    s=0.0
    do 100 k=1,n
      s=s+a(i,k)*b(k,j)
100    continue
    c(i,j)=s
200    continue
300  continue
```

図4 行列積のプログラム例

のようなループを記述することになり、計算機システムやコンパイラーは与えられたプログラムからベクトル化や並列化の可能性を抽出しなければならない（この例は非常に簡単なもので問題にはならないが、実際の計算では非常に複雑な計算式となり、ベクトル化の可能性や並列性の抽出が容易ではない）。それに対して、行列と行列の積という記述を計算機に与えることができれば、計算機システムは並列化することは非常に容易になる。例えば、APL言語では行列演算が非常に簡単に記述できる。

マクロレベルでの問題記述をするため、別角度から計算の手順を考える。計算機でのシミュレーション手順（現象から計算結果の認識まで）を考えると図5のようになる。

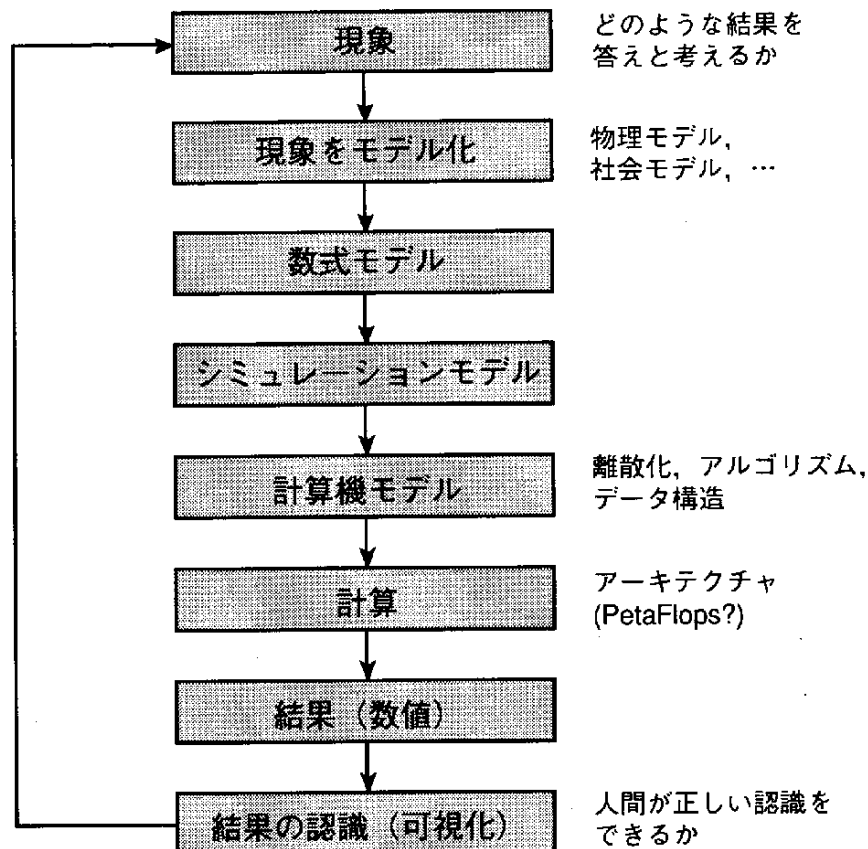


図5 計算の手順

図5でPetaFlops計算機が有効なのは、計算の部分であり、計算を効果的に進めるためには計算の前後の作業が効率的に行われなければ、全体として良い結果は得られない。特に、Peta Flops 計算機で計算の部分が高速化されることにより、計算の前後の使い易さがますます重要になる。PetaFlops 計算機の性能を十分発揮させるためには、上位レベル（マクロなレベル）での工夫が重要である。以下で簡単な例を示す。

この例は残念ながら並列処理の例ではなく逐次処理の例である。しかし、マクロなレベルで問題を記述することにより、並列処理も問題の並列性を抽出し易くなり、計算機ごとの特性も利用し易くなる（専用ライブラリの整備などで実現）ことは逐次処理も並列処理も同じである。一般的に高速化のレベルを考えると、以下のようになる。

1. ムダな計算を削除し、コーディングを変更する
2. 計算を行う場合のデータ構造を変更する
3. 問題を解くアルゴリズムを変更
4. 問題のモデル化そのものを変更

上記の各部分に対応する高速化の工夫をするとどのような効果があるかを簡単な例(下記)で示す。

$$S = \sum_{i=1}^{10000} (-1)^i \times i$$

最も単純にプログラムを作成するとプログラム 1 のようになる。

★プログラム 1

```
S = 0.0
DO I = 1, N
    S = S + (-1) ** I * I
CONTINUE
```

計算の中で冪乗を計算しないように工夫するとプログラム 2 のようになる (1 を適用)。
この工夫では約 4 倍速くなっている。

★プログラム 2

```
S = 0.0
A = 1.0
B = 0.0
DO I = 1, N
    A = -A
    B = B + 1.0
    S = S + A * B
CONTINUE
```

さらにループ内での乗算を避ける工夫をするとプログラム 3 のようになる (1 を適用)。
この工夫の効果は約 5 倍であり、プログラム 2 と比べて約 1.25 倍速くなっている。

★プログラム 3

```

S = 0.0
A = -1.0
DO I = 1, N
    A = A + 2.0
    S = S - A
CONTINUE
A = 0.0
DO I = 1, N
    A = A + 2.0
    S = S + A
CONTINUE

```

元々の問題を良くみると、計算結果はプログラム4のような簡単な式で計算できることが分かる（3を適用）。これを実行するとプログラム1に比べて、約9,000倍も速く計算できる。

★プログラム 4

```

IF (MOD (N, 2) .EQ.1) THEN
    S = S - N
ELSE
    S = N / 2
ENDIF

```

計算速度比は、あるスカラー計算機の場合、下記のようにになる。

プログラム	計算速度比
1	1 倍
2	4.06 倍
3	5.06 倍
4	9,606.16 倍

この例は非常に簡単な例であるが、プログラムレベルでの高速化はドラスティックな改善は不可能であり、ドラスティックな高速化はアルゴリズムの変更、モデル化の変更等でなければ

不可能であることを示している。図5の上の方へ行くほど、その効果は大きい。そのためには、モデル（プログラム）のマクロレベルでの記述が必要になる。

3.3.3.4 まとめ

PetaFlops計算機を使用するであろうユーザが本質的（直接的）に欲しいものは下記のようなものである。

- (1) 速く正しい結果の得られる道具
- (2) 問題の処理過程全体が速くなること

これを実現するためには、PetaFlops計算機のような速い計算機とモデル（プログラム）記述手法（非手続き型、マクロレベルなど）の変更とPetaFlops計算の前後の処理の高速化・高度化である。前後の処理の高速化・高度化には数値的処理だけではなく、非数値処理の高速化・高度化、認識（せまくは可視化）の高速化・高度化も重要である。

また、PetaFlops計算機が実現すると、現在、想定している用途以外にも新たな用途が必ず出現する。それは過去の例が実証している。そのためには、PetaFlops計算機が多くの人に利用可能となり、アプリケーションから見て有用なものであることが重要である。

<参考文献>

- [1] 福井義成、HPCを目指して、情報処理学会HPC研究会資料、93-HPC-46、pp.1-6（1993）。

3.3.4 並列数値シミュレーションとペタフロップス計算機

(横川三津夫委員)

3.3.4.1 計算科学と並列計算機

ここ数年、未知現象の解明や実験不可能な現象の解明、大規模構造物等の詳細な解析等を行うために、数値シミュレーション技術を駆使して研究を行う、理論、実験につぐ「第3の科学」と称される計算科学が注目を集めている。計算科学の進展は、単体プロセッサの高速化、主記憶装置の大容量化、コンパイラなどソフトウェア技術の発展などの計算機技術の発展によってもたらされている。

しかし、現在よりもさらに大規模な問題や、より詳細な解析を行うためには、複数の計算プロセッサをネットワーク結合した並列計算機を用いるのが唯一の解決法と考えられる。現在の並列計算機の動向を見ると、MIMD型のものが主流になっているが、そのプロセッサ・アーキテクチャやメモリ配置を見ると、商用化された並列計算機の種類は千差万別といっても過言ではない。このため、いろいろなサイトに並列計算機が導入されている現状においても、並列計算機の使用方法は未だ確立されていない。ある種の計算を行っていくときに、計算の流れを順序立てて考えれば良かった逐次型計算機やベクトル型計算機を、解析の道具としてきた応用分野の研究者・技術者が、それぞれのアプリケーション問題を解く場合、どういう並列計算機アーキテクチャの上で、どういうプログラミングを行ったらよいかという問題に対する決定的な指針は与えられていないのである。

従来用いているプログラムをそのまま並列計算機で並列実行できないことはもとより、一所懸命手作業で並列化しても、多くの分野の問題でその計算機能力を十分に引き出せないことが分かってしまった。ただ単に、カタログ性能値（理論ピーク性能値）が高いということだけで、アプリケーション・プログラムが高速に実行できるという幻想は脆くも崩れたのである。自動並列化コンパイラが開発段階にある現状では、コンパイラによる高速化の望みもない。今のような状況が続けば、ペタフロップス計算機が開発されたとしても、その性能の1%も発揮できないであろう。

少なくともペタフロップス計算機の開発が始まるまでには、アプリケーション問題や数値シミュレーション手法、そのカーネルとなる計算アルゴリズム等に対して、なんらかの並列計算モデルを構築し、最も適合する並列計算機のタイプや、その並列計算機の上での並列化方法等の技術を検討、蓄積しておく必要がある。そして、応用分野の研究者・技術者に並列化技術を正しく提示することができれば、並列計算機がより有効に使われ、ペタフロップス計算機開発の機運が高まることであろう。

3.3.4.2 並列計算モデル

アプリケーション・プログラムの並列化では、計算処理やデータのプロセッサへのマッピングが重要な問題である。ここでは、並列計算を行うときのデータアクセスパターンや演算処理の方法等を分類することを考えてみる。

いろいろなアプリケーション問題の並列計算モデルへの分類については、問題やそれを解くために用いられるアルゴリズムに内在する並列性を、いくつかのモデルで表現することが試みられている。Pancake [1] は、アプリケーション問題を並列処理するときの実行形態を問題アーキテクチャ (problemarchitecture) と呼んで、4つに分類した。

●Perfect Parallelism (Embarrassingly Parallel)

問題が完全に並列実行可能なモデルである。パラメータ・サーベイや、数種類のデータに対して同じ処理する問題などが含まれ、並列化が簡単に行える。

●Pipeline Parallelism

1つのジョブがいくつかのタスクに分かれて、それがパイプライン的に連結されて処理されるような計算モデルである。処理量の多い部分タスクは、パイプライン処理のボトルネックとならないように相対的に速いプロセッサで処理するなど、タスク間の負荷分散が必要である。

●Fully Synchronous Parallelism

次のフェーズの計算、例えば時間発展問題で1時間ステップの時間積分するような場合に、現時点のすべてのデータの計算結果が必要となる計算モデルである。

●Loosely Synchronous Parallelism

分割されたタスクやデータ量に対して、計算負荷が時間と共に動的に変化する計算モデルである。

科学技術分野の多くのアプリケーション問題では、非定常現象の時間発展シミュレーションを行うことが多く、計算処理全体を見た場合にはfully or loosely synchronous parallelismに分類される。

数々の数値シミュレーション手法の並列化の経験から、時間発展シミュレーション問題の分類では、Pancakeの分類が並列計算モデルとして十分な表現能力がないことが明らかである。1時間ステップ内の計算処理に対して、データのアクセスパターンやデータ転送パター

ンを考慮した並列計算モデルを考える必要がある。そこで、1時間ステップ内のデータアクセスパターンと、データ転送パターンという観点で以下のモデルを考えた。

●RALC (Regular Data Access, Local Communication)

演算がメモリ上のデータを規則的にアクセスし、並列計算を進める上で隣合ったプロセッサ間でデータ通信を行う計算モデルである。

●RAGC (Regular Data Access, Global Communication)

演算がメモリ上のデータを規則的にアクセスするが、並列計算を進める時にプロセッサ全体でデータ転送が必要な計算モデルである。3次元高速フーリエ変換が例としてあげられる。

●IALC (Irregular Data Access, Local Communication)

各プロセッサ内での計算では、メモリ上のデータを規則的にアクセスしないが、時間ステップを進めるときに、隣あったプロセッサ間でのみデータ転送が生じる計算モデルである。

●IAGC (Irregular Data Access, Global Communication)

各プロセッサ内での計算では、メモリ上のデータを規則的にアクセスせず、時間ステップを進めるときに、全プロセッサ間でデータ転送が生じる計算モデルである。

ここで、RALC、RAGCは、規則的にデータアクセスを行う手法のモデルであるため、ベクトル演算器をもつプロセッサに良く適合する。IALC、IAGCは、スカラー並列計算の方が望ましいモデルである。時間発展の各時間ステップは、各プロセッサで一致させる必要があるので、バリア同期を取る必要がある。

表1に、流体計算に用いられる4つの数値シミュレーション手法と分子動力学法を並列計算する場合に、ベクトル計算適合性と領域分割による並列化の時のデータ通信パターンという観点で分類したものを示す。これらの手法の並列計算モデルも同時に示す。有限差分法と格子ボルツマン法は、同じ並列化手法を用いることができ、同じ計算モデルRALCに分類される。DSMC法では、粒子が計算領域全体に移動する可能性を持っているので、全プロセッサが通信対象となる。

表1 各手法の分類

	ベクトル 計算 適合性	領域分割による並列化時の 通信パターン		並列計算モデル
有限差分法	○	隣接プロセッサ間	規則的	RALC
スペクトル法 (フーリエ法)	○	全プロセッサ間	規則的	RAGC
DSMC法	×	全プロセッサ間	不規則的	IAGC
格子ボルツマン法	○	隣接プロセッサ間	規則的	RALC
分子動力学法 (短距離力)	×	Cut-Off距離内 にある プロセッサ間	規則的	IALC

わずか5つの数値シミュレーション手法に対しても、適合する計算機のタイプが違うことを考えれば、ベタフロップス計算機もアプリケーション問題に応じて開発されることになるであろう。

3.3.4.3 大規模数値シミュレーション

ベタフロップス計算機を必要とする大規模数値シミュレーションにおいては、

- (1) パラメータ・サーベイのために、大量のケースの数値シミュレーションを行う場合、
- (2) 長時間の現象を細かい時間間隔で数値シミュレーションを行う場合、
- (3) 計算格子点数を増加させた結果、全演算量が増加する場合

の3つのケースが考えられる。

(1) のケースでは、各パラメータに対して最終的に得られた結果を統合、整理するための処理方法を考える必要があるものの、プロセッサ数が多ければ、大量のケースの数値シミュレーションが同時に実行可能であるので、超並列計算機が極めて有効である。例えば、大気モデルを用いた気象予測のような非線形現象では、いくつかの異なった初期状態から計算を開始し、結果の集合平均をとった確率的な結果が重要な意味を持つ。この場合には、一つ一つのパラメータに対して計算時間が多くなったとしても、完全並列の計算ができるので、高い実行効率が得られる。

(2) のケースでは、時間方向の並列性を見出し計算時間を短縮することが考えられる。通常、非定常時間発展の数値シミュレーションでは、時間微分に関して微小時間 Δt で離散

化し、時刻 t_0 の計算領域の値を用いて、時刻 $t_0 + \Delta t$ の値を求めているが、計算が時系列的に順序正しく実行することを保証するために、1時間ステップ毎に計算の同期を取らなければならない。このため、時間方向の並列性は期待できず、長時間現象の数値シミュレーションの短縮は、1時間ステップ内の計算時間を短くするしかない。しかし、空間的な離散点の数が少ない場合、すなわち1時間ステップ内の計算量が少ない場合は、並列計算による速度向上は望めない。計算量が少なければ高い並列度は得られないため、プロセッサ数を増加させてベタフロップス計算機を構成する方法は意味を持たない。

(3) のケースは、詳細な数値シミュレーションを行うか、計算領域を大きくする場合である。1時間ステップ内の計算において、高い並列化効率が得られることが分かっているならば、並列度を大きくすれば時間短縮が図られる可能性がある。しかし、総演算量が増えているから、全体の経過時間は変化なく、詳細な結果が得られるまでの実時間は変わらない。

以上の結果から、大粒度の並列化を確保しつつ、プロセッサ間の通信性能と計算性能のバランスの取れた数値シミュレーションが並列化効率を保つ上で必要である。また、計算形状が比較的単純な理学的シミュレーションと、実用問題を考える工学的シミュレーションでは、大分状況が異なることも考慮しておく必要がある。

3.3.4.4 数値シミュレーションの問題点

数値シミュレーションの問題点を2つ挙げておきたい。まず、計算誤差の問題である。離散化の精度を上げていくと、打ち切り誤差よりも丸め誤差が優位になっていくのはよく知られた事実である。偏微分方程式の時間微分項を、数値積分する時の誤差の議論は、その刻み幅が小さくなるほど、丸め誤差が顕著になることが分かっているが、定量的な評価があまりなされていない。計算機性能が向上して、単に離散化格子を増加させる傾向にあるが、さらに詳細な解析をするためには浮動小数点数の表現桁数を多くするなど、根本的なアーキテクチャの変更が必要になるとと思われる。

もう1つは、数値シミュレーション結果の取り扱いに関する問題である。すでに知られている現象を、数学モデルを改良したり、数値シミュレーションの精度を向上させたり、また多くの観測データ等を初期、境界条件とすることにより、再現することは可能と思われる。ベタフロップス計算機が開発されれば、かなりの現象が数値シミュレーションによってとらえることができるであろう。

しかし、実験や観測によっても検証不可能な現象に対して数値シミュレーションする場合、数値シミュレーションに用いられた数学モデルが本当に正しいと誰が保証できるのだろうか。また、すべての要因を数学モデルで考慮してないために、予測できない事象が起こりうる可

能性を誰が否定できるのだろうか。不十分なモデルによって得られた数値シミュレーション結果を、ペタフロップス計算機で計算したという理由だけで、その結果が一人歩きして使われてしまうことがないように、数値シミュレーション結果の取り扱いについては、十分議論する必要がある。

3.3.4.5 ペタフロップス計算機

ペタフロップス計算機が出現する頃には、ピーク性能によって計算機の性能を定義する悪習はなくなっていると思う。プロセッサをたくさん並べるだけのペタフロップス計算機では、多くのアプリケーションの実効性能は数10テラフロップスも達成できない。また、Embarrassingly Parallel に属するアプリケーション問題に対して、ペタフロップスの性能を達成したといっても、それは簡単な並列計算が可能であるから、これでペタフロップス計算機というのは不適切だろう。したがって、計算機の実効性能の評価方法を確立する必要があり、やはりアプリケーションごとの実効性能でペタフロップスと称するのがいいのではないだろうか。

さて、科学技術計算用のペタフロップス計算機は本当に必要なのだろうか。研究者や技術者、特に理学的な計算科学研究をしている研究者は、再現なく計算性能を求めている。しかし、ペタフロップス計算機が普及機とならない限り、その開発費は数兆円を超えるものと思われる、開発するならば国主導の開発体制、資金の調達が必要であろう。誰もが納得する適用応用分野が見つけられないうちは、ペタフロップス計算機開発のモチベーションは薄い。

<参考文献>

- [1] C. M. Pancake, "Is Parallelism for You?," IEEE Computational Science & Engineering, summer (1996) .

3.4 デバイステクノロジ

3.4.1 CMOSデバイス開発の将来予測

CMOSデバイスは、これまでスケーリング則を良く満たし、微細化によって飛躍的な性能向上がなされてきた。また、メモリの主要構成デバイスであるDRAMの集積度においても3年で4倍となるMoore's lawが満たされている。

これらの特性により、CMOSデバイスでは他の技術と異なり、非常に見通しの良いロードマップが作成されており、世界中の半導体産業がそれに従い、もしくはそれを実現するために熾烈な研究開発競争を行っている。そのロードマップは、米国のSIA (Semiconductor Industry Association) がThe National Technology Roadmap for Semiconductorsとして発表している。これは、1992年に初めて編纂され、1993年に半導体技術ワークショップの結論として頒布された。現在、1994年分が公表されており、1997年末に次編が公表される予定で、3年に一度改編されることになっている。

この、レポート作成には、米国の半導体技術の全ての分野が参加している。主となる参加者は、SEMATECH、Semiconductor Research Corporation (SRC)、および産業的なコンソーシアム（企業、大学、政府機関 (National Institute of Standards and Technology (NIST) ; the National Science Foundation (NSF) ; 商務省 (DOC) 、国防総省 (DoD) 、エネルギー省 (DOE)) および国立研究所) である。

以下は、主としてこの1994年版ロードマップに記載されたデバイス開発予測をもとに説明する。

3.4.1.1 デバイス性能予測

CMOSデバイス性能を左右する主要なパラメータは、ゲート部形成等の最小微細加工部分のルールである。現在、 $0.35\mu\text{m}$ が広く用いられているが、これが13年後の2010年には $0.07\mu\text{m}$ にまで縮小する。 $0.07\mu\text{m}$ でのCMOSトランジスタ動作は単体では既の実証されており、現在では、nMOS $0.04\mu\text{m}$ 迄の動作が認められている。したがって、後はいかに設計、製造技術を向上させ、LSI動作させるかが課題である。

CMOSデバイスは、主にメモリとプロセッサの二つに分けられる。

(1) メモリ

DRAMの集積度とチップサイズの予測を表1に示す。2010年には64Gbit製品が登場する。

その大きさは、2.8cm×5cmという巨大なものになりそうである。1997年のISSCC (the International Solid State Circuits Conference) でNECが4 G DRAMを発表したが、0.15 μ mルールで986mm²という大面積なものであった。大衆化された商品となるには、1～2世代進んだルールを用いなければならないので、2010年に大量、かつ安価に手に入るのは、4～16G Bit製品であろう。

表1 DRAMロードマップ

暦年	ルール (μ m)	集積度 (MBits)	チップサイズ	
			(mm ²)	(mm×mm)
1995	0.35	64	190	10×20
1998	0.25	256	280	12×24
2001	0.18	1000	420	15×30
2004	0.13	4000	640	18×36
2007	0.1	16000	960	22×44
2010	0.07	64000	1400	28×50

(2) プロセッサ

プロセッサの処理速度を決めるクロックの予測をチップサイズとともに表2に示す。2010年には1.1GHzで動作すると予測されているが、1997年ISSCCでDECが600MHzで動作するRISCプロセッサ（アルファチップ）を発表し、この動きは少し加速されるかも知れない（Intel P6 は400MHz動作を発表）。ちなみに、前回、1991年での予測を外挿すると、2010年では2GHzとなる。

表2 プロセッサロードマップ

暦年	ルール (μ m)	クロック (MHz)	チップサイズ	
			(mm ²)	(mm×mm)
1995	0.35	300	250	16×26
1998	0.25	450	300	18×18
2001	0.18	600	360	19×19
2004	0.13	800	430	21×21
2007	0.1	1000	520	23×23
2010	0.07	1100	620	25×25

3.4.2.2 要素技術の限界

上記のように、CMOSデバイスはスケーリング則にのり、予測どおりの素子が予測どおりの時期に手に入るように考えられているが、それを実現するためには、技術的ブレークスルーをいくつか解決しなければならない。そのいくつかを紹介する。

(1) 微細化

・リソグラフィ

微細化の鍵を握る加工技術であり、光学系とレジストに大別されるが、いずれにせよ使用可能な光源に大きく左右される。現在考えられている光源は、紫外光レーザー (KrF:248nm、ArF:193nm)、電子ビーム (EB)、X線である。微細加工限界は光源の波長程度であるので、2010年の $0.07\mu\text{m}$ を切るためには、EBかX線しかない。EBは微細加工の技術として、かなりの蓄積と実現性があるが、単純に一本のビームで回路を描いていくため、大量生産には向かないという致命的な欠点がある。X線は、等倍露光ながらステッパコンセプトを用いることができ、大量生産に使いそうだが、技術的に未知な部分が多く、膨大な開発費がかかる。

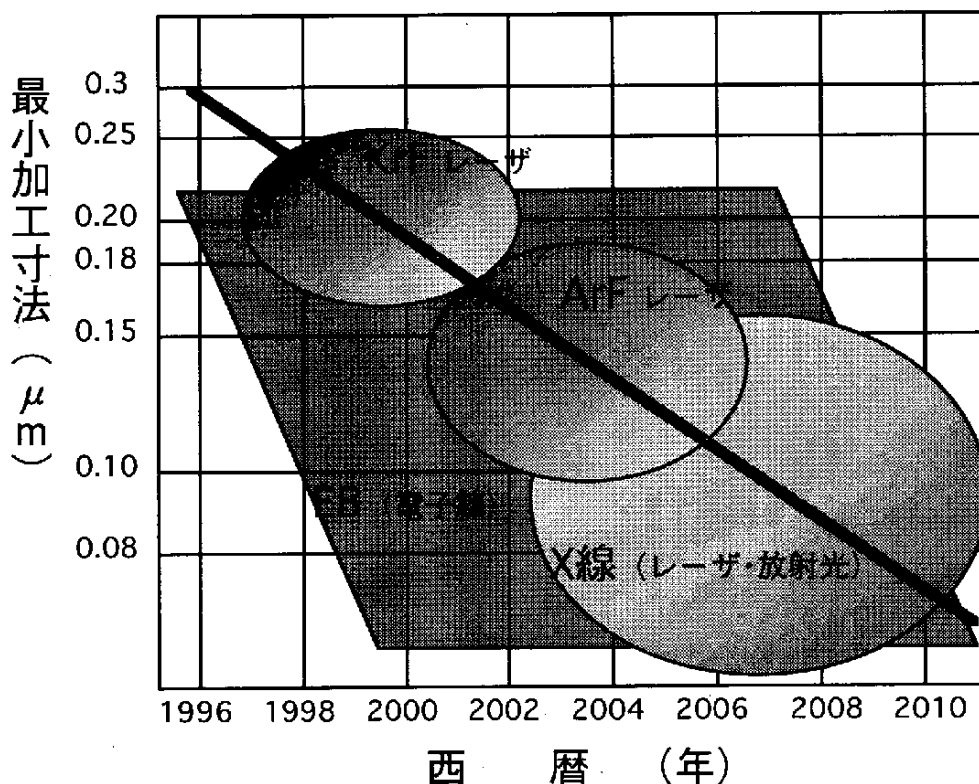


図1 リソグラフィ技術マップ

・配線

配線は、電源供給、情報伝達を受け持ち、その技術は高集積化、高速化を直接左右する。2010年に向かったのロードマップを表3に示すが、厚さ $0.1\mu\text{m}$ 、長さ $1\mu\text{m}$ の配線抵抗において、配線幅は現在の1/5でも、抵抗値は約9倍であり、配線幅の減少がいかに配線抵抗に影響を与えるかがよくわかる。抵抗値を減少させるために低抵抗率かつストレス/エレクトロマイグレーションに強い銅を含む金属材料や低容量を実現する低誘電率層間絶縁膜材料の開発が急務になっている。

表3 配線技術ロードマップ

暦年	ルール (μm)	抵抗値 ($\text{Ohms}/\mu\text{m}$)	容量 ($\text{fF}/\mu\text{m}$)	層数	総配線長 ($\text{m}/\text{チップ}$)	信頼性 (FITs/m) $\times 10^{-3}$
1995	0.35	0.15	0.17	4.5	380	16
1998	0.25	0.19	0.19	5	840	4.7
2001	0.18	0.29	0.21	5.5	2100	1.1
2004	0.13	0.82	0.24	6	4100	0.5
2007	0.1	1.34	0.27	6.5	6300	0.4
2010	0.07	1.34	0.27	7.5	10000	0.2

・セル構造

2010年にゲート長が $0.07\mu\text{m}$ になったとき、その長さ方向に並ぶシリコン原子の数は約200個であるが、Siトランジスタの電気特性を決めるSi中の不純物量は、通常(10^{16}cm^{-3} であるから体積で 10^{-6})線密度では1/100なので、長さ方向に不純物が1個か2個ランダムに並ぶことになる。この状態でも電子の運動をゲートで制御することは可能であるが、問題は同じ動作をさせるために、そのゲート電圧の値をチップ上のすべてのトランジスタで異なる値をとることである。いいかえると、LSI動作のためにはチップ内の不純物分布を超均一に(もしかしたら原子レベルで)しなければならない。これについては、現在でも解はない。

DRAMの記憶部を形成するコンデンサ部はトポロジカルには誘電体を2枚の金属電極ではさんだ形となっている。記憶を読みだしたり、リフレッシュレートを保つために、その容量は小さくすることができない。そのため、電極面積を極力大きくする工夫がなされているが、限界があり、新しい高誘電体材料の開発が待たれている。

(2) 低消費電力化

表4に電源電圧とハイエンドプロセッサの消費電力のロードマップを示す。電源電圧の低下にもかかわらず、最大電力は増加し、かなり厳しい状況である。ここで、消費電力は電源電圧を下げると減少するが、同時に、回路遅延も増大するというトレードオフを解決しないといけない。このため、電源電圧低減とともに、回路設計技術が重要である。トランジスタ数を減らすアーキテクチャ、スイッチング確率を減らす回路設計のためのCAD開発が必要である。

表4 電源電圧ロードマップ

暦年	ルール (μm)	電源電圧 (V)	最大電力 (W)
1995	0.35	3.3	80
1998	0.25	2.5	100
2001	0.18	1.8	120
2004	0.13	1.5	140
2007	0.1	1.2	160
2010	0.07	0.9	180

(3) 実装

表5にチップとパッケージを結ぶ配線のためのパッド数と、パッケージに備えられるピン数のロードマップを示す。MCM (Multi-Chip Module)を含むボード設計技術やアーキテクチャを含むシステム設計技術開発が必要である。

表5 実装技術ロードマップ

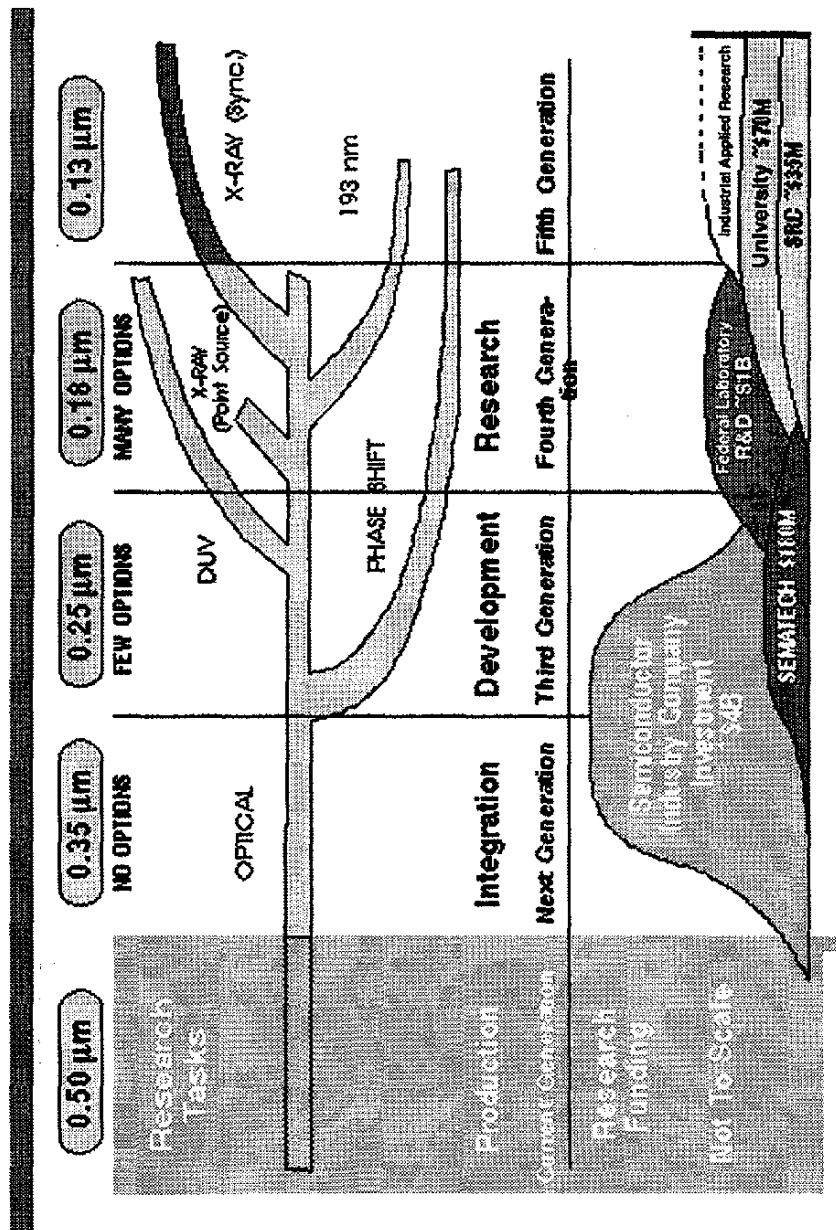
暦年	ルール (μm)	チップパッド数	パッケージピン数
1995	0.35	900	512
1998	0.25	1350	512
2001	0.18	2000	512
2004	0.13	2600	512
2007	0.1	3600	800
2010	0.07	4800	1024

(4) コスト

低チップ価格は、ペタフロップスマシン開発思想の根幹をなすC-COTS (Commodity Commercial-Off-The-Shelf) を支えるものであり、半導体産業が成長する原動力である。チップコストには開発コスト、製造ライン設置コスト、チップ製造コストなどがある。

・開発コスト

図2は0.13 μm 世代迄のリソグラフィの開発項目と合わせて、開発費の支出元を示したものである。将来技術開発に対し、国立研究所、大学の支出額が大きいことがわかる。

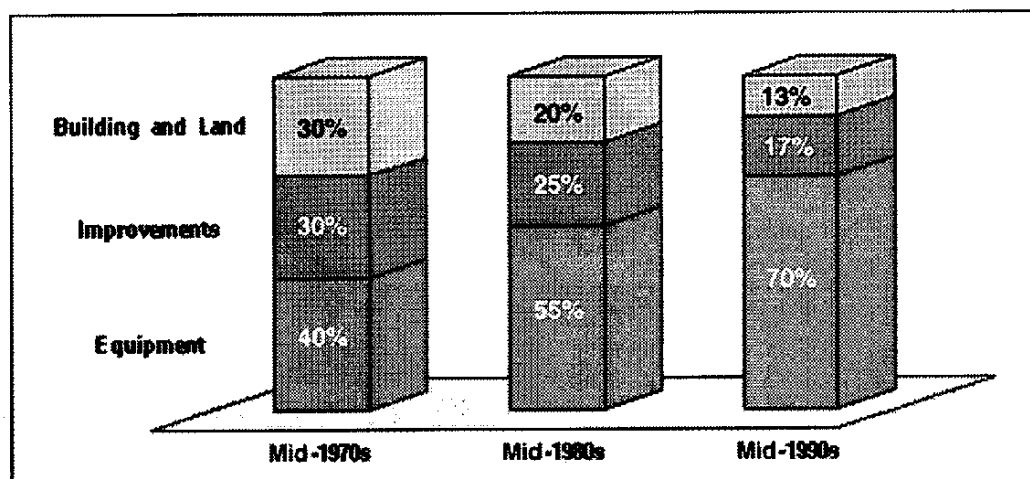


<http://www.semtech.org/public/roadmap/doc/intro.html>

図2 コストロードマップ

・ライン製造コスト

ライン製造コストは3年で約1.3倍となるといわれており、これからも増大していくと予想される。ここで、図3に1970年代から1990年代にかけてのライン製造コストの内訳の変化を示す。装置コストの上昇が際立っている。



<http://www.semtech.org/public/roadmap/doc/facint.html#0.2.SWFMI2.WYCEHD.PEM0HD.Z9>

図3 ライン製造コスト内訳

・CMOSデバイス製造での各部のコストロードマップ

表6にメモリ／MPUチップ、パッケージ、配線のロードマップを示す。各工程で単位あたりのコストは低減されなくてはならない。

表6 製造コストロードマップ

暦年	ルール (μm)	メモリ ($\mu\text{Cents/Bit}$)	MPU ($\mu\text{Cents/Tr}^*$)	パッケージ (Cents/pin)	配線 (photoを除く) ($\$/\text{cm}^2/\text{層}$)
1995	0.35	17	1000	1.4	0.29
1998	0.25	7	500	1.3	0.23
2001	0.18	3	200	1.1	0.23
2004	0.13	1	100	1	0.18
2007	0.1	0.5	50	0.9	0.18
2010	0.07	0.2	20	0.8	0.14

*) Tr は 1 トランジスタを表す。

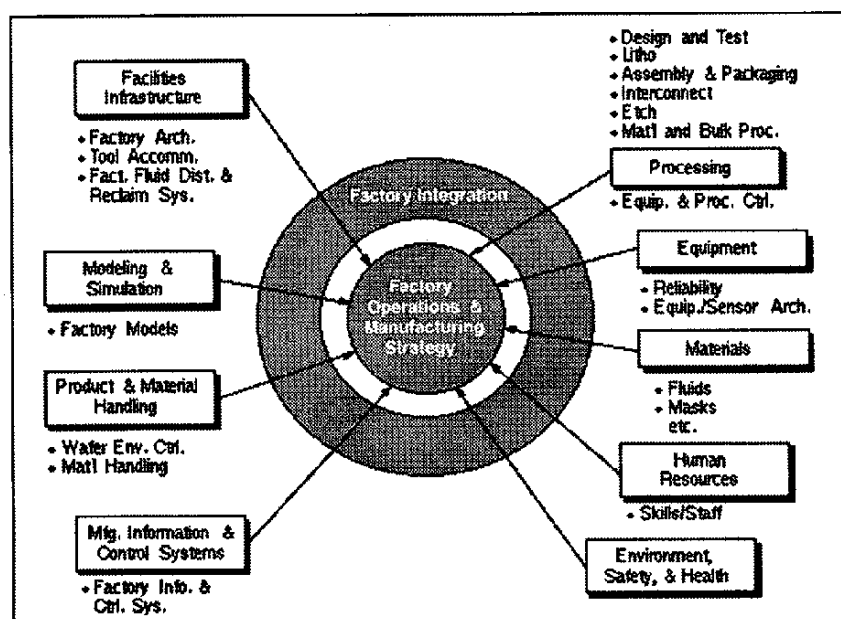
・その他

コストに密接に関係のある工程数の主要素のマスク数、TAT (Turn Around Time) を示すサイクルタイム、ライン製造効率を上げるウエハ直径、歩留りを左右するパーティクル径の要求トレンドを表7に示す。2010年には現在のパソコン用大型CRTと同じくらい大きな面積のウエハを取り扱わなくてはならない。

表7 その他の技術要素ロードマップ

暦年	ルール (μm)	最大マスク数	サイクルタイム (日)	基板直径 (mm)	最大パーティクル径 (μm)
1995	0.35	18	9	200	0.12
1998	0.25	20	10	200	0.08
2001	0.18	20	10	300	0.06
2004	0.13	22	11	300	0.04
2007	0.1	22	11	400	0.04
2010	0.07	24	12	400	0.03

一方、半導体製造技術の開発を早期、高効率、かつ不満足なく行うためには製造技術全体を一体のシステムとして取り扱うことが必然となっており、その構成、運営能力の向上が急がれている。図4に米国におけるCMOSデバイス製造のシステム指向を表す模式図を示す。



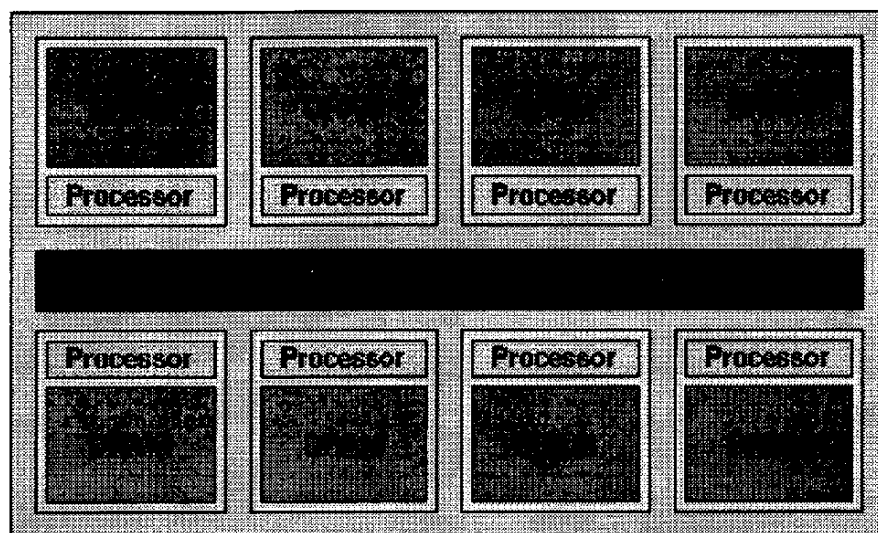
<http://www.sematech.org/public/roadmap/doc/facint.html#0.2.SWFM12.WYCEHD.PEM0HD.5A>

図4 米国の製造システム概念図

3.4.2 システムよる対応

以上見てきたように、CMOSデバイスの単なる製造技術の向上ではそれほど大きな高性能化は望めない。大規模なトランジスタの集積が可能になったとき、チップ上に最適なシステムを組み上げることによって、高効率に高性能化を達成する技術が必要である。ペタフロップスイニシアチブの一つの結論としてPIM(Processor In Memory)が必須技術として取り上げられているのは当然である。現在、九州大学を中心としたPPRAM(Parallel Processing RAM)コンソーシアム (http://kasuga.csce.kyushu-u.ac.jp/ppram/index_j.html) やシステムLSIをよりオープンにかつ競争的に開発できるようにインタフェースを統一しようとするVSI(Virtual Socket Interface)アライアンス (<http://www.vsi.org/>) などの活動があり、この方向での研究開発として注目される。

ペタフロップスイニシアチブではワークショップを通じ、PIMの予想性能を導出している(Proceedings of the Petaflops Systems Workshops, edited by M.J. MacDonald, 1996)。そこでは、SIAのロードマップを参照し、DEC ALPHA を基準としながら、SMP CPUチップ性能を理論推計している。結果は、2010年に $0.07\mu\text{m}$ ルール、チップ面積 609mm^2 として、クロック1.9GHz (SIA予測より大きい) を達成し、16CPU、2.7GビットDRAMを搭載したチップ性能が71Gflopsである。このチップであれば、ペタフロップス達成のためには単純計算としては14,000個余あれば良いことになる。これは、並列度は別として、チップ数だけを考えれば、ASCI Redの1.5倍にすぎない(ただし、このチップが汎用となる保証はなく、C-COTSを実現するかどうかも不明である)。このようなチップを現実のものとするためには、強力な設計技術と少数だが非常に高い能力を持った人材の育成が必要であると考えられる。



http://kasuga.csce.kyushu-u.ac.jp/ppram/japanese/ppram_framework.html

図5 PPRAMのチップレイアウト例

3.4.3 CMOS以外のデバイス

以上、ペタフロップスマシンを実現するデバイスとして本命視されるCMOSデバイスのトレンドについて述べてきたが、これでいくと、2010年に数GflopsのMPUが出現すれば良いほうである。このため、Petaflopsを実現するためには数十万から百万のプロセッサの並列動作が必要となる。これを実現するアーキテクチャ、ソフトへの負担は相当のもので、ペタフロップスイニシアティブのシステム担当からでさえ、並列度を下げるための代替が強く望まれている。将来的なデバイスとして超伝導素子、単電子トランジスタなどが挙げられ、システムとして量子コンピュータが考えられている。後二者は2010年以降の話であり、到底間に合うとは思えず、超伝導素子だけが、現在製品レベルのIC（A/D converter 3.5 μ m ルール、7GHz、1778 JJ）としての形を表している。

超伝導素子は動作速度、消費電力共にCMOSと比較して非常に有利であるといわれている。（超伝導素子の一種であるRSFQ（Rapid Single-Flux Qantum logic）の消費電力を表8に示す。）このRSFQを用いたLSIのロードマップ（Rikharevによる）を表9に示す。これによると2007年には150GHzのLSI動作が可能で、MPU数を二桁減らせることになる。

しかしながら、その実現性を疑っている人も多く、これに多くを期待するのは無理であろう。

表8 CMOSと超伝導素子の消費電力

	CMOS	RSFQ
1bitあたりの消費エネルギー(J)	10^{-13}	10^{-18}
10^{20} bitあたりの消費電力 (W)	10^7	10^2

表9 Rikharevによる超伝導素子開発ロードマップ

西暦	1998	2001	2004	2007
ルール(μ m)	3.5	1.5	0.8	0.5
ゲート数 (Kgate/cm ²)	10	30	100	1000
電流(kA/cm)	1	6.5	20	50
C(aF/ μ m ²)	45	60	67	75
clock(GHz)	150	300	500	700
LSI動作(GHz)	30	60	100	150
消費電力(μ W/gate)	0.03	0.06	0.1	0.15
cost/gate(millicent)		50	10	1

第4章 あとがき

本ワーキンググループ(WG)は、現在、商用マシンの開発が終り、市場に登場しようとしているテラフロップスマシンの後を受けて、ペタフロップスマシンなど将来の超高速計算機に関する技術調査を行なうことを目的として発足した。WGとしては、約半年の間に5回の討論を行ったに過ぎず、ペタフロップスという、このような大きなテーマに対しては、まだ準備段階が終了した時点であろう。米国ではペタフロップスマシンという言葉自体は、日本と比較した場合に、より広く認知されているように見受けられるが、その必要性や研究開発の方向性などに関して必ずしも目標が見えているわけではなく、まだ手探りの状態で議論を進めているようにも見える。このような観点からみて、わが国での本テーマに対する取り組みが、必ずしも遅れてしまったとは思えない。ただし、米国では、将来の方向性に関して、正しく認識し議論を行おうとする姿勢は少なくとも一歩先んじていると思われる。

本報告書において、各委員は、独自の判断でペタフロップスマシンに対する主張を展開しているわけであるが、いくつかの点でかなり共通の認識があるという印象を受けた。その1つはペタフロップスのコンピュータは超並列計算機にならざるを得ないという点である。ペタフロップスマシンの構成をハードウェアとして考えた場合に、現在のCMOS技術に基づいたシリコンを前提とすると2010年前後を想定した時に、プロセッサの単体性能としては、数GFlopsから数10GFlopsであろうと考えられることは、多くの専門家の一致をみるところである。そのように考えると、ペタフロップスを実現するためには、数10万規模のプロセッサの並列処理が必要ということになる。本報告書の中の各委員の意見に見られるように、実際はもっと様々な形態が想定されるわけであるが、本質的にペタフロップスの処理を行うためには、非常に多数のプロセッサによる並列あるいは分散的な処理を行わざるを得ないという点も、一致した見方である。

2つ目は、ペタフロップスを実現するための超並列アーキテクチャを持つマシンでは、最大性能と実効性能の乖離現象がさらに顕著に現われるという点である。このため、並列計算機の実効性能を最大限に引き出す並列ソフトウェアなどの技術がますます重要になるものと思われる。これには、数多くの研究課題が残されているわけであり、それを克服するためには、超並列システムの実機を研究開発し、その上でソフトウェア開発を積み重ねていかなければならないという点が指摘できる。

3つ目は、日米の技術格差が特に基幹ソフトウェアや産業基盤ソフトウェアにおいて顕著であるという認識であり、この点が将来のペタフロップスマシンの開発や普及において、わが国における弱点となる可能性があるというものである。

4つ目は、ペタフロップスのような大きな計算能力を持ったコンピュータを必要とし、また、それを使いこなすためには、それに見合ったアプリケーションとその利用方法などについて、しっかりした道筋が必要ではないかという点である。

最後の点に関しては、ペタフロップスのような巨大な計算能力を持ったコンピュータが出現すれば、それをベースに必ず新たな用途が生じるという意見もあったが、少なくともどのようなアプリケーションがペタフロップスマシンを必要とし、またどの程度までそのアプリケーションの問題が解決されるかを明らかにしておくことの重要性については、認識の一致があると考えられる。

平成8年度の本WGにおける調査・検討は、期間的にみて十分とは言えず、さらなる調査が必要であろう。平成8年度の議論から得られた今後の調査の方向性についての私見は、つぎのようにまとめられる。

- (1) ペタフロップスマシンのアーキテクチャよりも、ソフトウェア（コンパイラ技術やプログラミングシステム）およびアプリケーションに関する問題意識がより重要である。
- (2) 効率のよい並列処理を行うために、何が解決可能な問題であることを明らかにするとともに、その問題の要素が並列処理との関係でどのような実行の振る舞いをするか、あるいはすべきであるかについて明らかにすることが重要である。

さらにつけ加えれば、従来から、日本においては各種のプロジェクトが行われてきたわけであるが、ペタフロップスといった、はるか先をみつめた技術開発においては、よりグローバルな人類の利益を見据えた技術開発のための「目標」が必要であると考えられる。特に、情報処理技術に関しては、いわゆるグローバルイゼイション化がますます進行するという状況のなかで、我が国の技術的なポテンシャルをどのように高めていくかという点を考える必要がさらに高まっている。米国は、HPCCプログラムにおいて、Grand ChallengeやNational Challengeという研究開発プロジェクトを打ち立て、情報処理技術における最先端技術とインフラストラクチャの拡充という両面において、国をあげて取り組んできた印象がある。これ

から21世紀に向かい、わが国の情報処理技術に関するポテンシャルの高めるために、このような全世界的な視野の中で、戦略的に追求することがぜひとも必要であろう。また、ペタフロップスマシンの開発は、未踏の領域への挑戦でもある。このような観点から見ると、さらなる調査課題として、ペタフロップスマシンの可能性を探るとともに、これらの先進的技術の基盤を確立するために何をなすべきかを明らかにしていくことが必要である。また、技術の可能性と有用性を確かめるためには、具体的なアプリケーションを想定して、その問題がどのように解決されるか、あるいは解決されるべきかを議論することがさらに重要であると考えられる。

(山口喜教主査)

付属資料1 ペタフロップスマシンWG配付資料概略

PFM-WG-0001

Enabling Technologies for Petaflops Computing

(February/22-24/1994, App/Dev/Arch/Soft)

<http://nhse.npac.syr.edu/roadmap/petaflops/peta.html>

- ・ 1994年2月にPasadenaで開催されたワークショップの結果をまとめたもの。
- ・ MIT Pressから1995年7月に出版されている。日本で近日翻訳本が発売される予定(小林達・栗本武共訳、筑波出版会)。
- ・ ワークショップは、以下の4つの分野に焦点をあわせて実施された。

<編成されたWG>

・ Applications and Algorithms

この規模の処理能力の開発の必要性和能力

必要資源について知見を得た。

・ Device Technology

半導体、光デバイス、超伝導素子

それぞれについて強み、弱みを調べた。

・ Parallel Architectures

プロセッサ、通信、メモリサブユニットを含む3種の構造を調査。

実現しうるシステムを明確にするいろいろな技術力段階における最も望ましい

構成と機能要素の集合を調べた。

・ System Software and Tools

将来システムの実用的なアプリに対して、現在のソフト環境による主要な障害を明

確にする。代替の環境と有用性を本質的に増加させる機能とのかかわりあいをも調

べた。

<結論>

エグゼクティブ・サマリから

- (1) 実用的なペタフロップスマシンは現在の技術トレンドをベースにすれば約20年でできるであろう。
- (2) 近い将来ペタフロップスを要求するアプリは科学、技術、経済、社会、情報基盤、管理の分野の広い領域に亘っている。
- (3) なによりもコストが根本的な実行可能性と時間軸を支配している。

- (4)信頼性は管理できるだろうが、現在のMPPよりはるかに多くの部品を有するシステムを妨げる唯一の理由はコストである。
- (5)システム・アーキに基本的なパラダイムシフトは必要ない。詳細は現在と異なるだろうが、NUMA、MIMD(可能ならSIMD)の進化型が満足させる。

さらに、本文でペタフロップス実現の可能性ありとして提示されたアーキテクチャは、以下の3方式。

- a.グローバル・シェアド・メモリ
- b.マイクロプロセッサ・ネットワーク
- c.プロセッサ・イン・メモリ (PIM)

- (6)ペタフロップスマシンは直径が大きくなる。伝達遅延はシステムクロックサイクルで計られる。レイテンシ管理技術と百万のオーダのコンカレンシがこの規模のシステムのキーとなる。
- (7)メモリによっても支配される。しかし、科学技術の分野ではメモリ量は能力に対し線形以下の大きさであろう。ペタフロップスに対し30テラバイトオーダのメインメモリが要求される。
- (8)ペタフロップスマシンはより良いコストパフォーマンスを達成する技術の集合によって構成される。半導体技術はいくつかのロジック回路を伴ったメモリを支配する。その進歩は半導体産業の進展に拘束される。光デバイスは広いバンド幅とモジュール間通信及び巨大ストレージを提供するが、ロジックデバイスができない。超伝導JJ技術は、非常に高いロジック性能と比類のない低消費電力をもたらす。
- (9)システムがエンドユーザアプリにとって実用的となるには、プログラミングとリソース管理のためのソフトウェア方法論に大きな進展が必要である。また、ソフトウェアテクノロジー・ワーキンググループ（第七章、129ページ）によれば、100ギガフロップスマシン用のソフトウェアは、ペタフロップスに向かないとしている。

本文から

- (1)いつ：20-25年先-現在の技術トレンドが続き、技術が成熟した場合
- (2)誰によって：半導体デバイス、光デバイス、JJ開発が鍵を握る。また、ソフトウェア技術方法論、基盤開発が必要。
- (3)用途は：ターゲットが古めかしいプログラミング技法を使ったようなグランド・チャレンジスケールのアプリに限定されるなら、このようなシステムは使えないし、手に入らないし、実現しないだろう。成長する科学技術コミュニティと将来予想されるより広い

情報管理の一般巨大市場コミュニティ（先進的な高速プログラミング方法論とNIIとしての集積を伴っている）の両方をターゲットとすれば、経済的に存立するし、何百万もの情報に対して行動する消費者によって直接・間接的に使われるだろう。

- (4)研究開発型製品における長期先行性：短期間のインパクトを持たなくても競争に勝つためやるべき。
- (5)本流となるハード・ソフトウェアの利用：システムは商用ソフトとコンパチでなくてはならない。また、プログラミングモデルは商業的成功に自信をもって並列プログラミング世界に入らせるISVを可能にしなければならない。
- (6)アプリを越えた価値：創造力、技術、さらに多くの舞台で予知できないが巨大な価値と重要性を持つ米国の競争力の本質的な能力をもたらす。
- (7)知的シナジー：このワークショップではペタフロップスコンピューティングの大きな理解もあるが、それを超えて重要で即効の結果はコンピュータ科学に対する国家の主要な貢献者のいくつかの間でシナジーとアイデアの交換・ふくらましができたことである。

<これからやるべきこと>

- (1)超伝導技術
- (2)アプリケーション資源要求の規模拡大
- (3)将来的なアプリケーションのシナリオ
- (4)アーキテクチャーの詳細
- (5)SIA計画の拡張
- (6)プログラミング方法論
- (7)コンセプトとアプローチの代替案
- (8)アーキテクチャ的アプローチの代替案
- (9)安価なテラフロップスマシン
- (10)進捗のレビュー

PFM-WG-0002

The Proceedings of The Petaflops Frontier Workshop

(February/6/1995, Arch/App)

<http://cesdis.gsfc.nasa.gov/~creschke/peta/report/report.html>

- ・ 1995年2月にMcLeanでで開催されたワークショップの結果をまとめたもの。
- ・ このワークショップは、アーキテクチャ／テクノロジーとアプリケーション／アルゴリズム

が対象。

- ・ 第一節（5ページ～）Introductionには、以下が記述されている。
 - (1)いくつかのキーとなる出来事
 - (2)今回のワークショップに先立つ諸活動
 - (3)ワークショップの目的とアプローチ
 - (4)報告のための組織
- ・ 第二節（10ページ～）Issues for Petaflops Computersには、ワークショップでの成果や結論を含め、ペタフロップス・コンピューティングに対する論点のサマリが記述されている。
- ・ 第三節（17ページ～）Workshop Organizationには、このワークショップの議事次第や発表題目18件（19ページ）、主催者、発表者、出席者の情報が記述されている。
- ・ 第四節（24ページ～）Overview of Presentationsには、このワークショップの発表内容の概要について記述されている。
- ・ 第五節（28ページ～）Architecture and Technology Issues and Challengesには、アーキテクチャとテクノロジーに関する以下の6件の発表内容の要約が記述されている。
 - (1)Heterogeneous Computing: One Approach to Sustained Petaflops Performance
 - (2)Processors-In-Memory (PIM) Chip Architectures for Petaflops Computing
 - (3)A Petaops is Currently Feasible by Computing in RAM
 - (4)Design of a Massively Parallel Computer Using Bit Serial Processing Elements
 - (5)Non von Neumann Instruction Set Architecture as an Enabling Technology in Grand Challenge Systems
 - (6)Taming Massive Parallelism: The Prospects of Opto-Electronic CRCW-Shared Memory
- ・ 第六節（69ページ～）Applications and Algorithms: Issues and Challengesには、アプリケーションとアルゴリズムに関する以下の発表内容の要約が記述されている。

Enabling Data-intensive Applications through Petaflops Computing
- ・ 第七節（71ページ～）Discussion and Conclusionsには、ワークショップでの結論や将来に対することが記述されている。

PFM-WG-0003

P.E.T.A.: PetaFLOPS Enabling Technologies and Applications

(Arch/App/Soft)

<http://www.aero.hq.nasa.gov/hpcc/petaflops/peta.html>

- ・ P.E.T.A.は、ペタフロップスコンピューティングに関する、WWW上のリソース・センタ

一である。ここを見ると、ペタフロップス・コンピューティングに関するワークショップや会議の情報、出版物、議事録などの情報を得ることができる。

・今回配布の資料には、以下に関する記述が載っている。

(1)PetaFLOPS Architecture and Systems Structure

ペタフロップスに関する論点が、質問と回答という形で簡単に書かれている。

(2)PetaFLOPS Applications and Software

ペタフロップスコンピューティングの適用分野、ソフトウェア技術やアーキテクチャの論点となる項目、アプリケーション・グループのゴールなどが簡単に書かれている。

(3)1995 PetaFLOPS Computing Summer School/Workshop

ペタフロップスのアプリケーションとアルゴリズムに関するワークショップの紹介。

PFM-WG-0004

Applications and Algorithm Challenges for PetaFLOPS Computing

(1995 Petaflops Computing Summer School/Workshop, August/14-23/1995)

<http://www.mcs.anl.gov/summer.study/index.html>

(1)ゴールの設定

障壁とロードマップの明確化

(2)アプリケーション・ケーススタディ

科学技術の分類、各研究分野での要求性能を別表として掲載

(3)ケーススタディの詳細記述

生物学、気象・海洋のモデリング、化学・物理学(天体物理)、薬学、工学
自動推論、経済学、核兵器

(4)アルゴリズム: 科学技術計算に用いられるアルゴリズムの列挙

(5)ソフトウェア: 現状の効率は15%、向上の指針は少ない。

プログラミング言語、モジュール化、レイテンシ、並列I/O

(6)アーキテクチャ: パソコンのネットワークによる並列化

- ・対称マルチプロセッサ
- ・プロセッシング・イン・メモリ
- ・細粒度MIMDシステム

(7)性能のモデリング: ボトルネックについての情報を得る

- ・現状のモデルは既に使えなくなっている(企業の研究者の言)
- ・ isoefficiency function

(8)主な成果

- ・商品化される部品を使ったとして20-25年後に実現できるだろう。
- ・プロセッサアーキテクチャの提案
- ・レイテンシ管理、メモリ必要量、高性能プロセッサの要求等があるが、あまり具体性があるようには思えない。

これからの実施項目

- ・アプリケーション実行上での制限項目のリステイング
- ・ Support research
- ・ Sponcer development
- ・ Point design studies
- ・ Fund advanced system software
- ・ Sponcer additional workshop
- ・ petaFLOPS oversight committeeの設立

PFM-WG-0005

PetaSoft '96

(Petaflops Summer Study on System Software, June/17-21/1996)

<http://www.aero.hq.nasa.gov/hpcc/petaflops/petasoft96/ps.html>

- ・ スポンサー: ARPA, DOE, NASA, NSA, NSF
- ・ Petaflops Architecture Workshop のFindingsとRecommendationsをもとに議論
- ・ ベタフロップスシステムのソフトウェアコンポーネントのための韻律則を評価する。
- ・ 11月にProceedingsが出る予定。
- ・ <http://www.cacr.caltech.edu/calendar/petasoft/>にシンポジウムの様子を写した写真のみ掲載

PFM-WG-0006

Petaflops Architecture WorkShop- DRAFT Findings, DRAFT Recommendations

(April/22-25/1996)

- ・ Findings

<http://www.aero.hq.nasa.gov/hpcc/petaflops/petasoft96/paws.outcome/findings.html>

アーキテクチャ、素子テクノロジー、アプリケーション/アルゴリズム、システムソフ

トウェアについて詳述されている。

SPD(Special Purpouse Device)-最初のペタフロップス・コンピュータは特殊なものであるが、5年以内に実現するかもしれない。価格は控えめに見ても1千万ドルするだろう。

・ Recommendations

<http://www.aero.hq.nasa.gov/hpcc/petaflops/petasoft96/paws.outcome/recommend.html>

アーキテクチャ、素子テクノロジー、アプリケーション／アルゴリズム、システムソフトウェア、一般事項について詳述されている。

一般事項では、米国内のスーパーコンピュータ産業がペタフロップスの能力をつけるよう政府のてこ入れを主張している。また、準ペタフロップスを実現する中間的プログラムの必要性が説かれている。

PFM-WG-0007

Intel Homepage

<http://www.ssd.intel.com/tflop.html>

Sandia National Lab. にテラフロップスをめざした20GFlopsの装置を納入(96/4/9)、年末までにテラフロップスを達成予定。

システムはASCIプログラムのアプリをサポートするために、インテルParagon Operating system をLight-Weight Kernel(LWK)によって拡張したものである。

ASCI(Accelerated Strategic Computing Initiative) Teraflops Machineの説明

- ・ 1.8 Tflops peak performance
- ・ 4500 dual node
- ・ 9000 Pentium Pro
- ・ 600 GBytes memory
- ・ 2 TBytes RAID disk strage
- ・ 85 cabinets
- ・ 1600 square feet
- ・ 400 MBytes/s transfer speed between two nodes
- ・ 50 MBytes/s cross-section bandwidth between all nodes
- ・ 800 kW of power
- ・ 4,600万ドル (\$46million)

Teraflops computer is first step in pursuit of nuclear freedom.

<http://www.ssd.intel.com/press/record1.html>

- ・ Pragonの紹介 (装置の写真入り)

PFM-WG-0008

Concurrent Supercomputing Consortium

1995 Annual Report

<http://www.cacr.caltech.edu/publications/annreps/annrep95/cover.html>

インテルTouchstone Deltaがインテル Trex(Paragon XP/S Model L38)に置き換えられたこと、NSF 資金により74の新ノードが設置できたこと等を含めたトピックスの紹介。以下に目次を記す。

(1)Introduction

- ・ Enhancement of CSCC Facilities
- ・ Creation of Caltech's Center for Advanced Computing Research
- ・ Future Challenges

Scalable I/O Initiative Under Way

Stimulating Progress in High-Performance Computing

Outreach Activities

CSCC Mission and Strategy

(2) Computational Research

(Research Articles by Topic. Author Index also found here.)

(3) CSCC Administration

(4) Computing Facilities

(5) Recent Publications

(6) World Wide Web Documents

(7) Acknowledgments

FM-WG-0009

The 2nd Workshop On The Petaflops Frontier

- ・ Frontiers '96に先だって開催されたワークショップのAgenda

- ・ワークショップ、シンポジウム参加の事前登録者名簿

PFM-WG-0010

The Petaflops Systems Workshops

- ・今年（1996年）に開催されたPetaflops Architecture WorkShop(PAWS'96)とPetaflops System software Summer study (PetaSoft'96)の二つのプロシーディングを合わせたもの。
- ・The 2nd Workshop On The Petaflops Frontier 登録時に受付で配付された。
- ・Abstract、Contents、Findings、Recommendationsを抜粋。前回配付した資料PFM-WG-0006 PAWS DRAFT Findings Recommendations が改変されており、さらにPetaSoft'96の結果が追加されている。

PFM-WG-0011

Proceedings of Frontiers'96

- ・The Sixth symposium on the Frontiers of Massively Parallel Computation のプロシーディング。
- ・1996年10月29日～31日の発表分を掲載してあるもので、このうちContents、Conference committee、Program Committee、Referees、Petaflops Computing/Point Design Studiesを抜粋した。

PFM-WG-0012

Frontiers'96 出張報告

- (1)米国出張レポートOHP集 AITEC 高張
 - (2)海外出張報告 AITEC 河西
 - (3)Frontiers'96参加報告書 電総研 関口
- 詳細は本書付属資料2、3参照。

PFM-WG-0013

結合網とルータに関する最近の話題（天野委員発表OHP）

PFM-WG-0014

Supercomputing'96 報告 (山口主査発表OHP)

PFM-WG-0015

MPC++ (石川委員発表OHP)

PFM-WG-0016

PetaFLOPS Computing (ソフトウェアの観点からの話題提供)
(笠原委員発表OHP)

PFM-WG-0017

Peta xOPSマシンに向かったのメモリ・アーキテクチャのお話 (中島 (浩) 委員発表OHP)

PFM-WG-0018

PFM-WGアプリケーションの観点からの話題提供 (福井委員発表OHP)

PFM-WG-0019

並列ベクトル計算対スカラー並列計算 (横川委員発表OHP)

PFM-WG-0021

21世紀の半導体技術 (東芝 海野所長講演OHP)

PFM-WG-0022

ペタフロップスマシンの実現性～ソフトウェア構築の観点から～ (妹尾委員発表OHP)

PFM-WG-0023

大規模並列における応用とその支援ツール (中島 (克) 委員発表OHP)

付属資料2 Frontiers '96 出張報告

Frontiers'96はThe Sixth Symposium on the Frontiers of Massively Parallel Computingの略称で、IEEE Computer Societyが後援、NASAが協賛して開催された。

1. 開催場所

米国メリーランド州アナポリス、ローズアナポリスホテル

2. 日程

96/10/27(日) - 10/28(月) ワークショップ

96/10/29(火) - 10/31(木) シンポジウム (テクニカルプログラム)

3. 出席者

出席者人数は、前年とあまり変わらず、事前登録者は111名、このうち日本から6名が参加した。AITEC以外の日本からの出席者は、以下のとおり。(敬称略)

- ・雨宮真人 (九州大学)
- ・中村 壽 (高度情報科学技術研究機構)
- ・中島研吾 (三菱総研)
- ・牧野淳一郎 (東京大学)

4. 講演内容

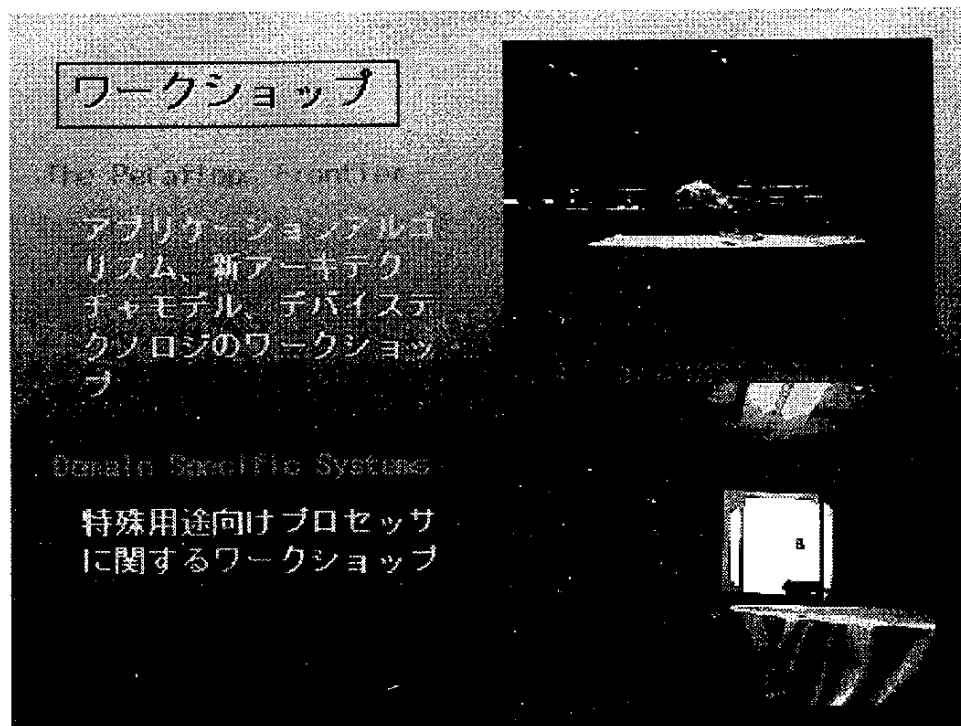
4.1 ワークショップ

The Petaflops Frontier、Domain Specific Systemsの二つのワークショップが、同時並行に開催された。前者は、アプリケーションアルゴリズム、新アーキテクチャモデル、デバイステクノロジーに関するワークショップである。メインのワークショップのこともあり、出席者の多くが、このワークショップに参加した。後者は、特殊用途向けプロセッサに関するワークショップで、新しくできたワークショップである。参加者が10名不足と低調である。このため、以下、"The Petaflops Frontier" について報告する。

"The Petaflops Frontier" は、講演、一般発表、パネル討論で構成される。今回の主要テーマは、NSFがファンドし、DARPAとNASAがそれを補助しているPoint Design Studiesと呼ばれる100テラフロップスマシンの研究成果発表である。この研究は、10年で100テラフロップスマシンの完成を目指すもので、研究成果は、ワークショップとシンポジウムのそれぞれで発表された。

また、超伝導デバイス関連の発表が数件あり、そこには意外な熱気があった。

100テラフロップスマシンの研究以外の発表は、ペタフロップス達成までの道のりが長い
ためか、焦点の定まった発表はないようである。



4.1.1 講演 1 : TERA Looks at Peta (テラからペタへ) : Issues in Compilers and Operating Systems

講演では、OSボトルネック、マルチサーバボトルネック、パラレルプログラミング、混在モードパラレルプログラミング、スケジューリング法に対するプログラマのコントロールに関するコンパイラとオペレーティング・システムの問題点が論じられた。

4.1.2 講演 2 : An Introduction to Petaflops Point Design Studies

NSFによる100テラフロップスマシンの研究紹介である。

現時点でのアーキテクチャの特徴は、以下のとおりである。

- ・ プロセッサ100万台並列動作
- ・ 不均一メモリアクセス・モデルまたは分散メモリアーキテクチャ
- ・ 複合マルチレベルメモリ階層
- ・ 複雑なソフトウェア
- ・ Grand Challenge User Feel (グランドチャレンジのユーザニーズ)
- ・ 入出力問題

1997年の焦点となる分野は以下のとおりである。

- ・プログラミング環境とツール
- ・グラフィクスと可視化
- ・HPC Application

4.1.3 Architecture and Technology に関するPoint Design Studies

NSFファンドによる8件の研究発表である。

(1) Persuing a Petaflops: Point Designs for 100 TF Computers Using PIM Technologies、

P.M.Kogge et.al. ノートルダム大

PIMテクノロジーはピン数に制限されず、CPUコアが簡単にできる。CPU-DRAMバンド幅を大きくでき、トランジスタ効率が良い、といった利点が主張された。

(2) Hybrid Technology: Multi-Threaded Architecture、G.Gao、SUNY、Caltech/JPL

1000台の100GHz動作のプロセッサに小規模の超伝導メモリキャッシュをつなぎ、この部分で相互接続する。そのバスにはバッファを通して77Kに冷却した1GHz動作のSRAMを接続し、光パケットを通して256MBのDRAMに接続されるといういろいろなテクノロジーの混合。

(3) Design Studies on Petaflops: Special-Purpose Hardware for Astrophysical Particle

Simulations、F.Summers et. al.、コロンビア大、イリノイ大

GRAPEについての発表。

(a)GRAPEは2000年までにペタ実現、汎用に比べ3-10年進んでいる。

(b)汎用にすれば安くなるが、このままでも10Mドル未満でできる。

(c)N-体問題はやりやすい。

(d)ソフト的にはなにもやることはない。

(e)G-4について 100MflopsのHOST、30MHz、0.6Gflops/pipeline

1692pipelineで1Tflops---HARP chipはMCM

(f)G-6について0.25 μ m、150MHz、15pipeline/chip ---130Gflops

104Chips-----Petaflops実現

(4) The Illinois Aggressive Cache-Only Memory Architecture Multiprocessor (I-ACOMA)、

J.Torrellas et.al.、イリノイ大

(5) MORPH: A Flexible Architecture for Executing Component Software at 100 TeraOPS、

A.A.Chien et.al.、イリノイ大

(6) Architecture, Algorithms and Applications for Future Generation Supercomputers、V.Kumar

et.al.、ミネソタ大

(7) Hierarchical Processors- and- Memory Architecture for High Performance Computing、

R.Eigenmann et.al.、ノースウエスタン大

(8) A Scalable-Feasible Parellel Computer: Implementing Electronic and Optical Interconnections for 156 TeraOPS Minimum Performance、A.T.Chronopoulous、Wayne州立大

4.1.4 超伝導素子に関する発表

- (1)A/D converter 3.5 μ m rule、7GHz、1778 JJ ができている。
 (2)動作速度、消費電力はCMOSと比較して非常に有利と主張されている。

	CMOS	RSFQ
1bitあたりの消費エネルギー(J)	10^{-13}	10^{-18}
10^{20} bitあたりの消費電力 (W)	10^7	10^2

(3)アーキテクチャの形態

Point Design StudyでSUNYのGaoが発表した形体(100GHz $\times$ 1000processors+超伝導メモリ)
 + (77K 1GHz SRAM) +256M DRAMが再掲されていた。

(4)ロードマップ

西 暦	1998	2001	2004	2007
ルール(μ m)	3.5	1.5	0.8	0.5
ゲート数 (Kgate/cm ²)	10	30	100	1000
電流(kA/cm)	1	6.5	20	50
C(aF/ μ m ²)	45	60	67	75
clock(GHz)	150	300	500	700
LSI動作(GHz)	30	60	100	150
消費電力(μ W/gate)	0.03	0.06	0.1	0.15
cost/gate(millicent)		50	10	1

- (5)高温超伝導素子について感触をさぐる質問に対して、Rikharevは「High Tcは何もできない」と答えた。他にも「ゴミだ」と言う人もありさんざんであった。超伝導素子自体を「SFだ」という人もおり、期待と実際のギャップを感じた。

- (6)大容量メモリについては基本的に半導体メモリを使用すると結論していた。

(7)HYPRES社について

- ・ TATは3~4weeksで10release/yearの実績がある。
- ・ 集積度は20,000JJまでいった。
- ・ high volumeが可能。

- ・ demonstrated yieldとreliabilityを達成できると主張。
- ・ 2KRAM(10K以上のJJ) 、 144GHz counterを作製した。
- ・ クリーンルームの写真を見せられたが、帽子と白衣（クリーンスーツではない）だけでとてもクリーンとは思えない

(8)TRW社について

Super Electronicsとしてとらえている（Power densityに注目）

- ・ 作製技術の信頼性：critical current比が1年に亘り1.0～1.2におさまる。
- ・ クラス10のクリーンルームを使用している。（まともそうである）
- ・ 現在3インチを使用しており、将来は6インチへステップアップする。
- ・ test equipment：小型冷却機を使用しているが、1素子ずつ測定なので、スループットが悪い。
- ・ パッケージは300milで800pHと低インダクタンスを達成している。
- ・ シンセサイザ（2.02GHz、2000A/cm²、4phase clock）を作製した。
- ・ multi Gbit packet SW、10 GHz communication SWを目標としている。
その構成は、Input--GaAs--JJ Amp--SFQ--JJ Amp--GaAs--Laserの階層をとる。

4.1.5 アプリケーションとシステムソフトウェアに関する発表を以下に示す。

- (1)System Software for Petaflops Computers、I.Foster、Argonne国立研究所
- (2)Compiling for Petaflops、B. Carlson、Intermetrics
- (3)The Size of the Digital Universe、S. Hoban、CESDIS
- (4)Why Do Computational Requirements Cluster at 10 Petaflops-Days ?、G. Lake、ワシントン大
- (5)N-body Methods at Petaflops、J. Salmon、Caltech

4.1.6 パネル討論 1 Architecture and Technology Issues

- ・ 司会者と12人のパネラーが議論を先導した。
- ・ 話題としてメモリバンド幅、メモリ階層構造、パーティション、プログラミング・モデル、製作、超伝導デバイス、言語、量子コンピューティングが登場した。

4.1.7 パネル討論 2 Any and all Petaflops Issues

- ・ 問題提起の一人+会場の人々による討論があった。
- ・ ペタフロップスは、1,000,000以上の並列性の中味は、1,000以上のユニット×1,000以上の内部並列性である。
- ・ デザインの検討にコストモデルが必要

- ・ユーザがメモリ階層をコントロールできる？という意見とすべてが自動化されるべき？という意見が出た。

4.2 シンポジウム（テクニカルプログラム）

講演、パネル討論、一般発表で構成され、食事中にスピーチがなされた。

4.2.1 講演 1 : From ASCI to Teraflops

ASCIは、DOEのテラフロップス・マシンプログラムである。

- (1)目的：核兵器の安全性と性能を予測する技術開発（による予算獲得）
- (2)必要な技術
 - ・仮想製作のための三次元シミュレーション
 - ・流体物理+化学反応+爆発のシミュレーション
- (3)2001年には+30Tflops/10TB、2003年には100Tflops/30TBを実用化する
- (4)1996年度予算は8,500万ドルであり、うちわけではアプリケーションソフトがトップ
- (5)導入されるシステムは次の3システムである。

- ・ Red system : Sandia National Lab.

Intel 1.8Tflops, 9072CPU, 608GB

- ・ Blue systemはSMPクラスタで、hierarchy memory, distributed shared memory, MPIを含み、次の2システムが導入される。

Pacific Blue:Lawrence Livermore

IBM 3.3Tflops, 4096 Power PC, 512node

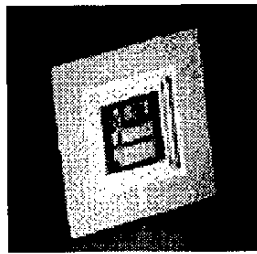
Mountain Blue:Los Alamos

CRAY 3.1Tflops, 3072 Processor, 1.5TB (1998年)

4.2.2 講演 2 : Parallelism in the Deep Blue Chess Automaton

Deep Blueは、IBMが開発したチェスマシンである。

- ・ 1996年の勝敗はコンピュータの1勝3敗2引分けである。
- ・ IBMチェスマシン性能は汎用のパラレルマシンの10Teraflopsに相当する。
- ・ Deep Blue Chess Chipと呼ばれるチェス専用チップを開発した。
- ・ RS6000/SPからVulcanネットワーク経由で利用している。



(<http://www.chess.ibm.park.org/deep/blue/dbphotos.html>)

Deep Blue Chess Chip

4.2.3 Independence Day

- ・大型機小型機を含め、Unix機メーカーが、結局勝つことを映画にたとえた。
- ・Post RISCはMultiple Independent OperationとSingle Cycle Dispatch & Executionである。
- ・Exemplar SPP2000 と HP PA8XXXを紹介
- ・2000年のスーパーコンピュータはテラフロップスマシン、1TBメモリ、汎用MPP、Unix OS、フォートランとC言語で構成されると予測。

4.2.4 パネル討論 1 : How Do We Break the Barrier to the Software Frontier ?

- ・研究投資だけでなく開発投資が必要
- ・チャレンジとは Extreme Scale Parallel SystemとNetworked Virtual Supercomputerである。
- ・新しいスタートのために、新プログラミングモデルが必要
- ・問題は、まずいアーキテクチャである。
- ・HPCシステムソフトウェアテクノロジセンタをスタートする。
- ・現時点では、優れたデザインより原理原則のほうが重要

4.2.5 パネル討論 2 : Petaflops Alternative Paths

H/W : COTS ? 安く早くできる ? アーキテクチャは ? レイテンシは ?

S/W : 10万以上のプロセッサをどうする ?

という問いに対して、次の解答があった。

- (1) PIMがソリューション
- (2) CMOSは最終コンポーネント
- (3) DSM、COWアーキテクチャが条件を満足
- (4) 答えは、Mixed Heterogeneous Computing
- (5) Superconducting Deviceに期待

(6) ペタ・マシンの性質

- ・ 1-100 T B メインメモリ
- ・ 1PBセコンダリメモリ
- ・ 1TB I/O
- ・ +60,000CPUs×1M parallelism
- ・ メモリ部分で100,000個のピン

(7) アプリケーションの候補

- ・ Just in time aircraft design and manufacturing
- ・ Designing new materials
- ・ Modeling U.S. economy
- ・ Cosmology
- ・ Genome comparison
- ・ Drug design
- ・ 3D heart models
- ・ Real time atmospheric turbulence computation
- ・ 他

4.2.6 一般発表

以下のテーマに関係する発表があった。

- ・ Scheduling
- ・ Routing
- ・ Applications & Algorithms
- ・ Petaflops Computing / Point Design Studies
- ・ SIMD
- ・ I/O Techniques
- ・ Memory Management
- ・ Synchronization
- ・ Networks
- ・ Performance Analysis

4.3 食事中的スピーチ

- ・ Ceremony to transfer MPP to Smithsonian Institution
MPPがスミソニアン博物館に展示されるために、その鍵を受け渡しするセレモニー
- ・ Direction and Goals in Advanced Computing

- A perspective on the National Agenda

このスピーチに対し、日本人以外の出席者から、「日本との共同研究について、意見を聞きたい」という質問があった。その回答に期待したが、共同研究の是非についての回答はなく、「ペタフロップスを目指すシンポジウムに、これだけの人が集まったという事実が大きい」という抽象的な回答に終わった。

5. 入手資料

- ・ Frontiers'96 Proceedings
- ・ PetaFlops Systems Workshops Proceedings

付属資料 3 Frontiers '96参加報告書 電子技術総合研究所 関口智嗣

The Sixth Symposium on the Frontiers of Massively Parallel Computingという正式名称を持ち通称Frontiers '96と呼称されるIEEE主催の会議に参加したのでその報告を行う。

本会議は10月27日(日)から31日(木)までの5日にわたり、前半27日と28日はワークショップ形式、後半の3日間はシンポジウム形式で開催された。報告者は日程の都合から後半のシンポジウムにのみ参加した。

会場となったアナポリス市(Annapolis)はメリーランド州(Maryland)の州都であり、米国首都ワシントンDCから40分ほどの郊外にある。また、ここにはNAVAL Academyや軍港も近くに擁した首都圏の軍事拠点でもあり、水兵さんの姿がよく見かけられた。しかし、町そのものはこじんまりとした米国では数少ない歴史を感じさせる町でもあった。それも、そのはずで首都がワシントンDCに移る以前に半年ほどであるが連邦の首都として位置づけられたことがあったようだ。このような地理条件で開催されたのは、このワークショップ主催の中心となっているグループが主にNASA関係であることが大きく影響していると思われる。すなわち、NASA Goddard Space Flight Centerが同州のGreenbelt市にあり、会場からは30分程度のドライブで通えるという地理関係にあること。さらに、先にも述べたように、ワシントンDC(ならびに郊外)まで1時間程度の距離にあるため、NASA関係はもとより、エネルギー省、DARPA等のファンディングエージェンシーからの参加が容易であり、また非常に目立ったことも特徴であろう。この他にも省庁からのリタイア組が多い地元のGeorge Mason Universityから年配(ご高齢)の先生方が参加され、元気に討論に参加しておられた。

参加者は130名程度、そのうち米国以外からの参加は日本からわれわれを含めて7名、中国から1名、オランダから1名であった。日本人の比率が高く、Massively Parallel Computingに対する興味と期待の高さを物語っているのであろう。

会議のセッション構成は以下の通りである。

招待講演: "From ASCI to Teraflops", John Hopson, ASCI

Session 2A: Scheduling 1

Session 2B: Routing

Session 3A: Applications and Algorithms

Session 3B: Petaflops Computing / Point Design Studies

Panel: "How Do We Break the Barrier to the Software Frontier?"

招待講演: (タイトル失念、IBMチェスマシンDeep Blueの履歴)

Session 5A: Scheduling 2

Session 5B: SIMD

Session 6A: I/O Techniques

Session 6B: Memory Management

Panel: "Petaflops Alternative Paths"

招待講演: "Independence Day", Steve Wallach, HP-Convex

Session 8A: Synchronization

Session 8B: Networks

Session 9A: Performance Analysis

Session 9B: Petaflops Computing / Point Design Studies

今回でMassively Parallel Computingに関しては10回目の記念大会と位置づけられていた。今年からSteering Committeeも入れ替わったため、歴代の実行委員長などが勢揃いしたLunchonやBanquetにおいて、しつこいくらいに先駆者たちの先見性を称えていた。日本でもSWoPPが来年でちょうど10年目であるから、ほぼ同程度の歴史を持っているといえよう。

本シンポジウムのセッション構成の特徴としてはPetaflops Computing / Point Design Studiesというのがあげられる。これは、シンポジウムを通じて2セッション、8件の発表があった。これは今年度からNSFが総額100万米ドル程度の予算を立てて、調査研究を公募し採録された8件の提案書に関する進捗状況報告会という位置づけであった。具体的な8テーマについてはすべてフォローできなかったが、いくつかのテーマとして

1. GRAPEを用いたシミュレーションエンジン(Princeton他)
2. ジョセフソン接合を用いた高速デバイス(Caltech他)
3. Illinois Aggressive Coma (U of Illinois, CS)
4. Polaris Compiler (Purdue他)
5. PIM: Processor In Memory (Stanford?)
6. 光デバイスによる高速(広域?)ネットワーク

などが確認できた。ただ、予算額からも想像できるようにまだ実現に向けて具体的な話しがあるわけではなく調査基礎研究としての提案に過ぎないようである。目標は今後20年でPetaflopsを実現するための要素技術開発である。

このシンポジウムの雰囲気としては、先行したワークショップでかなり議論が尽くされたようで、あまり活発なものではなかった。参加者の誰かの発言であったが、「われわれは今、遠い星を射ようとしている。そのために、最初にやることべきことは月を正確に射る技術ではないか」と。すなわち、誰もが目標となるゴール(ペタフロップス)は理解できる、がそのためのアプローチがバラバラでこれでは月にも到達することができないとの懸念なのであろう。確かに、それぞれのプロジェクトが掲げるゴールははるか遠方に設定してあり、すべてが実現することは非常に困難である。仮にビッグプロジェクトとなるのであれば、これら

の要素技術をいかに組み合わせしていくか、またコーディネートしていくかという手腕が問われるであろう。

パネル討論について。非常にタイトルとしては興味深い2つのパネル討論が行われた。まず、"How Do We Break the Barrier to the Software Frontier?"というものであった。Paul Messina (Caltech) がチェアをし、パネラはIan Foster (ARNL), John Salmon (Caltech), Bob Lucas (DARPA), Bill Carlson (IDA-CCS), Rod Oldehoeft (DoE) であった。討論の概要としては

1. 実はまだ本当の障害はまだわかっていない。というのも、どのようなアーキテクチャでいくのか、メモリ構成、システム構成、デバイス技術などまったく何も仮定されないこの状況でソフトウェアの隘路は議論できないとの声。
2. (いつもの議論であるが) レガシーコードをどのように扱うか。もちろん、こうしたコードは扱う必要なしという声が多数。反論としてHPFなどの活動があげられるが、ペタフロップスのI/Oなど、どのように想定できようかと再反論。
3. オブジェクト指向型言語についてはHPC++のような活動を加味してもまだまだ実用に耐えられないし、今後も難しいであろうとの声。これよりも現在のFortranコンパイラからの並列性抽出と制御の方が重要ではないかとの指摘。やはり関数型言語だ！との声はJack Dennis氏であったか(?)

といったところが、議論されていたが、やはり具体的な論点に欠けていたと思われる。"Petaflops Alternative Paths"については、NASA Amesで以前からベンチマーク・性能評価研究をやり、最近ではNas Parallel Benchmarkで著名なDavid Baileyが司会を務めた。JJデバイス技術から分散ソフトウェアまで幅広く人材を集めたが、かえって散漫する結果を招いたといえる。いずれにしても、こうしたテーマでのパネル討論は「いいっぱなし」になるので非常に進行が難しい。さて、内容としては、

1. こうした議論の前にアプリケーションをまず考えなくては、という声に対して、軍需、セキュリティ、地球シミュレーション(?)という解があげられた。仮に、ペタフロップスマシンをJAPANが最初に作った場合に前2者の用途に使えるのかという冗談交じりの質問が出たが、「コピーを作って、米国企業が売ればよい!」とかわされた。
2. 本来の議論に戻って、単独のマシンとしてペタフロップス級のものができなかったとし

たらという前提で、今後マルチプラットフォーム、マルチアーキテクチャ、マルチソフトウェアモデルを統合するソフトウェア基盤システムが重要であり、それを用いて分散コンピューティングを実現するのだという意見に皆うなづく。

3. それぞれの技術に対する個別の質問がなされ、JJが思ったより進んでいるのに驚く。

後藤先生の磁束量子ジョセフソンのアイデアに基づくものであるようだ。GRAPEをJJで作るのか！？との声はパネル終了後の意見。

招待講演ではやはりJohn Hopson (Los Alamos National Lab.) の "From ASCI to Teraflops" と控えめな表題が注目を集めた。ASCIとはAccelerated Strategic Computing InitiativeというDoEがその存在を賭けて始めたビッグプロジェクトである。これは、核実験全面停止条約の世界的合意を受けて、地下実験全廃、新規開発禁止となるため、現在保有の核兵器をどのように管理・処理を行っていくかを計算機にてシミュレートしようという壮大な計画である。ちょっと考えてみただけでも、フェムト秒スケールの核反応から年のオーダーの放射能による影響までをとりこもうというのであるから、その複雑さは地球シミュレータの比ではない。2005年をめどに、3D-Full Physicsを取り込んだアプリケーションの開発と計算機での実現をDoE核兵器関連3研究所 (Sandia, Lawrence Livermore, Los Alamos) の連携強化により推進する計画である。このあたりに、DoEの生き残り戦略が見えてくる。「超並列計算機」という立場でみると、すでにASCI Red (Sandia), ASCI Blue (Pacific: Livermore, Mountain: Los Alamos) という2つ(実質は3つ)の調達を完了している。DoEの責任者Paul Smith氏によれば、「計算機技術の研究開発は直接実施せず、すべて民間からの調達として遂行するのが大きな特徴であると語ってくれたが、この手法は以前、我が国が「大型研究開発制度(大プロ)」で実施し、米国から「官民癒着・護送船団」などと散々非難されたものと対象こそ違え、まったく同じ物ではなかろうか。

いずれの調達もまず今年度中にプロトタイプを納入し、その後、リフレッシュするという手順になっている。RedはIntelが受注し最終的に9,072台の200MHz Pentium Proを接続し、Paragonのような設計にするとと思われる。Pacific BlueはIBMが受注し、8台構成SMPをノードとしてSP-2のような高速スイッチにより512ノードまで接続する計画である。Mountain BlueはCRI/SGIのORIGIN2000を採用し、3,072台構成とし、Los AlamosのACL(Advanced Computing Lab.)に納入される1,024台システムと最終的には協調運転を行い、4,096台システムとして稼動することを条件としている。ORIGINはStanford DASHのアーキテクチャを踏襲した商用開発版といえる。ASCIは商用システムをベースとしているため、着実なマイルストーンが設定してあるともいえる。

個別の講演としてはイリノイ大のA-ICOMAアーキテクチャ(KSRに酷似)やテネシー大の行

列演算におけるFault Toleranceに関するものが興味を引いた。

なお、11月17日より米国ピッツバーグにて開催されるIEEE/ACM Supercomputing'96においても"Petaflops Computing"と題したRound Table (パネル討論のようなもの) が開かれる。ここでも、Messinaがコーディネートし、今回のシンポジウムの報告と、今後の動向について議論される予定である。また、ASCIについても詳細な講演シリーズがあるようだ。

以上、文中にも記載した通り、目標はわかっているが、最初の一步、マイルストーンの設定に誰も解を見いだせないでいるのが現状ではなかろうか。そのために、とりあえずは予算措置のひとつと位置づけている研究者も多い。また、例えばASCIのようにアプリケーションを絞り込むのが技術開発においても重要となろう。我々の場合には何とするか、今後の委員会で議論が待たれるところである。

本書の全部あるいは一部を断りなく転載または複写（コピー）することは、
著作権・出版権の侵害となる場合がありますのでご注意ください。

ペタフロップスマシン技術に関する調査研究

© 平成9年3月発行

発行所 財団法人 日本情報処理開発協会

先端情報技術研究所

東京都港区芝2丁目3番3号

芝東京海上ビルディング2階

TEL (03) 3456-2511

