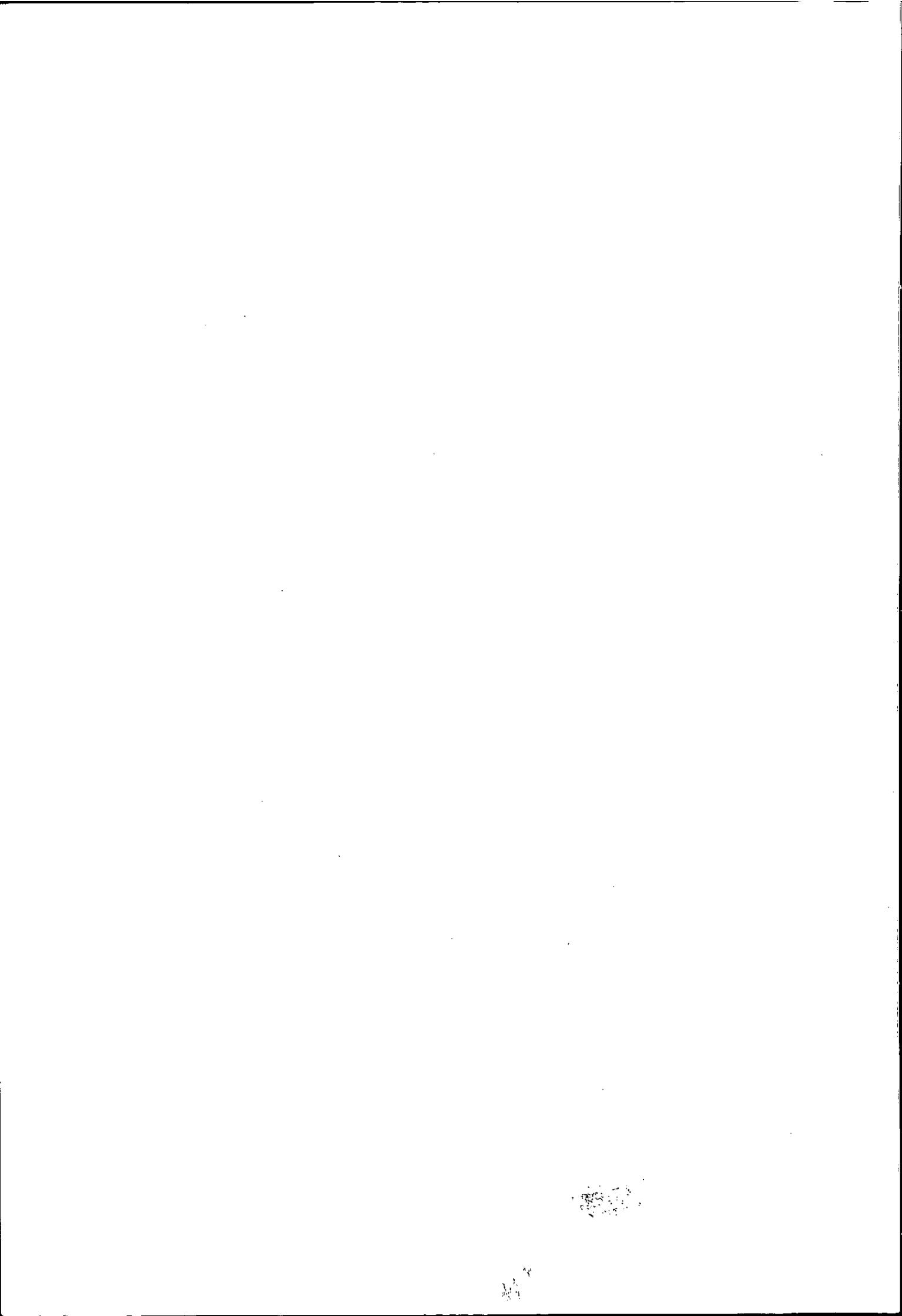


資 料

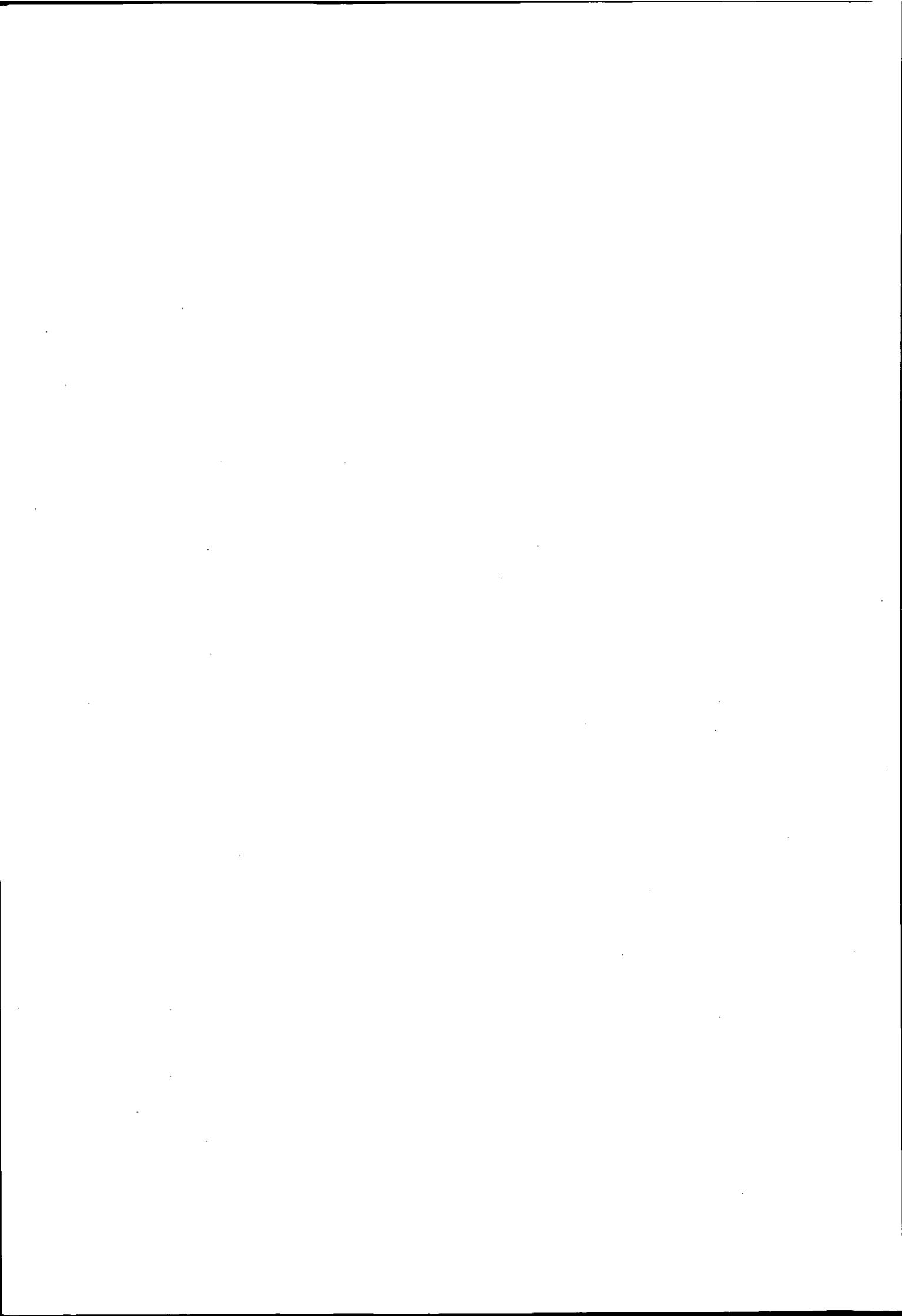
ペタフロップスマシン技術に関する調査研究Ⅱ

平成10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



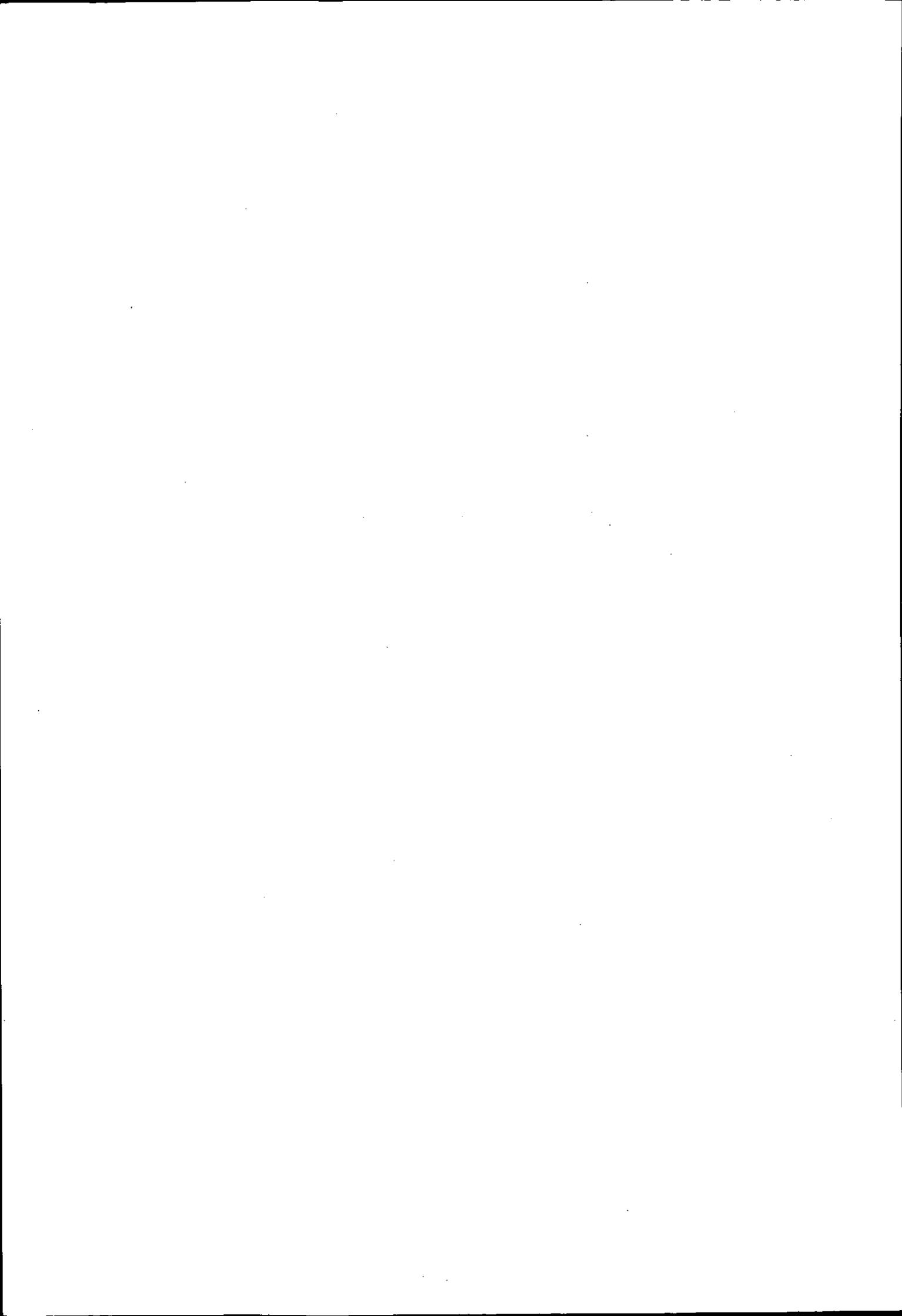




KEIRIN



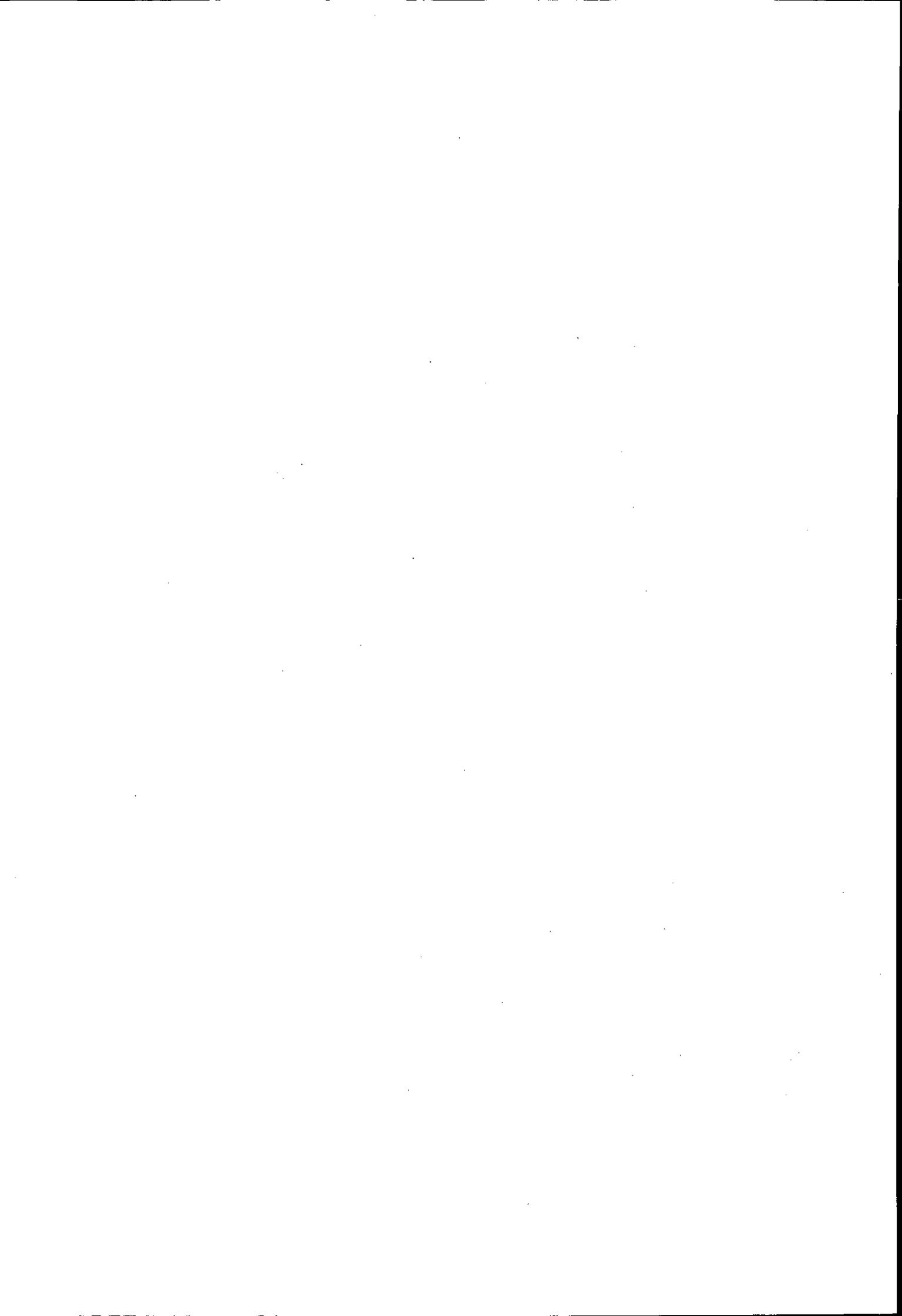
この事業は、競輪の補助金を受けて実施したものです。



# ペタフロップスマシン技術に関する調査研究 II

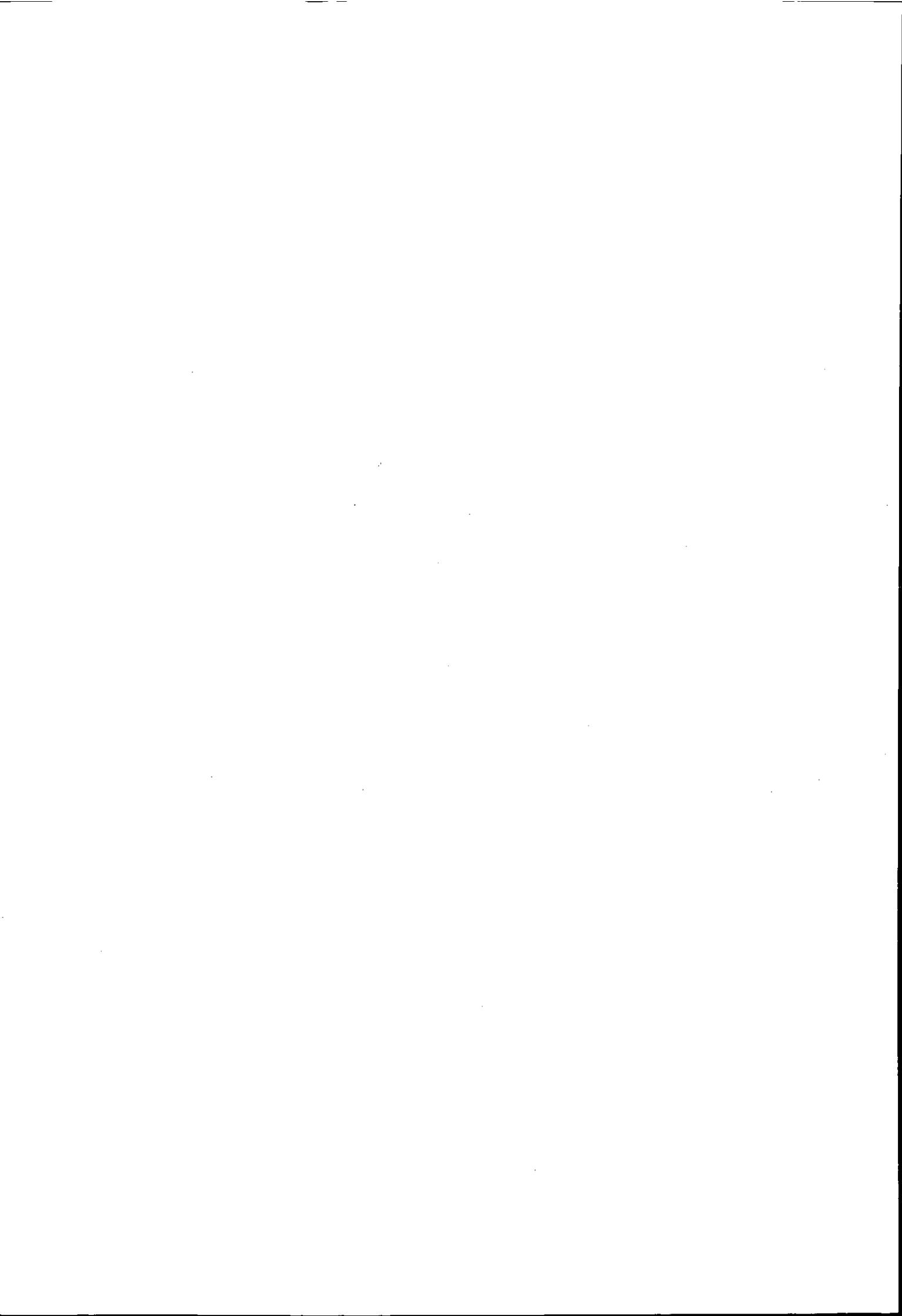
## 目 次

第1章 まえがき	5
第2章 米国の研究開発動向	7
第3章 わが国における研究開発の課題と展望	
3.1 アーキテクチャ	
3.1.1 直接解法マシン (Custom Computing Machines) (天野委員)	35
3.1.2 ペタフロップスマシンに実性能を伴わせるためには (久門委員)	43
3.1.3 高精度予測と高性能計算 (中島浩委員)	49
3.1.4 ペタフロップスマシンに向けた技術開発のありかた (濱中委員)	58
3.2 ソフトウェア	
3.2.1 System on a Chip化時代のシステムソフトウェア基盤技術 (石川委員)	64
3.2.2 並列化コンパイラにおける研究課題と展望 (笠原委員)	69
3.2.3 並列プログラミングインタフェースの現状と課題について (妹尾委員)	74
3.3 アプリケーション	
3.3.1 計算科学と産業基盤ソフトウェアの研究開発の課題 (関口委員)	83
3.3.2 大規模並列処理に期待のかかる応用について (中島克人委員)	88
3.3.3 アプリケーションの過去・現在・未来 (福井委員)	96
3.3.4 計算科学のための超高速計算機開発に向けて (横川委員)	103
第4章 技術ロードマップからみた研究開発の方向性	
4.1 アーキテクチャ	
4.1.1 シングルチップマイクロプロセッサ (天野委員)	109
4.1.2 「付加的」論理のロードマップ (中島浩委員)	114
4.2 ソフトウェア	
4.2.1 分散並列システム (石川委員)	116
4.2.2 自動並列化コンパイラ研究ロードマップ (笠原委員)	118
4.2.3 並列言語インタフェース/ツール (HPC分野) (妹尾委員)	119
4.3 アプリケーション (福井委員)	120
第5章 あとがき	121
付属資料	
付属資料1 気象シミュレータ	125
付属資料2 SC97出張報告書	136
付属資料3 HECC研究プロジェクト概要	148
付属資料4 SIA ロードマップ	159
付属資料5 ワーキンググループ資料リスト	166



## 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 主任研究官
- 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  
並列分散システムソフトウェアつくば研究室長
- 委員 笠原 博徳 早稲田大学理工学部電気電子情報工学科 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 NEC C&C メディア研究所 研究専門課長
- 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 豊橋技術科学大学 情報工学系 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 チームリーダー
- 委員 濱中 直樹 (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 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 担当役
- 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

本報告書は、昨年を引き続いて先端情報技術研究所(AITEC)内に設置された「ペタフロップスマシン技術調査ワーキンググループ(WG)」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」に向けた技術開発に関する意見をまとめたものである。本WGは、具体的な調査を行うために、国立研究所、大学、メーカ等の若手研究者、11名から構成されている。委員の方々は、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多忙な方々であるが、時間を割いて議論に参加していただいた。WG内での議論や本報告書の記述においては、各委員の専門的な分野はもちろんのことであるが、そのような専門にとらわれずに、より幅広く活発な議論や意見を提出していただいた。また、AITECを中心に、今年度は米国における技術開発の政策に関して調査を行ったので、その調査結果も本報告書に掲載してある。

昨年度の調査においては、ペタフロップス規模のスーパーコンピュータの開発を目標として仮定した場合の、問題点や疑問点などを洗い出すことを目的とした。これに対して、今年度のWGの目標としては、ペタフロップスを念頭におきながらも、近未来、5年ないし7年後くらいを視野に入れた議論を行うこととした。そのなかで、「今あるものの次にくるべきものは何か、または、何がきてほしいか」についての議論を行い、これをベースに報告書をまとめることとした。その理由としては、ペタフロップスはだいたい先の目標であるのでターゲットが見えにくいという点と、ペタフロップスという遠いターゲットを見ながら、現時点における日本の競争力、技術力を正しく把握するとともに、それを高める方向を示すことの方が現時点での日本の情報処理産業にとってより重要であり、またその方が結局はペタフロップスに向かう近道であると考えたからである。したがって、今年度は、現時点での、あるいは少し先の技術は何かを調査し、これからの国としての技術開発の戦略をどのような方向で行うべきかに関する1つの指針を打ち出す方向での議論を中心にした。

本WGは、平成9年10月から平成10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、まず、第1、2回目の会合において、各委員から、「5年～7年後の技術をにらんで、今何をなすべきか」というテーマで、問題提起をしてもらい、問題点の把握における認識の違いを明らかにした。それ以降は、各委員の方から、専門分野における技術のロードマップを提示していただき、それに基づいて、何をなすべきかについて議論を行った。それらの結果が、本報告書に反映されていると期待している。

本報告書は、本WGで行われた討論をふまえて、5年後の技術を見据えた各委員の見解をまとめたものである。記述にあたっては、昨年度と同様に、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つぎのような点を重視してもらうように要請した。

(1) 中期的 (5~7年) 技術課題の抽出

- ①「今あるものの次に来るべきものは何か、または、何がきて欲しいか」
- ②「高性能コンピューティング技術を必要とするアプリケーションはあるか」

(2) 開発状況の把握

「今走っている、もしくは計画されている研究プログラムは、アプリケーションを意識した研究となっているか (できれば具体例をあげる)」

(3) 政府支援のありかた

「どのような支援が望ましいか (環境、制度、他)」

(4) 日米比較

- ①「技術開発にあたり、米国優位となる日米の差は何か」
- ②「それぞれの専門に近い分野で日本が対等もしくは優位の部分があるか、また、それは何か」
- ③「米国に任せ、日本が手をつけてはいけない分野はあるか、またそれは何か」

なお、各委員の信じる独自の意見だけを示されても、根拠のない意見とみなされた場合には、説得力に欠ける。そこで、各委員からの独自の見解の他に、いくつかの技術についてはそのロードマップをまとめることとした。具体的には、ハードウェアやデバイス関係のロードマップとしてシングルチップマイクロプロセッサのロードマップ、並列システムの潮流やシステムソフト関係の動向を判断するためのロードマップとしてクラスタコンピュータのロードマップを、並列プログラム言語関係のロードマップとして HPF(High Performance Fortran)に関するものを、それぞれ記述してもらった。また、アプリケーションのロードマップにも挑戦してもらったが、これを一般的に示すことは難しいので、アプリケーションを構成する色々な軸をあげ、各軸での方向性を示してもらった。このようなロードマップ自体も、デバイス系を除けばかなり個人的な見解に左右される面も多々あると考えられるが、一般的な技術の傾向を把握し、各委員の見解を補足したり、あるいは、逆の意味で対立点が明確になるなどの効果が期待されると思っている。

また、実際に大規模なアプリケーションに関する将来動向を探るために、本 WG での講師として東京大学気候システム研究センターの木本昌秀助教授を招いて気象シミュレーションに関する話をしていただいた。この講演内容についても本報告書の資料として掲載してあるので参考にさせていただきたい。

(山口喜教主査)

## 第2章 米国の研究開発動向

### 2.1 はじめに

米国における高度コンピューティング関連(ペタフロップス(Petaflops)を含む)研究開発の主体は、CIC委員会(Committee on Computing, Information, and Communications)にあり、特にHECC(High End Computing Committee)が主導している。ここで策定されるプログラムは、省庁連携プログラムであり、法律で規定されたHPCC(High Performance Computing, and Communications)プログラムが進展したものである。現在のCICプログラムは、HPCCがテラフロップス(Teraflops)コンピューティングの実現をめざしたのに引き続き、その3桁上のペタフロップスコンピューティングを視野に置いている。

ペタフロップスコンピューティングについては、長期的視点を持ち、20年を実現期間とする研究開発検討が1991年よりペタフロップスイニシアティブによって検討されてきたが、最近、ついに国家目標として浮上し、計画が加速してきている。

テラフロップスコンピューティングに関して、CICプログラムでは、アプリケーション、もしくはシステムソフトウェア研究に重点を移しており、ハードウェア開発は、むしろDOE(Department Of Energy)のミッションプログラムとしてのASCI (Accelerated Strategic Computing Initiative) が実行されている。

ASCIは、1995年より10年で\$1Bを支出するプログラムであるが、2004年までに100Tflopsを実現するPathForwardプロジェクトが実行されている。2004年以降のASCI計画は明確ではないが、2010年、Petaflopsをねらうという動きも出ている。

本章では、WWW、国際会議、外部委託調査等で得られた情報に基づき、ペタフロップスコンピューティングを念頭に置いて、研究開発関連米国政府組織、予算、研究テーマなど、米国のハイエンドコンピューティング動向について調査した結果を記述する。

### 2.2 米国政府関連の高度コンピューティング研究開発組織

#### 2.2.1 米国の研究開発関連政府組織とプログラムの概略

米国政府の研究開発投資額は、全分野で\$70B、IT(Information Technology)分野では\$2.7Bで、民間からのそれぞれの投資額 \$182B、\$22.5Bを考慮すると全体の28%、11%をそれぞれ占めることになる。この占有率は、政府投資関連研究開発テーマが、アイデア発

掘から調達としての商品開発支援までを網羅し、かつ挑戦的なテーマに投資されていることを考慮すると、決して低いものではなく、むしろ研究開発の最先端を支える重要な役割を担っていると考えるべきであろう。

このような、研究開発への投資を戦略的に行う組織構造としくみを図2.1に示す。大統領の強力な政策イニシアティブがまず存在し、その政策立案・実行の効率を向上させるため、組織の壁をなくすことを考慮した形態となっている。

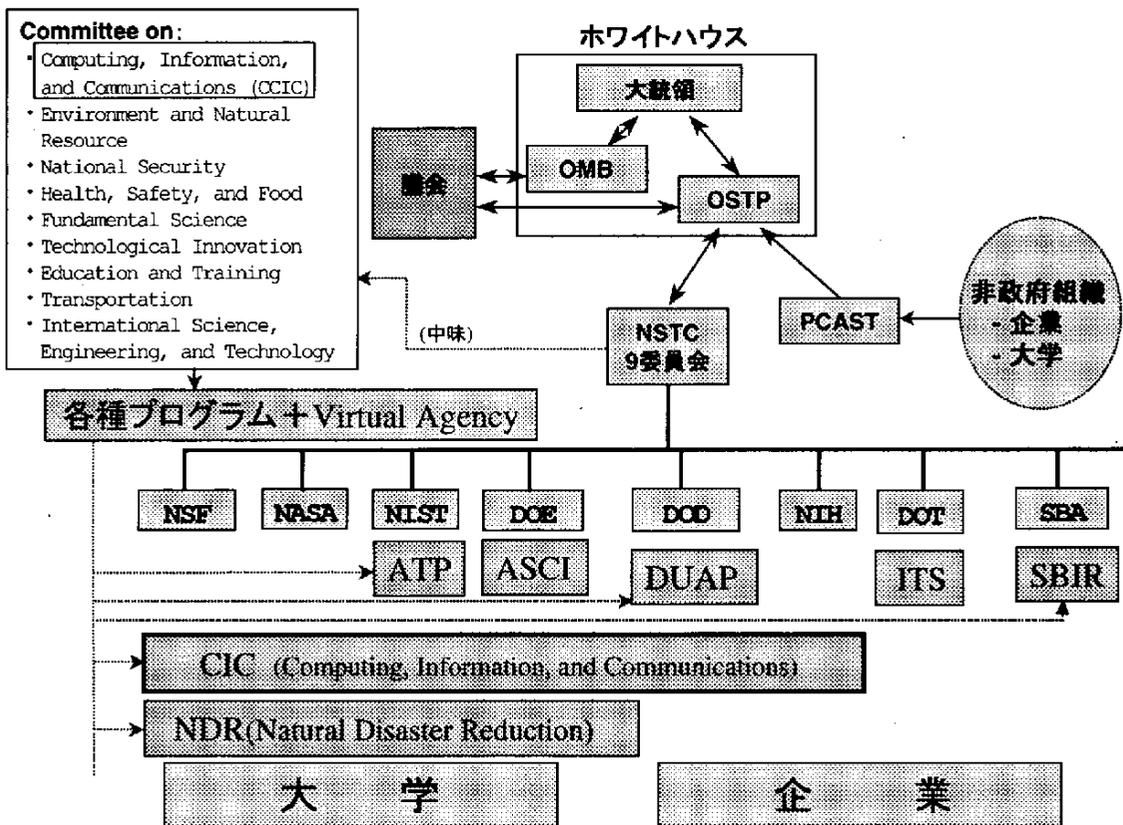


図2.1 米国の研究開発関連政府組織としくみ

図2.1で、大統領府における科学技術政策局(OSTP)は、科学・技術の領域での大統領への専門的アドバイスを行い、かつ国家科学技術委員会(NSTC)を通じて省庁間の調整の支援をする。科学技術に関する大統領諮問委員会(PCAST)は、OSTPのトップと非政府組織（企業、大学）からの委員19名（委員長 John Young）で構成されており、産業界と大学からのアドバイスを大統領とNSTCへ与える。行政管理予算局（OMB）は、財務省と独立しており、予算の作成、各省庁の予算監督、効果評価、議会との折衝を行う。NSTCは、大統領を座長とし、副大統領、科学技術関連の省庁の長など政府高官で構成されるVirtual Agency

で、科学技術政策の戦略的な方向性の決定や、予算の提案を行う。また、図2.1に示すような、各科学的分野毎に9つの委員会を設立し、その座長は、省庁の高級官僚が務める。

NSTC で策定された政策は、省庁が単独もしくは連携して実行する各種プログラムや、中小企業庁(SBA)などの Virtual Agency のミッションとして具現化され、実行される。例としては、

#### 複数省庁実施

CIC(Computing, Information, and Communications)プログラム

: CIC 委員会所管(情報通信)

NDR(Natural Disaster Reduction)プログラム

: Environment and Natural Resources 委員会所管(環境安全)

ATP(Advanced Technology Program)

: NIST(標準・技術院)所管(企業化に向けた研究開発支援)

SBIR(Small Business Innovation Research Program)

: SBA(中小企業庁)所管(中小企業支援)

#### 単独省庁実施

DUAP(Dual Use Applications Program) : DOD(国防総省)所管(民生技術の軍事転用)

ASCI(Accelerated Strategic Computing Initiative)

: DOE(エネルギー省)所管(核兵器備蓄・開発)

ITS(Intelligent Transportation Systems) : DOT(運輸省)所管(次世代輸送システム)

などがあげられる。

### 2.2.2 CIC 委員会 (Committee on Computing, Information, and Communications)

#### (1) 組織

米国の高度コンピューティング領域でのリーダーシップを確保する政策実行が議会から要求されており、High Performance Computing Act (Public Law 102-194、1991)が成立している。この法律に基づいて HPCC (High Performance Computing and Communications) イニシアティブが設立されたが、1997年度、CIC 委員会に改組された。

CIC 委員会の下には技術政策小委員会、アプリケーション委員会、CIC 研究開発小委員会 (CIC R&D Subcommittee) がある。アプリケーション委員会は、26 の省庁の代表が集まり、パイロットプロジェクトなどを通し、共通に使用できるアプリケーション開発推進を行う委員会である。

CIC プログラム策定・実行の中心となる CIC 研究開発小委員会は次の 5 つのワーキンググループ(WG)を有し、政策決定に必要な技術プログラムや予算の立案を行っている。

- ① High End Computing and Computation WG (HECC)
- ② Large Scale Networking WG (LSN)
- ③ High Confidence Systems WG (HCS)
- ④ Human Centered Systems WG (HuCS)
- ⑤ Education, Training, and Human Resources WG (ETHR)

### CIC 組織図

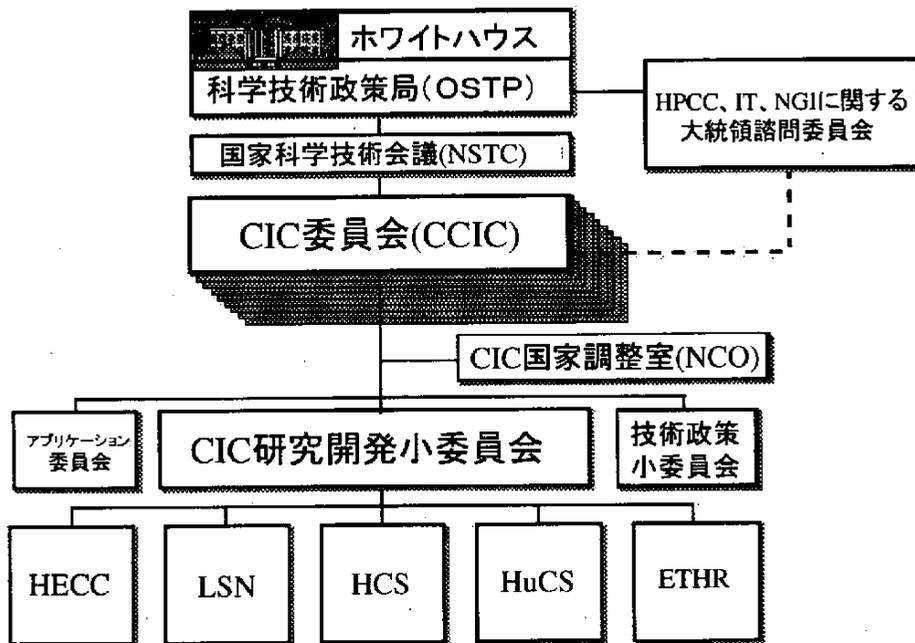


図 2.2 CIC 委員会組織図

#### (2) 予算

上記 5 つの研究開発 WG テーマに割り振られた予算を表 2.1 に示す。

表 2.1 CIC プログラム 研究予算 (\$M)

年度	HECC	LSN	HCS	HuCS	ETHR	合計
1997 年	444.6	252.7	30.0	240.4	40.8	1,008.5
1998 年	463.2	288.3	33.2	281.3	38.7	1,103.7

予算割当てが大きなプログラムは、HECC、LSN、HuCSであり、高度コンピューティング、ネットワーク、人間との関わりについての研究開発に重点投資していることがわかる。

### (3) 組織変更

1997年度設立時から若干の組織変更が行われている。連邦ネットワーク委員会は廃止され、技術政策小委員会が設立された。また、高度コンピューティング、通信、情報技術及び次世代インターネットに関する諮問委員会 (Advisory Committee on High Performance Computing and Communications, Information Technology, and the Next Generation Internet) が設立されている。これは、1997年2月、クリントン大統領の指令により設立された。機能としては科学技術政策局 (OSTP) を通じ、国家科学技術会議 (NSTC) へ上記領域技術での助言や情報を与えるためのものである。また、本諮問委員会は、次の領域での問題を独自に検証し、提言する。

- ① HPCC プログラムの実行経過
- ② 次世代インターネット (NGI) イニシアティブの計画立案と実行の経過
- ③ HPCC プログラムの改善の必要性
- ④ HPCC サブプログラム間のバランス
- ⑤ HPCC とそのアプリケーション分野での米国優位確保に貢献する研究開発かどうか
- ⑥ その他

先に述べたとおり、大統領への助言機関として PCAST があるが、本諮問委員会は上記の領域に特化されたものであり、より専門的、具体的な議論が行われていると考えられる。本委員会の会合は 1997年2月より 1998年3月まで 4回開催されており (第1回 (1997年2月)、第2回 (1997年6月)、第3回 (1997年12月)、第4回 (1998年3月))、その中で CIC 委員会の幹部が CIC プログラムについての方針、状況説明などを行っている。その内容開示状況は、説明 OHP 等が WWW にて公開されており、極めてオープンである。

委員会の構成メンバーは、以下に示す座長 2名、委員 23名で、コンピュータ、ネットワーク、コンテンツの最前線に立つ企業、大学から人選している。

座長： Ken Kennedy (Rice University)

Bill Joy (Sun Microsystems)

委員： Eric A. Benhamou (3Com Corporation)

Vinton Cerf (MCI Communications)

Ching-Chih Chen (Simmons College)  
David Cooper (Lawrence Livermore National Laboratory)  
Steven D. Dorfman (Hughes Electronics Corporation)  
David W. Dorman (PointCast、前CEO of Pacific Bell)  
Robert Ewald (Silicon Graphics, Inc.)  
David J. Farber (University of Pennsylvania)  
Sherrilynn S. Fuller (University of Washington)  
Hector Garcia-Molina (Stanford University)  
Susan Graham (University of California, Berkeley)  
James N. Gray (Microsoft's Bay Area Research Center)  
W. Daniel Hillis (Walt Disney Imagineering, Research and Development, Inc.)  
John P. Miller (Montana State University)  
David C. Nagel (AT&T Labs.)  
Raj Reddy (Carnegie Mellon University)  
Edward H. Shortliffe (Stanford University )  
Larry Smarr (University of Illinois)  
Joe F. Thompson (Mississippi State University)  
Leslie Vadasz (Intel Corporation)  
Andrew J. Viterbi (QUALCOMM)  
Steven J. Wallach (Hewlett-Packard Company's Convex Technology Center)  
Irving Wladawsky-Berger (IBM Corporation)

## 2.2.3 HECC (High End Computing Computation) プログラム部分領域

### 2.2.3.1 研究方針

HECC の研究開発領域は、最先端のブレークスルーを実現する、量子・生物・光コンピューティングの概念の探索に加え、それを可能とする革新的なソフトウェア・ハードウェアを開発することであるとされている。ここでは、物理的、化学的、生物学的モデリングと複雑な系におけるそれらのプロセスのシミュレーションを行うアルゴリズムの研究など、情報技術が中心となる科学技術アプリケーションの研究が重要視される。

このようなハイエンドの要求を満たす高度コンピューティング能力を満たすウルトラコンピュータを開発し、かつそれを商用部品(Commodity Component)で実現することによって、開発結果を速やかに一般市場に浸透させ、米国のリーダーシップを確保することが、研究理念である(図 2.3)。

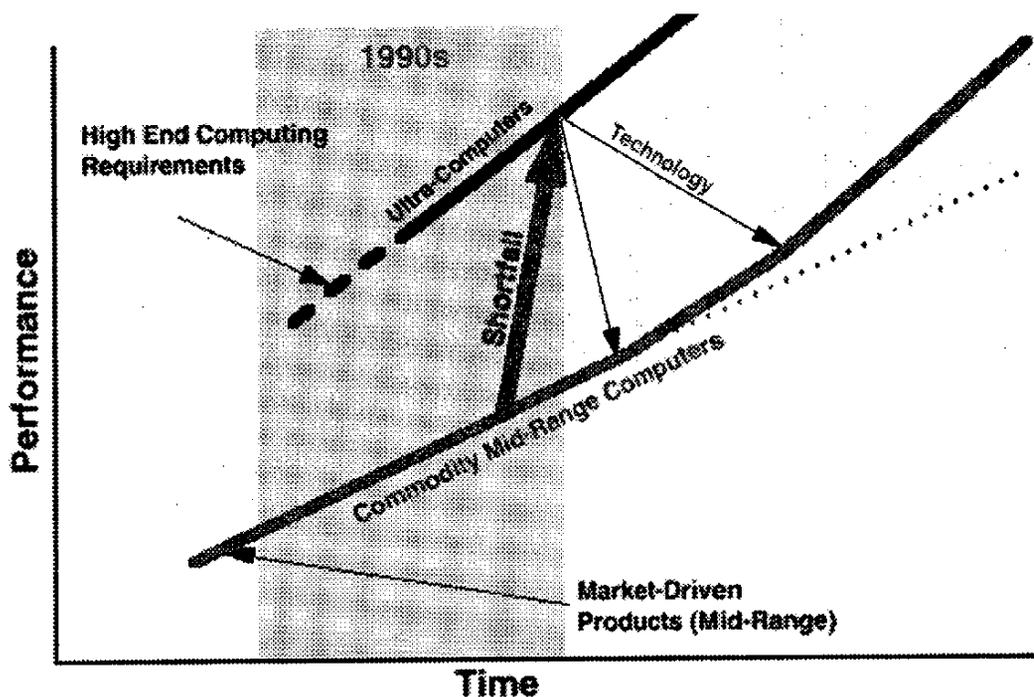


図 2.3 HECC 研究理念

<http://www.ccic.gov/talks/smith-9Dec97/slide/9.html>

HECC 推進事項として、具体的に次の4項目があげられている。

(1) システムソフトウェア技術

政府、産業、学術の広範な領域にわたり、テラフロップスレベルのシステムの使いやすさ、有効性を大きく改善する。

特に、分散、スケーラブルなシステムのための並列ソフトウェアツール、OS、プログラム開発環境、性能モニタリングに焦点を当てており、革新的な言語とそのコンパイラ、デバッガ、性能モニタ、スケーラブル OS と I/O、プログラム開発環境、科学計算の可視化、データ管理を開発プロダクトとしている。大規模データ管理を行うために、エクサバイトストレージから大規模高速メモリへの転送技術が必要とされている。

(2) 未来世代のコンピューティングの先端的研究

ペタフロップスレベルのコンピューティングとエクサバイトレベルの大規模ストレージに必要な研究・技術開発を行う。

まず、共有メモリプログラミングモデルにおいて、マルチスレッド、キャッシュアルゴリズムなどによりメモリレイテンシを克服すると考えられている。研究開発はこれらの技術を支援するソフトウェア、アプリケーション、アーキテクチャ、コンポーネント技術に注力している。また、数百プロセッサで構成される SMP システムから数万プロセッサで構成されるシステムへのソフトウェア移植技術も研究対象である。

コンポーネントとしては低温、高温超伝導デバイス（例えば RSFQ）が Si デバイスとともにロジック、メモリ構成要素として考えられている。プログラマブルな材料による不均一なコンピューティングという問題に対し、技術的、アルゴリズム的、アーキテクチャ的の確立が研究されている。WDM(Wavelength Division Multiplexing)によるオンチップ、オフチップ間の光学的相互接続やホログラフィックメモリによる大規模ストレージも期待されている。ナノテクノロジーによるロジックデバイスや DNA などの巨大分子への情報蓄積を可能とする生物学的材料の開発など基礎研究も行われる。

(3) 実存のアプリケーションへの技術確立

HECC 技術は省庁での使用が最初に行われ、ついで、HPC 施設を必要とする計算科学の領域でより進んだ使用がなされる。キーアプリケーションが完全に実行できる事を保証するインフラを整える。

高度計算科学とキーアプリケーションを完全に動かせるアルゴリズムの開発を行い、また省庁が使用するアプリケーションに焦点を当てる。多くの CIC 関連省庁はその責

務を果たすための科学的アプリケーション開発プロジェクトを支援しているが、大規模計算や大容量データの操作が必要とされている。それらのプロジェクトは、天文学、気象の世界モデル、短期天気予報、蛋白質合成、量子色力学に至る広い領域に及んでいる。計算機科学のアルゴリズム研究は、巨大スパースマトリックス解法、検索・ソート・パターンマッチングなどに必要である。将来にわたり、ハイエンド計算システムで、膨大な並行処理、フォールトトレランス、レイテンシ隠蔽のアルゴリズム研究が必要とされている。

#### (4)ハイエンドコンピューティングの研究インフラ整備

研究のためのコンピュータ環境、すなわち大規模試験システム、高性能コンピュータおよび大規模ネットワークの全体能力を向上させる。このため、均整の取れた大規模最先端ハイエンドコンピューティングインフラの整備を促進する。

最先端のコンピューティング環境整備を計画・実行し、かつ維持していくことにより、計算研究環境、大規模試験基盤、高性能ネットワークの全ての能力をひきだす。CICが支援するセンタは、計算処理能力とネットワーク帯域幅をバランスさせた向上を行い、提供する。ここで確立した設備は、大規模計算を必要とする領域の研究を続けていく上で重要な役割を果たす。

### 2.2.3.2 HECC 研究予算

HECCの研究テーマは、1996年度から1997年度の間でHPCCイニシアティブがCIC委員会に改組した時点で再編された。1996年度時点でHPCS(High Performance Computing Systems)、ASTA(Advanced Software Technology and Algorithms)と分類されていた研究テーマの大部分は、HECCに所属することになった。1996年度から1998年度における予算の変遷を次表に示す。1996年度から1997年度にかけては、改組によりトータルとしては減少しており、スリム化したと考えられる。しかしながら、1998年度要求では増加しており、この分野の研究の重要性が強く主張されていると思われる。

表2.2 HECC関連予算の年度変遷 (\$M)

	1996年度	1997年度	1998年度
HPCS	146.7	HECC 445.0	HECC 462.4
ASTA	371.0		

HECC 関連研究は、NSF、DOE、NASA、DARPA、NSA、NIH、EPA、NOAA、NIST の9省庁・機関でなされており、1998 年度予算では、表 2.3 のように配分されている。ここでは、\$10M を超える予算を獲得している研究分野を同時に掲載する。

表2.3 1998年度省庁別HECC予算 ( )内は1997年度

省庁予算計 (\$M)	\$10Mをこえる研究分野	予算 (\$M)
NSF 132.9 (129.17)	Supercomputer Centers	53.17 (57.73)
	Computing Systems	51.02 (45.95)
	Applications	28.71 (25.49)
DOE 90.80 (93.29)	ACTS Research	90.80 (93.29)
	National Energy Research Scientific Computing Center	26.50 (29.76)
	HPC Resource Providers	16.56 (15.63)
NASA 90.10 (87.20)	Grand Challenge Support	48.60 (56.75)
	System Software	17.00 (7.56)
	Testbeds	24.50 (15.39)
DARPA 84.80 (75.05)	System Environments	84.80 (75.05)
	Scalable Systems and Software	35.20 (27.17)
	Microsystems	15.90 (14.29)
	Embeddable Systems	15.00
	Defence Technology Integration and Infrastructure	(14.00)
NSA 26.42(25.93)	Supercomputing Research	24.20 (23.43)
NIH 23.74 (23.40)	なし	—
EPA 5.38 (6.58)	なし	—
NOAA 4.30(6.30)	なし	—
NIST 3.99( 5.79)	なし	—

NSF : National Science Foundation

DOE : Department of Energy

NASA : National Aeronautics and Space Administration

DARPA : Defense Advanced Research Projects Agency

NSA : National Security Agency

NIH : National Institute of Health

EPA : Environment Protection Agency

NOAA : National Oceanic and Atmospheric Administration

NIST : National Institute of Standards and Technology

### 2.2.3.3 研究プロジェクトの例

HECCにおける研究プロジェクト例として、1998年度版 Blue Bookにトピックスとして掲載されたものを、1997年度版 Implementation Planを参照し、各省庁毎、各項目毎に分類した概略を紹介する（付属資料3に、より詳しい内容を掲載している）。

#### (1) NSF（全米科学財団）

##### (a) Supercomputer Centers

PACI (Partnerships for Advanced Computational Infrastructure)

NSFのスーパーコンピュータセンタを中心に、通常の計算能力の1桁～2桁上のハイエンドシステムを有する2ヶ所のサイトとの共同研究。

##### ① NPACI (National Partnership for Advanced Computational Infrastructure)

California 大学 San Diego 校 (UCSD) が中心。大規模データ解析。

##### ② NCSA (National Computational Science Alliance)

Illinois 大学 Urbana-Champaign (UIUC) 中心。可視化コンピューティング技術。

##### (b) Computing Systems

将来を見通したコンピューティングシステムのハードウェアについての研究。

ベタフロップス関連の研究テーマが、入っている可能性が高い。

##### (c) Applications

高度コンピューティングの必要性のアピール。

##### ① 体積レンダリング技術 (Purdue 大学)

##### ② 非線形電子輸送研究 (ナノデバイス関連研究)

##### ③ Macromolecular Structure (UCSD Computational Center)

#### (2) DOE（エネルギー省）

##### (a) ACTS (Advanced Computational Testing and Simulation)

##### ① Scientific Template Library プロジェクト

(国立エネルギー研究計算科学センタと Argonne、Lawrence Berkeley、

Los Alamos、Oak Ridge の4国立研究所)

##### ② ASCI Academic Strategic Alliance

##### (b) National Energy Research Scientific Computing Center

ここでは、関連が深いと思われるGrand Challenge関連トピックスも含める。

##### ① 軽原子核研究 (多体の原子核相互作用に関する知識と核子の振る舞いの予言)

- ② 数値トカマク乱流プロジェクト (NITP)  
(核融合炉の性能を最適化シミュレーション)
  - ③ 最適設計技術 (感度解析、Argonne 国立研究所)
  - ④ MDH の表面反応性 (りんご酸脱水素酵素 (MDH) の反応機構解析)
  - ⑤ 分子動力学 (クラスタの最適構造探索)
- (c) HPC Resource Providers
- Grand Challengeへの最先端計算機リソース (Intel Paragon、IBM SP2、Cray T3E)  
や、異機種からなるコンピューティング環境の提供
- (3) NASA (航空宇宙局)
- (a) Grand Challenge Support
- ① 計算航空科学プロジェクト (高速度問題、先端亜音速技術、回転翼航空機)
  - ② 地球と宇宙の科学プロジェクト (地球天候、太陽系の動力学、太陽風、宇宙の生成)
- (b) Testbeds、System Software (HECC 研究開発センタ)
- 並列システムソフトウェアツール、アプリケーション開発
- (4) DARPA (国防総省高等研究計画局)
- (a) System Environments
- High Performance C++ (並列プログラミング標準モデル、アプリのポータビリティ)
- (b) Scalable Systems and Software
- ① スケーラブルシステム及びソフトウェア (テラフロップクラスシステムのための先進的技術)
  - ② M-Machine (MIT、細粒度並行動作コンピュータ)
  - ③ National Compiler Infrastructure プロジェクト (共通プラットフォーム開発)
    - ・ Intermediate Format (SUIF) プロジェクト (Stanford 大学、Harvard 大学、Rice 大学、California 大学 Santa Barbara 校)
    - ・ Zephyr プロジェクト (Virginia 大学、Princeton 大学)
  - ④ 商用の工学用ソフトウェアの並列化 (ベンダの投資リスク最小化、米国産業市場へのスケーラブル並列ソフトウェア導入)
  - ⑤ 並列システムソフトウェアツール (ポータブル、産業標準と合致するソフトウェア)
- (c) Embeddable Systems (分類名が1998年度から新設)
- Quorum (グローバルな異種分散コンピューティング、OS への統合)

(5) NSA (国家安全保障局)

Supercomputing Researchプログラム(知的処理の計算機支援を一桁改善する)

(6) NIH (国立衛生研究所)

① バイオメディカル研究 (National Center for Research Resources)

② Biomolecular computing (蛋白質構造・他の分子・薬品の構造的、機能的変化の予測)

③ 分子生物学計算と AChE シミュレータ (アセチルコリンエステラーゼの解析)

(7) EPA (環境保護庁)

科学における可視化 (大気と海洋シミュレーション結果の出力の可視化)

## 2.3 ペタフロップスイニシアティブ (Petaflops Initiative) の動向

### 2.3.1 ペタフロップスイニシアティブの活動状況

Petaflopsの性能を有するコンピューティングシステムの研究開発検討は、1991年、Purdue大学で開催されたHPCCのグランドチャレンジに関するワークショップに始まり、関連ワークショップ、シンポジウムが毎年開催されてきた。この活動を先導してきたのは、NASAを始めとする NSF、DOE、DARPA、NSA、BMDO(Ballistic Missile Defence Organization)という、高度コンピューティングシステムを最も必要としている省庁によって推進されるペタフロップスイニシアティブである。

ペタフロップスコンピューティングに関する研究開発は、1996年前半まではペタフロップスイニシアティブの活動の範囲を出ていなかった。しかし、1996年中にはNSFのPoint Design StudyのAwardが発表され、CIC委員会予算編成時(1996年末)の幹部のプレゼンテーションなどでワークショップ開催などの実績を強調し始めていた。1997年11月に出されたBlue Bookでは、ついに、HECC(CIC)プログラムのターゲットとして明記されるようになり、さらに、前出のPetaflops Algorithms Workshopの報告がNCO(CIC国家調整室)ホームページに直接掲載されるようになった。ペタフロップスイニシアティブの活動を集約したPETAホームページがNASAに設置されているが、NCOとの関連は単にリンクが張られていたのみであった。今回の直接の掲載は、HECCにおいて、本格的に軸足が変わった証だと考えられる。

公表された活動として、選ばれた専門分野をターゲットとしたワークショップが開催されており、システムソフトウェア、アーキテクチャ、アルゴリズムに関するワークショップが以下のように開催されている。

- ・Petaflops Summer Study on System Software(PetaSoft) (1996)
- ・Petaflops Architecture WorkShop(PAWS) (1996)
- ・Petaflops Algorithms Workshop (1997)

また、シンポジウムとして、1996年に第6回超並列処理フロンティアシンポジウム(Frontiers'96、IEEE)が開催されているが、1997年には開催されなかった。次のFrontiers'98は、1998年11月にAnapolisにて開催される模様である。

一方、NASAのSaini氏によると、"Enabling Technologies for Petaflops Computing"の第2版が近々出版される予定である。

研究テーマへの具体的な連邦予算適用は、前出のNSF Point Design Studyによる

100Tflopsを実現する8テーマへの支援があったが、それに加え、次のようなFundingが確認できる。

- ・ DARPA ITO (Information Technology Office) 支援研究  
研究テーマ: Hybrid Technology MultiThreaded Computer Architecture for Petaflops Computing  
支援対象者: T. Sterling (JPL), K. Likharev (SUNY), P.M. Kogge (Notre Dame), G. Gao (SUNY) 他
- ・ NASA Grant NAG5-2998  
研究テーマ: PIM Architectures for Petaflops Computing  
支援対象者: P.M. Kogge (Notre Dame) 他

### 2.3.2 ペタフロップスイニシアティブの研究方針

ペタフロップスの実現を牽引する要素として次の4つが考えられている。

- ・ 自然進化 (Natural Evolution) : 現在のトレンドの延長技術
- ・ データインテンシブ (Data Intensive) : アプリケーションからの要求
- ・ 近道 (Rapid Path) : 用途限定マシンによる達成
- ・ 跳んだ技術 (Leap Technology) : コンポーネント分野の飛躍

これらの要素は、開発成果物の交換を通じて相互に関連しており、図2.4に示すような概念が示されている。

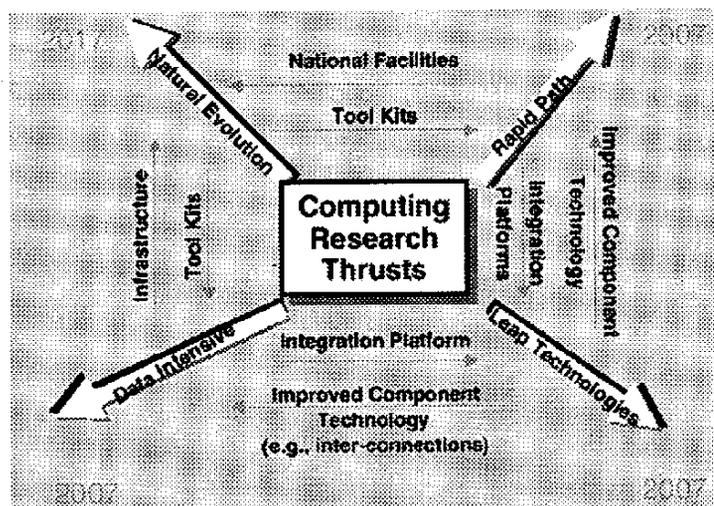


図2.4 ペタフロップスに向かうための要素

<http://www.ccic.gov/talks/petaflops-24Jun97/slide29.html>

ここで、自然進化道程として考えられているシステムは、

- ① Pure COTS (Beowlf-class PC クラスタ、NOW/COW)
- ② COTS plus (例、CRI T3E、HP Convex SPP-2000)
- ③ トレンド

1998年：コスト \$ 20,000M、消費電力 1,300MVA

2010年：コスト \$ 500M-1,000M、消費電力 30MVA

であるとされており、これを2007年までに、どのようにして性能アップしていくかの概念が必要である。コンポーネント技術、プラットフォーム、ソフトウェアツールキット、インフラの確立を媒介として4要素を互いに発展させようとしていると考えられる。

### 2.3.3 現行の技術動向

CIC委員会での幹部の説明では、1996年NSF主催の Point Design Studiesで提案された手法を中心とした技術が向かう方向として例示されている。

特に、ハードウェアの分野で期待されている研究として次のような項目が挙げられている。

- (1) システムのレイテンシを隠蔽するPIM(Processor In Memory)

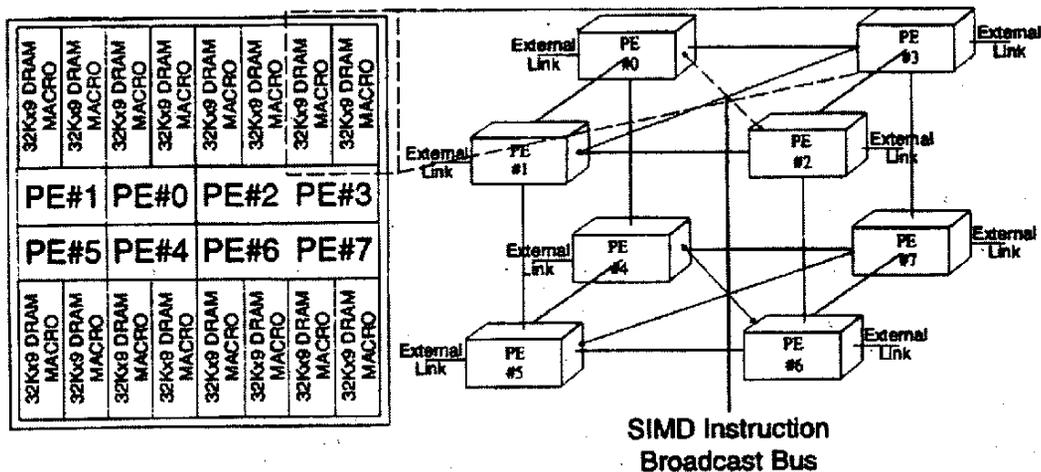


図2.5 PIMイメージ

<http://www.ccic.gov/talks/petaflops-24Jun97/slide17.html>

(2) HTMA (Hybrid Technology Multithreaded Architecture)

CMOSデバイス開発のロードマップを打破する近道として、超伝導素子 (RSFQ) が期待されている。そのシステムイメージは、階層的メモリアーキテクチャであり、極低温での超伝導素子+超伝導メモリ、液体窒素温度冷却のSRAM、室温動作DRAMというものである。

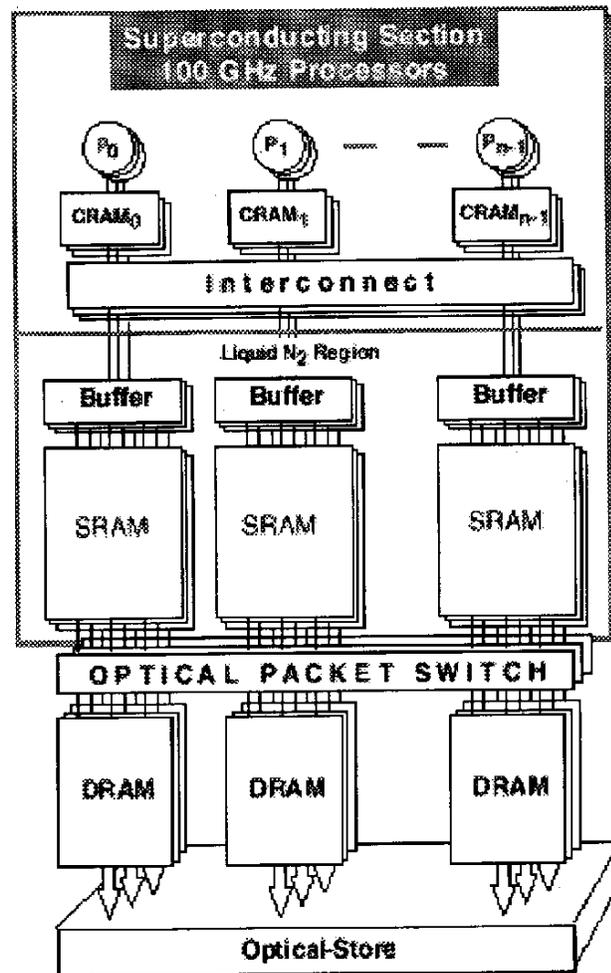


図2.6 Hybrid Technology Multithreaded Architecture

<http://www.ccic.gov/talks/petaflops-24Jun97/slide24.html>

(3) ストレージ技術：光学的ホログラフィ技術

期待される性能：TeraByte規模、100Gb/sのバンド幅、超低消費電力

使用される物性：光屈折、分光のホールバーニング

使用される技術：Near term

## (4) 光学接続

表2.4 予想される光学接続性能

	例	伝送速度	消費電力	光源
現状	HPとAT&A SONET	100 Mb/s	1-2 Watts	LED*
		2.5 Gb/s	10 Watts	LD*
10年後		100 Gb/s	1 Watts	LD (WDM or TDM)*
20年後		1Tb/s	10 Watts	LD (WDM and TDM)*

\* LED(Light Emitting Diode)、LD(Laser Diode)

WDM(Wavelength Division Multiplexing)、TDM(Time Division Multiplexing)

## (5) Grape-6

宇宙・天体力学分野の専用マシンとしてのGrapeは、最も早くPetaflopsを実現することが、期待されている。汎用マシンと比較して問題によっては10年のリードをしていると考えられ、2000年のPetaflops実現をめざしている。

## 2.3.4 将来に向かっての課題

現在、ペタフロップスイニシアティブで、問題として挙げられている項目を以下に列挙する。

デスクトップを含めた全てのクラスのコンピューティングにハイレベルの並列性は避けられないと考えられているため、これらの問題点は、単にハイエンドコンピューティングだけでなく、全てのコンピューティングについて生ずる問題であり、問題の解答は、大規模科学計算の領域をはるかに超えたインパクトを持っているとされている。

## (1) ハードウェア

- ① COTSハードウェアコンポーネントを用いて、使い物になるペタフロップシステムが構築出来るか？
- ② RSFQ超伝導素子を光学結合するような技術は、COTSによるシステムに勝てるか？
- ③ 消費電力は受け入れられるレベルまで落とせるか？
- ④ 分散メモリのMIMDは、満足なものか？もしくは、何か他のアーキテクチャが必要か？
- ⑤ レイテンシと多重のメモリ階層を管理するのに、どんなハードウェアが必要か？

(2) ソフトウェア

- ① どんなOSが 10万から100万プロセッサを信頼性良く管理できるか？
- ② 根本的に新しいプログラミング言語は必要か？もしくは、現在ある言語の拡張でいけるのか？
- ③ 特化した新しい言語構造は必要か？
- ④ I/O、デバッグ、可視化を取り扱う最も良い方法は何か？

(3) アルゴリズム

- ① 既知のアルゴリズムのlatency-tolerant variantsは存在するか？
- ② 各種アルゴリズムの作用回数、メモリ要求、データ局在等の特性は将来システム上でscaleできるか？
- ③ 古典的アルゴリズムで満足か？あるいは、完全に新しいアルゴリズムが必要か？

(4) アプリケーション

- ① 10万を超える並行スレッドを必要とするペタフロップスアプリケーションを構築できるか？
- ② 提案されているシステムデザイン上で多様なアプリケーションを実現する最適な方法は何か？
- ③ 将来アプリが必要なI/O、メモリは何か？
- ④ ペタフロップスシステムによってどんな新アプリケーションが動かせるか？

## 2.4 ASCI (Accelerated Strategic Computing Initiative) プログラムの動向

## 2.4.1 ASCI計画

ASCIは包括的核実験禁止条約の締結により、米国での核兵器の安全性、信頼性、性能維持あるいは新規開発続行を可能とすべく、クリントン大統領が提唱した Nuclear Weapons Stockpile Stewardshipなるプログラムに対し、科学的アプローチの重要な位置を占める計算機シミュレーション開発プログラムである。

このプログラムは、国家安全保障会議(National Security Council)でとりあげられたDOEの存続がかかるミッションプログラムで、NSTC内ではCIC委員会よりCNS(Committee on National Security)での議論が先にあり、CICとは別にスタートしたように推測される。

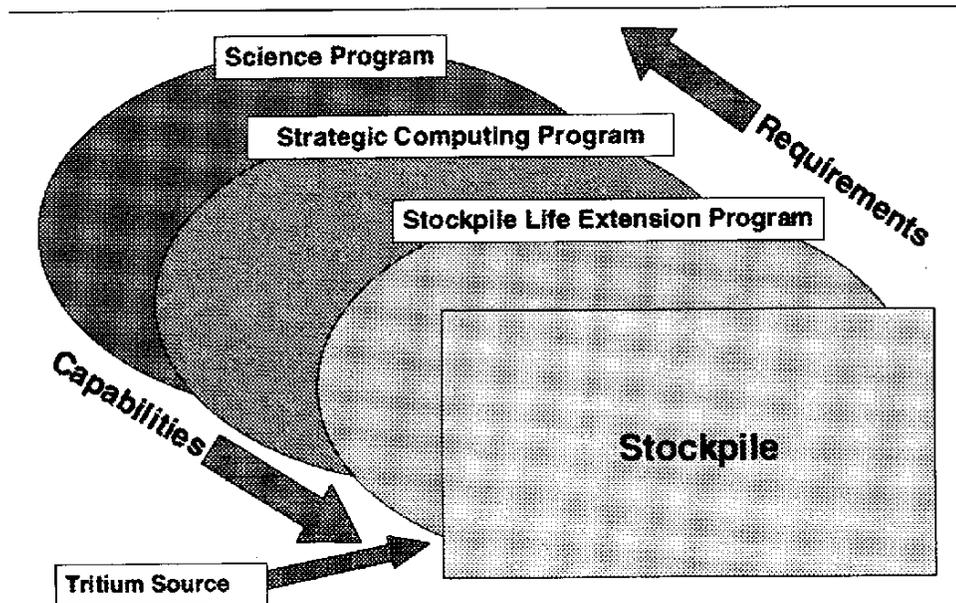


図2.7 Scientific Stockpile Stewardship

<http://www.llnl.gov/asci-alliances/pp/art/fig11.gif>

図2.7は、Scientific Stockpile プログラムの要求と、それを実現する科学的プログラムとの関連を表している。この中で、Strategic Computing Programというのが、ASCIであり、その背景には、解決すべき広範な科学的問題が存在している。

当初、DOEは、このプログラムの実行に、10年で10億ドルの投資を予定し、1995年6月よ

り開始した。2004年までに100Tflopsのコンピューティング性能を実現することを目標としたが、現在、10-100 Tflopsのレンジを視点に置いたPathForward プロジェクトが具体化されている。

ASCIプログラムのマイルストーンの詳細を図2.8に示す。

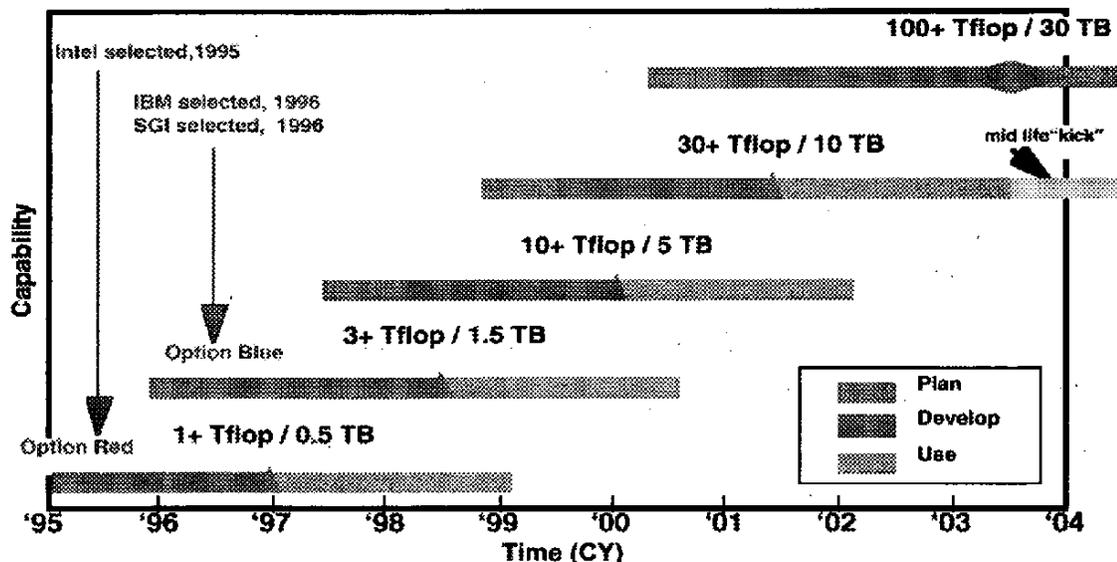
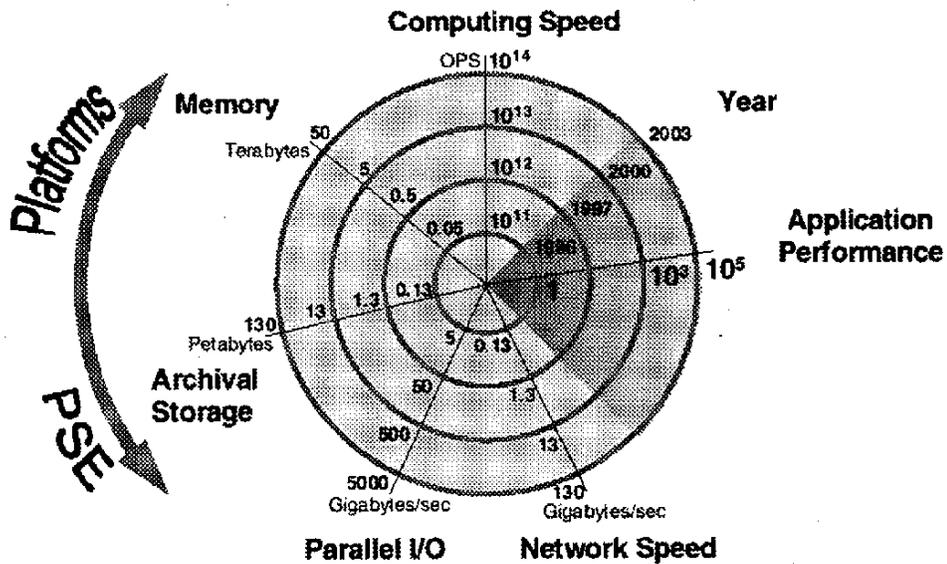


図2.8 ASCIロードマップ

<http://www.llnl.gov/ascii-pathforward/art/final01.gif>

図2.8の10+Tflops/5TBは、後述のLawrence Livermore 国立研究所のASCI Whiteであると考えられる。また、Los Alamos国立研究所の記述によるとPathForwardに加え、Option 4という記述が存在するが、具体的に何を示すのか不明である (<http://www.lanl.gov/projects/ascii/Platforms.html>)。

ASCIプログラムを構成する各コンポーネントの達成予定ロードマップが公表されており、図2.9のように整理されている。



PSE: Problem Solving Environment

図2.9 ASCIコンポーネント技術開発道程

<http://www.llnl.gov/asci/images/target.gif>

ASCIプログラムの研究主体は、DOEの核兵器研究所として認識されている3国立研究所、すなわち、Sandia 国立研究所、Lawrence Livermore国立研究所、Los Alamos 国立研究所である。Sandia 研究所は、Lockheed-Martinによって運営され、Lawrence Livermore、Los Alamos 両研究所は、California大学によって運営されている。

1997年度のASCI要求予算額は、全体で\$121.6Mであり、その内訳を次に示す。

- ① アプリケーション     \$54.9M
- ② 問題解決環境整備     \$23.5M
- ③ プラットフォーム     \$33.7M
- ④ 戦略的提携             \$6.1M
- ⑤ 三研究所共通         \$3.4M

#### 2.4.2 プラットフォーム開発

図2.8の Option Red、Option Blueは、上記3研究所を中心に開発が行われ、そこで用いられる超並列スーパーコンピュータは、米国の3コンピュータメーカーに開発を含む調達が行われる。

行われた。表2.5にその内訳を示す。

表2.5 現在開発中のプラットフォーム

国立研究所名	Sandia 国立研究所	Lawrence Livermore 国 立研究所	Los Alamos 国立研究所
計算機名	ASCI Red	ASCI Blue Pacific	ASCI Blue Mountain
メーカー	インテル	IBM	SGI
使用MPUまたは 製品名	PentiumPro (200MHz)	RS/6000SP	Origin2000 MIPS R10000(195MHz)
調達予算	\$55M (当初\$46M)	\$93M*	\$110.5M
目標性能	1.8Tflops メモリ585GB ディスク容量40TB	3.1Tflops(1999年度) メモリ2.5TB ディスク容量75TB	3.1Tflops(1999年度) メモリ1.5TB ディスク容量75TB
実績	1.34Tflops (1997年秋)	不明	1024PEでアプリ動作 (1998年)

\*1996年度投資：\$27M (1996年度全ASCI投資 \$85M)

### 2.4.3 ソフトウェア開発

ASCIでは、HECCの項でも述べたが、大学を巻き込んだソフトウェア開発に、CIC関連の投資が行われている。すなわち、HECCのDOE担当分、Strategic Alliance Centerが運営され、大学とのAllianceが成立している。

- ① レベルⅠ：アプリケーションを中心に 97年会計年度スタート
  - ・参加大学：Illinois大、Caltech、Utah大、Stanford大、Chicago大
  - ・予算：初年度\$2M、2～3年間は \$4M～5M/年支出される
- ② レベルⅡ：公募を行う
  - ・基本分野：Data Manipulation、Scalable Parallel Algorithm
  - ・アプリ分野：計算物理、Energetic Material等
  - ・選考基準：Affordability、Capability、Applicability、Feasibility
  - ・スケジュール：1998年1月8日募集締切、98年3月Award発表
  - ・参加資格：大学のみ（レベルⅠの大学は含まれない）
  - ・\$.01M～0.4M/年/テーマ、総額\$3M

一方、最近の報道によると、1998年度、ASCI Redの最終段階前のソフトウェア開発をする場としてPittsburgh Supercomputing Centerが任にあたり、そのためのコンピュータ使

用料として\$4.5MをDOEが支給するとされている。契約は、まず1年と予定されている。Pittsburgh Supercomputing Centerは、Carnegie Mellon大、Pittsburgh大、Westinghouse Electric Corp. で運営されているが、1998年度からNSFからの援助が打ち切られている。

#### 2.4.4 ASCI研究開発理念

先に述べたように、ASCIの達成すべき目標は、核兵器関連技術の研究開発であるが、その中味は、物理、化学、計算機科学など広範かつ一般的な科学分野のハイエンドを構成するものである事から、DOEは、この開発は、画像処理技術による戦場での情報優位、暗号の解読能力の保持にも役立つと主張しており、これにより予算獲得を容易にしようとしていると考えられる。

さらに、ASCIで考えられているようなC-COTS (Commodity-Commercial Off The Shelf) を用いたハイエンド技術の開発によって、高度情報技術製品が市場原理で開発されるよりもより早期に実現できるようになり、これが米国の競争力を維持、高めることに貢献すると主張している (図2.10)。

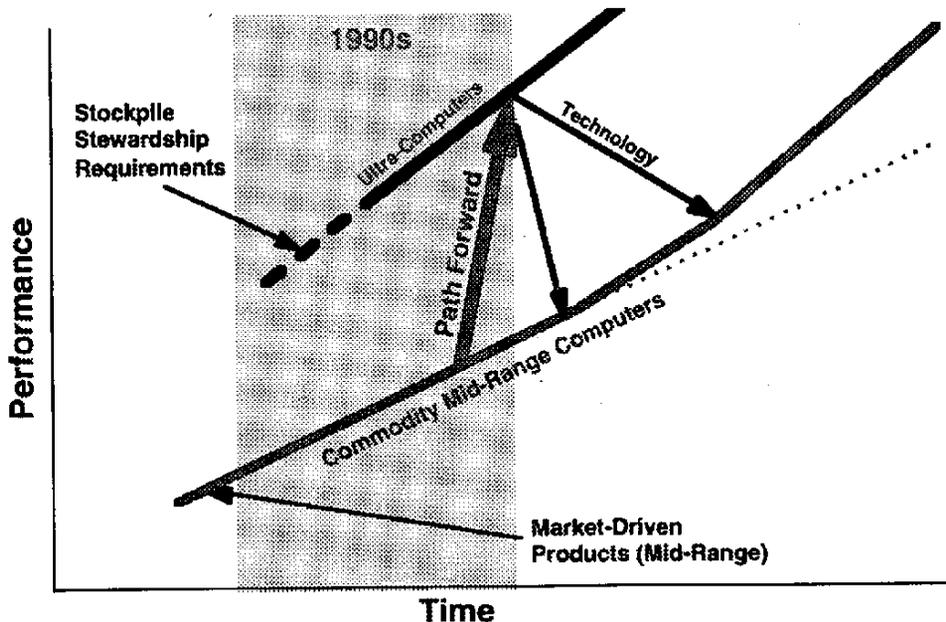


図2.10 ASCIプログラム研究開発理念

<http://www.llnl.gov/asci-pathforward/art/final02.gif>

図2.10は、一部の用語の違いこそあれ、HECCの諮問委員会でのCIC委員会幹部の説明にも用いられており(図2.3)、米国のハイエンドコンピューティング開発の共通概念を良く表しているものと思われる。

ここでその原動力となるものとして提案されており、かつ実行に移されているのが、ASCI PathForward プロジェクトである。

## 2.4.5 ASCI PathForward プロジェクト

### 2.4.5.1 計画概要

ASCI PathForward プロジェクトは、ASCIの10-30-100 Tflopsレベルの研究開発を行うプログラムであり、第一段階として30Tflops実現(2001年)に、\$10M/年の継続的投資が予定されている。目的は、Scientific Stockpile Stewardshipの一環としての核備蓄実験シミュレーションを行うことはもちろんであるが、クリントン大統領が強調するように、最新技術の民間市場への転用も視野に入っている。

### 2.4.5.2 ASCI White

30Tflopsを実現する前段階としてのASCI WhiteがIBMに発注されている。公表内容は、

- ① 発注者：DOE Lawrence Livermore 国立研究所
- ② 受注者：IBM
- ③ 受注金額：\$85M
- ④ 性能：10TeraOps at 2000年 (flopsと記述していない)
- ⑤ 仕様：IBM's 最新、最速の RS/6000 (8,000 processors)  
床面積 17,000 平方フィート(1,579m<sup>2</sup>)  
電力 6.2 MW(含む 冷却、mechanical equipment)  
<http://www.llnl.gov/asci-pathforward/pf-news.html>

であり、Blue Pacificの後継であると考えられる。

### 2.4.5.3 100Tflopsに向けての計画

30-100 Tflopsをねらう技術について、コンピュータメーカーと開発契約が取り交わされており、これは1998年2月3日に公表された。DOE 3国立研究所とIBM、DEC、Sun、SGIの4社が開発を担当し、4社は開発した知的所有権をすぐに市場に売り込むことができる契約であるとされている。開発目標は、2004年までに100Tflopsを実現することで、4年間で\$50Mの予算がついたと報道されている。以下に、ASCI ホームページに公表されている内容に沿って計画を詳述する。

#### (1) 開発技術

PathForwardは、大規模コンピューティングシステムに用いる相互接続技術、データストレージ技術、システムソフトウェア及びツールの開発に注力する。これらの技術は、民間での開発が行われていないが、30Tflopsを達成する上で、最もクリティカルな技術であり、企業が将来の製品や市場に有益なものであると主張されている。開発技術は、本質的にスケーリングとインテグレーションの技術であり、コモディティの部品よりなるウルトラスケールのコンピュータシステムを実現する技術である。

#### (2) 研究調達受注メーカー

2001年までに30Tflopsを達成するプログラムのため、次の4企業が選定され、総額\$50Mが投資される。ここでは、1万個のコモディティプロセッサを相互接続し、30 Tflopsから100 Tflops(2004年)の性能を達成する技術を開発する。

- ① DEC：UNIXベースのAlphaServerをSMP接続した256ノードをつなぐプロセッサ相互接続技術の開発とデモンストレーション。
- ② IBM：100Tflopsまでのシステムに使われる、高速かつ低レイテンシでスケラブルなスイッチング技術の開発。
- ③ SGI/CRAY：将来のルータ、スイッチ、通信ライン、チャネル及び相互接続に用いられるであろう先進的な信号処理と相互接続技術の開発と評価。
- ④ SUN：相互接続の構造を確立し、相互接続モニタ、リソース管理、MPI能力のスケラビリティと正当性を実証することにより、ハードウェアとソフトウェアの現実性を検証する。

### (3) 省庁連携によるコンポーネント開発

ASCIが必要とするシステム開発では、Stewardship Programの複雑なシミュレーションを行うため、巨大データ蓄積システムの物理サイズ縮小、データ・情報の書き込み速度の向上が必要で、このため他省庁との連携を行う。この省庁間連携はDOEとNASA、及び DODの間で行われる。ここで開発される技術は、オプティカルテープドライブ技術で、書き込み速度 25 MB/s、蓄積サイズ 1 Terabyteの性能を持ち、通常の大ささを持つオプティカルテープカートリッジを使用するというものである。

### (4) パートナシップによるソフトウェア開発

PathForwardによってソフトウェア開発でも民間企業とのパートナーシップを確立し、国家安全保障を考える上で必要な判断材料を提供するために、ツール開発のより完全な政府・産業共同研究支援を行おうとしている。ここでは、計算ベースの核抑止力の検証と確認を行うシミュレーションソフトウェアコードの開発を行う予定である。

また、次のようなシステムソフトウェアとツールの分野の研究が必要とされている。

- ・ Stockpile simulationsに用いるアプリケーションコードの開発
- ・ 分散並列OSと並列スケラブルプログラミング環境の開発

最近、DOEに属する機関(Argonne 国立研究所、Los Alamos 国立研究所、Lawrence Berkeley 国立研究所、Lawrence Livermore 国立研究所、Oak Ridge 国立研究所、Sandia 国立研究所、DOE Offices of Defense Program and Energy Research)で上記ソフトウェア領域で現在存在するもしくは将来生ずる必要開発項目の初期的洗い出しと検証およびプライオリティづけに関する共同研究が行われているという記述があり、これは結果として他の省庁・学界・企業を巻き込んだ次期共同研究につながると考えられている。

## 2.5 むすび

米国の高度コンピューティング研究開発は、常にハイエンドなフロンティア領域をターゲットにおき、世界に先んじることを国是とし、理念を明確にすることによって、実効のある政策を次々に実行しているように見える。これは、この領域での研究開発が軍事などの国家安全保障に密接に関連し、かつ、未踏領域の拡大によって利益を享受してきた米国の歴史と文化に裏打ちされた国民の合意があるからであると考えられる。

現在の計画・実行状況から、2007年～2010年にかけて1Pflopsの性能を有するウルトラコンピュータが誕生すると考えられ、これは、1万～100万の間の並列度を有する超並列型

コンピュータであることは間違いないであろう。このコンピュータの開発過程で得られた分散・並列技術のハードウェア・ソフトウェアに関するノウハウは、民間企業の製品開発に生かされ、情報技術の広い範囲で米国の優位を動かぬものとすると考えられる。

一方、日本の高度コンピューティング開発においては、分散並列コンピューティングを研究するRWCP、2001年度 32Tflopsをねらう「地球シミュレータ」があるが、いずれもペタフロップスレベルの開発計画への連続性を見出すことができない。日本でも2000年、200Tflopsをねらい、近い将来1Petaflopsを実現するであろう超並列専用計算機「Grape-6」があるが、技術的波及効果において米国の幅広い研究と比較すると規模の小ささは否めないと考えられる。

国としての研究戦略として、「ものまねを排する」という理念を持つならば、いかに無駄に見えてもハイエンドなフロンティア領域の研究に投資することを避けてはならないと思われる。

## 第3章 わが国における研究開発の課題と展望

### 3.1 アーキテクチャ

#### 3.1.1 直接解法マシン(Custom Computing Machines):

##### 打倒プログラム格納型計算機への最後の希望

(天野英晴委員)

##### 3.1.1.1 はじめに

高性能コンピューティングを行う最も簡単かつ現実的な方法は、現状では、高性能マイクロプロセッサを多数用いた並列計算方式であろう。この分野は、一部の日本企業、研究者が健闘しており、米国独占が強まる計算機科学の他の分野に比べて、まだしも希望を持つ事ができる。したがって本報告書も中心はこのタイプのアーキテクチャということになるであろう。そこで、この辺の記述は、他の章に譲ることにして、本節ではより将来、7～10年後に、高性能コンピューティングの主流となり得る新しいアーキテクチャである直接解法マシン(Custom Computing Machines)を紹介し、その技術課題、開発状況等をまとめる。

##### 3.1.1.2 直接解法マシン(Custom Computing Machines)とは

PLD(Programmable Logic Device)は、ユーザが手元でプログラミングすることのできるデバイスであり、1970年代から小規模の書き換え不能なものが一部の組合せ回路に用いられていた。80年代にCMOS技術の導入により、再書き込み可能で、順序回路も含む広い範囲の回路を実装することができるようになり、デジタル回路の実装用デバイスとしての地位を確立した。さらに、90年代に入って、大規模なFPGA(Field Programmable Gate Array)あるいはCPLD(Complex Programmable Logic Device)が登場するに至って、システムをまるごと実装可能なデバイスとして進化を遂げた。最近は、100MHz近い周波数で動作する高速なデバイスや、数10万ゲートの実装密度や内蔵RAMを持つものも出現し、ASICに代わってシステムインテグレーションの主役になりつつある。このうち、内部の配線データ(Configuration Data)をSRAMに格納するタイプは、配線データを入れ換えることにより、電源を入れたままで、ハードウェア構成を変えることができる。この性質を利用し、状況に応じてハードウェア構成を変化させたり、解法アルゴリズムを直接ハードウェア化させることのできるシステムを可変構造システム(Reconfigurable Machine)あるいは直接

解法マシン(Custom Computing Machine : CCM)と呼ぶ。

CCMが実現可能になったのは、数千ゲート以上の規模で10MHz以上の動作が可能なSRAM型FPGAが安価に利用可能になった1990年代に入ってからである。実装例の多くは、専用システムの構造に柔軟性を持たせるタイプであるが、このうちのいくつかは、広い範囲に利用することができる汎用システムを目指している。

海外の代表的なシステムを表1に、国内のシステムを表2にまとめる。ここでは、代表として、米国の計算機科学センター(IBM SRC)のSplashプロジェクト[1][2]のみを紹介する。他のCCMについては、最新のサーベイが[3][4]に掲載されているので、参照されたい。

表1 海外のシステム

システム	FPGA	構成の特徴	応用分野
Splash	XC3090×32	リニアアレイ	汎用エンジン (DNA塩基配列マッチング etc)
Splash 2	XC4010×16	リニアアレイおよびクロスバ	汎用エンジン (文字列検索、画像処理 etc)
PRISM-I	XC3090	プロセッサと結合	汎用コプロセッサ (ハミング距離 etc)
PRISM-II	XC4010×3	プロセッサと結合	汎用コプロセッサ
ArMen	XC3090×4	トランスピュータとキューブ結合	MIMDマシン (画像処理、気体シミュレーション etc)
QacChip	カスタム	RISCパイプライン中に埋め込み	組み込み用プロセッサ (画像処理 etc)
DISC-II	CLAy31×3	カスタム命令モジュールを動的書き換え	汎用プロセッサ (画像処理 etc)

表2 国内のシステム

名前	場所	FPGA	構造	Main target
RM-I	Kobe U.	XC3090×4	Shared bus	Logic/Fault simulation
RM-II	Kobe U.	XC4005×9	Cross bus	Logic/Fault simulation
RM-III	Kobe U.	XC4005×6	FPIC	Logic/Fault simulation
RM-IV	Kobe U.	XC4006×16	FPIC	Logic/Fault simulation
FLEMING	Keio U.	XC5215×6	Circular network	Virtual Hardware emulation Queuing/Markov analysis
SOP	NEC	XC4010	Aptix board	General purpose (Data flow control)
OPERL	Nagoya U.	OR2C40	PCI bus board	General purpose
Cameroon	Execute system	XC3030	PC98 bus board	Function test

SplashプロジェクトではFPGAとメモリからなる計算セルを直線状に並べることにより、汎用計算機と専用計算機の間中間的な特徴を持つ可変構造システムの研究開発に取り組んでいる。図1に Splash 2 の構成を示す。この図は 3 ボードからなるシステムを示しているが、各ボード上では Xilinx 社の FPGA である XC4010 が計算用に 16 個(X1~X16)、制御用に 1 個(X0)使用され、これらがクロスバを介して結合されている。また、各 FPGA は 256K ×16 ビットのローカルメモリを持っている。計算用の 16 個の FPGA はリニアアストリックアレイを構成しており、その結合の幅は 36 ビットである。X0 の FPGA はクロスバの制御にも用いられる。

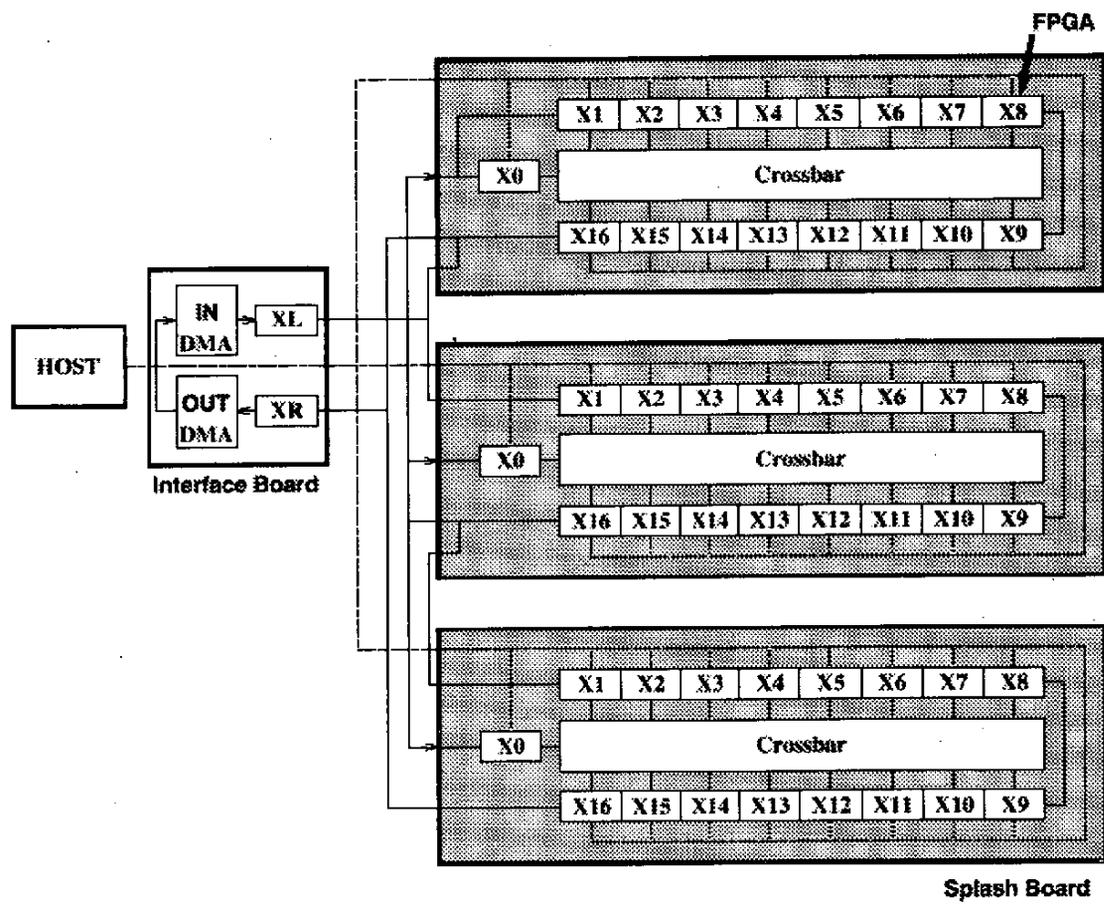


図1 Splash ボード

Splash ボードは、インタフェースボード上のそれぞれ 2 つの入力 FIFO(First-In First-Out)と出力 FIFO を介して、ホストの SUN Sparc Station II と通信することができる。入力 FIFO は 36 ビットのデータを X1 に送り、出力 FIFO は 32 ビットのデータを X16 から受け取る。図1からも明らかなように、入力 FIFO の内容を X0 に与えることにより、入力 FIFO からのバスは SIMD (Single Instruction stream and Multiple Data stream)

バスとしても機能するようになっている。その場合、クロスバは命令をブロードキャストする一方で、データを X1 から X16 に分配する。このようなクロスバの制御は X0 が行う。また、各チップからは 1 本の割り込み制御線が、それぞれホストと接続されている。

Splash 2 のプログラムはハードウェア記述言語である VHDL で記述される。その際、VHDL の動作レベル記述と構造レベル記述を混在させることができる。VHDL でプログラムを記述した後は、Splash 2 シミュレータを用いて機能検証とデバッグを容易に行うことができる。したがって、記述したアルゴリズムが機能的に正しいことを確認するまでは、論理合成などの比較的時間のかかる処理を行う必要がない。

また、論理合成された回路を実際にシステム上で動作させる段階でのデバッグ機能も充実しており、シングルステップ実行機能や各 FPGA 内部のフリップフロップを読み出すリードバック機能などを備えている。特に、リードバック機能は、デバッグ時だけでなく内部レジスタに格納された計算結果の読み出しにも利用することが可能である。

Splash プロジェクトでは、幅広い分野のアプリケーションが実装され、多くの成果を挙げている。Splash 2 の前身である Splash 1 では DNA の塩基配列間の距離を求める計算で、スーパーコンピュータ CRAY-2 の 330 倍の性能を得て大きな注目を集めた。また、Splash 2 でも文字列検索や画像処理などのアプリケーションが実装されている。このように Splash では、そのリニアアレイ構造を活かせるパターンマッチングや画像処理など、常に入力されてくるデータをパイプライン的に処理するようなアプリケーションでは大幅な性能向上を示している。一方で、FPGA 間の結合の自由度が低い点や任意のメモリへのアクセスが困難な点などから、有効性のある適用分野もある程度限定されている。

### 3.1.1.3 CCM の問題点

CCM の問題点としては以下の 2 点が従来から指摘されている。

- ① 問題アルゴリズムを高いレベルで記述し、それを質の良いハードウェアに変換するシステムティックな方法が存在しない。Splash のようにいちいち VHDL で記述するのではプログラマがたまらない。
- ② システムのサイズを越える規模の問題が解けない。これはちょうど仮想記憶のない計算機のようなものである。

この問題点の解決については、筆者らの提案する方法を含め、様々な方法が提案されて

おり、見直しはある程度ついている。前者の問題については問題をデータフロー言語や拡張C言語で記述し、データフローグラフを経てハードウェアに変換する方法は既に確立している。後者の問題についてはマルチファンクションあるいはマルチコンテキスト機能を持つFPGAやDRAM混載型FPGAによる仮想ハードウェアの実現方法が検討されている。いずれも困難な点は残されているが、道筋はついていると思う。論文[5][6]などを参照されたい。

ここでは、対象をスーパーコンピューティングに絞って、より本質的な問題点について議論する。CCMは確かに特定の分野では、ワークステーションや時にはスーパーコンピュータを上回る性能を実現する。前述のSplashは、DNAの塩基配列間の距離を求める問題、神戸大学のRM-Xでは、論理シミュレーション、故障シミュレーション、我々の開発したFLEMINGでは、マルコフ解析、待ち行列解析で、ワークステーションやスーパーコンピュータの数倍から100倍近い性能を発揮している[3][4]。しかし、これらの問題は、以下の二つの性質のいずれをも持っていないものに限られている。

- ① 高精度の浮動小数点を扱う数値計算を行わない。
- ② 巨大な記憶容量を必要としない。

(1) 10倍×10倍のハンデ

IEEE標準の浮動小数点演算を行う場合、FPGAと、高性能プロセッサに用いられる専用の浮動小数点演算用LSIでは、動作速度で10倍、実装面積で10倍の差があるといわれている。これは以下の原因に基づいている。

- (a) FPGAの実装プロセスは、最新の専用LSIのプロセスより一時代遅れている。
- (b) 専用の浮動小数点演算用LSIでは、レイアウトレベルで既に最適化されたマクロセルを用いることができる。これに対してFPGAは論理回路をSRAMのテーブルで構成したり、配線をトランスファークラップを用いて切り替えたりするため、遅延時間、実装面積共に不利である。

このうち、(a)に関しては、ごく最近になってようやくFPGAに最新のプロセスが用いられるようになり、改善されつつある。しかし(b)の問題点はFPGAが柔軟性と引き替えにした本質的なものである。したがって、将来にわたってもFPGAによる浮動小数点演算器は専用LSIに比べ速度、面積の両面で5倍以上のハンデを背負うことになるだろう。このハンデを克服するためには以下の方法が考えられる。

(a) アプリケーションにあったデータ幅を用いる。

現在の IEEE 標準の 32bit/64bit 浮動小数点データ型は多くのアプリケーションをカバーするが、アプリケーションによっては、長すぎたり、短すぎたりする。対象アプリケーションに適合するデータ幅が標準と大きく異なる場合、CCM はこれに合わせたデータ幅を利用することにより、ハンデを大きく縮めることができる。場合によっては、1回の演算速度で優位に立つことができる。ニューラルネットワークシミュレータや、一部の信号処理で、このような利点を生かして高速化した例が報告されている。ただし、この方法を用いるためには、様々なデータ幅で高い効率を実現する演算器のマクロが整備されていることが必要である。

(b) 並列演算、他の演算との組み合わせ、データの流れの最適化を図る。

通常の浮動小数点演算器も多数接続して並列処理を行うことができるが、CCM の場合は、接続を柔軟に変化させ、データのやり取りを含めて最適化することができる。また、アプリケーションによっては、ビット演算やパターンマッチング等 FPGA の得意な演算と組み合わせることも可能である。ただし、この利点を発揮させるためには1チップ上に複数の演算器を実装できるようになることが前提である。

この両者の組み合わせがうまく行けば、CCM は同一コストで構成された高性能演算プロセッサによる並列計算機と比べても高い性能を実現できるだろう。まともに IEEE 標準データで戦っては将来に渡っても勝ち目がないことを認識すべきである。

(2) メモリとの接続はチップ内部で。

CCM がメモリを大量に用いる処理に弱いのは、メモリとの接続が通常の計算機に比べ貧弱なためである。なにしろ通常のプログラム格納型計算機は CPU とメモリを結ぶラインが命であるため、最大限に高速化してある上、キャッシュもついているし、容量に関しては仮想記憶によってディスクを借りることだって可能である。CCM もそれぞれの構成要素の FPGA にメモリを接続した構成を取る場合が多いが、容量、速度共にプログラム格納型計算機には遠く及ばない。

この点を解決するためには、DRAM 混載型 FPGA が不可欠である。DRAM 混載型 FPGA は、チップ内に複数の回路を持つ方式（マルチコンテキスト）への利用も期待されるが、単純にデータの記憶という点でも重要である。内部に複数の DRAM モジュールが装備され、演算器と柔軟に接続することができれば、メモリと演算器の接続に柔軟性を欠くプログラム格納型計算機に比べて DRAM の転送容量をうまく利用することができる可能性もある。

### 3.1.1.4 CCMスーパーコンピュータ実現へのシナリオ

以上、CCM型のスーパーコンピューティングシステムが現実となるのは、

- (a) 1チップ上に複数の浮動小数点演算装置を格納できる規模のFPGA(100万ゲートほどか?)が登場し、
- (b) これにDRAMが混載されており、
- (c) さらに、様々なデータ幅の浮動小数点マクロが整備されている必要がある。

もちろん、CCM全体の問題であるプログラミングの問題と仮想ハードウェアの問題もクリアされていることが望ましい。これらの条件のうち、(a)は放っておいても2、3年後にはクリアされるだろう。他の条件も努力次第で5年以内に整備することが可能であり、3年後くらいにこれを見越して大型プロジェクトをスタートさせるのがよいと思う。日本が大型プロジェクトとして主にやるべきことは(b)である。DRAM混載型FPGAは既にNECにより提案されており[7]、このチップの性能はMITで開発中のDISC-II[8]をはるかに上回っている。しかし、現状では商業ベースで採算が取れる見込みがないため、実際に製造することができないでいる。FPGAの実装技術は日本は遅れているが、DRAM混載チップについては世界に先んじているので、この周辺の技術開発については政府のプロジェクトでもっと後押ししてもよいと思う。(c)については大学の共同プロジェクトで整備する計画があり、数年後には実現する予定である。

さらに、日本企業は、日本で生まれた技術をもう少し積極的に利用してもいいのではないか。例えばXilinx社が昨年提案し、特許を取得したTime Multiplexed FPGA[9]は、SRAMモジュールをFPGA内部に複数持つことにより、高速に内部回路を切り替える方式であるが、1990年に富士通が特許を取得したMPLD[10] (Multi-functional Programmable Logic Device) とまったく同じアイデアである。この方式に関しては10年近く先んじていたのである。MPLDの特許は国内のものだが、この方式は我々がWASMIIプロジェクトで利用していることから、特許の存在を明記した英語の論文を多数書いており(例えば[5])、Xilinx社も当然認識していたと思う。

とはいえ何より問題なのは、特許を取ってもまったくチップを実現できず、結局Xilinx社による商品化を待つしかなかったことで、大変に残念である。今後も、仮想ハードウェアや適応型ハードウェア等の分野で、日本ではるかに先んじて提案されている技術を、米国の企業が(多分無断で)実現していく状況が生じてくることが予想される。もったいない話である。

参考文献

- [1] T. Miyazaki: "Reconfigurable Systems: A Survey," Proc. of ASP-DAC'98 pp. 447-452, 1998.
- [2] H. Amano, Y. Shibata: "Reconfigurable Systems: Activities in Asia and South Pacific," Proc. of ASP-DAC'98 pp. 453-458, 1998.
- [3] M. Gokhale, W. Holmes, A. Kopser, S. Lucas and D. Lopresti: "Building and Using a Highly Parallel Programmable Logic Array" IEEE Computer, Vol. 24, No. 1, pp. 88-89, 1991.
- [4] J. M. Arnold, D. A. Buell, E. G. Davis: "SPLASH2" Proceedings of the 4th ACM Symposium on Parallel Algorithms and Architectures, pp. 316-322, 1992.
- [5] X.-P. Ling and H. Amano: "WASMII: a Data Driven Computer on a Virtual Hardware" Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, pp. 33-42, 1993.
- [6] T. Yamauchi, S. Nakaya, N. Kajihara: "SOP: A Reconfigurable Massively Parallel System and Its Control-Data-Flow based Compiling Method," Proc. of IEEE Symposium on FPGAs for Custom Computing Machines, pp. 148-156, 1998.
- [7] M. Motomura, Y. Aimoto, A. Shibayama, Y. Yabe and M. Yamashina: "An Embedded DRAM-FPGA Chip with Instantaneous Logic Reconfiguration" Proceedings of the Symposium on VLSI Circuits, 8-1, 1997.
- [8] M. J. Wirthlin and B. L. Hutchings: "Sequencing Run-Time Reconfigured Hardware with Software" Proceedings of ACM/SIGDA International Symposium on Field Programmable Gate Arrays, pp. 122-128, 1996
- [9] S. Trimberger, D. Carberry, A. Johnson and J. Wong: "A Time-Multiplexed FPGA" Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, pp. 22-28, 1997.
- [10] 吉見昌久 (富士通株式会社): "マルチファンクションプログラマブルロジックデバイス" 公開特許公報(A), 平 2-130023, 1990.

### 3.1.2 ペタフロップスマシンに実性能を伴わせるためには

(久門耕一委員)

#### 3.1.2.1 始めに

ペタフロップスマシンを検討する際に、とかく peak パフォーマンスに目を奪われがちになる。しかし、単純にプロセッサの性能と数を掛け合わせた値には意味がなく、必要とされる性能はアプリケーションプログラムを実行した場合の実効 flops である。ペタフロップスマシンは大規模な並列計算機で構成せざるを得ず、peak パフォーマンスと実効パフォーマンスの乖離が顕著になると考えられる。本報告ではこのような性能の乖離を引き起こす要因となる、ネットワークに関して検討を行う。

#### 3.1.2.2 ペタフロップスマシン実現に必要なネットワーク

ペタフロップスマシンの実現性の検討を行う場合、単純にプロセッサの性能とプロセッサ数を掛け合わせた値は意味がない。当然、目的とするアプリケーションを実行した場合の実効 flops をどのように向上させるかが重要である。数値計算向けのシステムを構成する要素の中の主だったものとして、

- CPU
- メモリ
- I/O
- プロセッサ間接続網

がある。このうち、peak flops 性能に影響を及ぼすのは CPU とメモリのみで、他は flops 性能には直接反映されない。CPU とメモリに対して既存のハードウェアがどのようなバランスを持っているのかを見るため、1CPU あたり、ほぼ 2Gflops の演算性能を持つ富士通の VPP700 と、日本電気の SX-4 の諸元を以下に示す。スループットは  $1W = 8 \text{ Byte} = 1 \text{ floating word}$  として表した。

表1 ベクトル型コンピュータの諸元

システム名	ベクトル性能 /CPU	メモリスループット
VPP700	2.2Gflops	2.28 GW/s
SX-4	2 Gflops	2 GW/s
T90	1.8 Gflops	1.6 GW/s

このように、ベクトルプロセッサのベクトル性能とメモリスループットはほぼ1対1に設計される。それに対して、疎結合部分を持つVPP700のノード間スループットは0.14 GW/sとなっておりCPU間結合網のバンド幅は1桁以上小さいことになる。

プロセッサ数が十分に少ない場合には各プロセッサが各自のメモリとの間で処理を行い、別プロセッサとの間でデータの交換頻度が少ない計算方式を取る場合にはこのようなスループット配分で構成が出来る。

しかし、ベタフロップスマシンを構成するために必要なプロセッサ数は、次の表のように10万プロセッサ程度に達すると考えられる。プロセッサ数が極端に多くなると、メモリバンド幅に比べ、プロセッサ間の通信を大幅に減らすことは困難であろう。そのため、メモリ対CPU間スループットの比率は、より1対1に近付けなければならない。

表2 1Pflops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

このことを、比較的素性が良いと思われるFFT演算で考えてみる。FFT演算を1プロセッサで実行する場合、一般に $2^n$ 点のFFT演算はオーダとして $n(2^n)$ 回のバタフライ演算が必要で、メモリへのアクセスも同じオーダ必要となる。一方、2プロセッサで実現する場合には、1プロセッサあたり $n(2^n)/2$ 回のバタフライ演算と、同数のメモリアクセス、 $2^n$ 回のプロセッサ間通信が必要になる。これを表にまとめると次のようになり、通信量はメモリアクセスとオーダ的には同程度必要になることが分かる。

表3 FFT 演算に必要なメモリアクセス回数、通信回数

CPU 台数	演算回数	メモリアクセス回数	通信回数	通信/メモリ比
1	$n(2^n)$	$n2^n$	0	0
2	$n2^{n-1}$	$n2^{n-1}$	$2^n$	$2/n$
$2^m$	$n2^{n-m}$	$n2^{n-m}$	$2m \cdot 2^{n-m}$	$2m/n$

このように、分割数を多くすると通信量が増加することは、全体の計算をメッシュ上に切られたプロセッサに割り当てた時、メッシュの1目の体積が計算量に、1目の表面積が通信量に比例すると考えても説明がつく。

従来のベクトルプロセッサでは、接続ビット幅をかなり広くとることによりCPUメモリ間のバンド幅を広げて来たが、大規模な並列計算機のCPU間接続網をビット幅が広いネットワークで作成することは物理的に不可能である。したがって、高いスループットを達成するために、伝送路1本あたりの転送bit rateを高くすることが必要となる。

現在商品化あるいは商品に近い形で実現されている相互接続網技術でもっとも高速なものは10Gbit/s程度のものである。これは、ワード換算で0.2GW/sの転送レートをもつ相互接続網とすることができ、線あたり400Mbit/sのデータ転送レートで4B並列伝送することにより実現されている[3]。

ネットワークバンド幅をメモリバンド幅と同程度すなわち、flops値と同程度にするためには、10Gflopsのマシンには、10GW/sのメモリバンド幅と10GW/sのスループットを持つ相互結合網が必要になる。このためには、現在のネットワークスループットを更に1桁以上向上させる必要がある。現時点では、電気伝送で数メートルの距離を信号あたり4-10Gbpsぐらいの転送レートで伝送することは実現可能な範囲であると考えられている[1][2]。

### 3.1.2.3 不均等なリモートアクセス時間

このように、ネットワークスループットを向上させた場合でも、ネットワークのレイテンシを減らすことは出来ない。10万プロセッサを3次元に構成すると考え、 $50 \times 50 \times 50$ の3次元メッシュ構造に並べると考える。

単体のプロセッサの大きさを冷却のための装置、空間を含めて20cm立方とすると、縦横高さ10mの大きさになる。この場合最も遠い対角線の距離は直線距離で17mになる。ここを20cm/nsのケーブルで直線で配線すると85nsの遅延が電線上の伝送遅延として必要である。想定されるシステムの要素プロセッサの速度を10Gflopsと仮定すると、最大遅

延時間は電線のみで 1,000 flop 程度の演算に相当する時間である。物理的に近接しているノードでは、必要な遅延時間はもっと小さくすることが出来る。ノード毎に考慮すべき点を少なくするために、すべてのノード間の通信を最大遅延まで遅くすることも出来る。しかし、全体が遅くなってしまふのでノード間通信は不均等にせざるを得ないだろう。例えば、図1に示すように、ネットワークとしてはメッシュネットワークが物理的実装に一致するため、隣接ノードを高速に結ぶ実装が容易である。

高速伝送路の技術として、光結合が検討されることが多いが、光通信は短距離伝送においては必ずしも適切な通信方式とは言えない。これは、現状の光通信における一般的技術では、電気的処理を施す際に直並列変換を施してデータ幅を増加させ bit rate を下げている。低速クロックの部分で多くのバッファリングが行われる結果、point to point の遅延時間が大きくなってしまふ。また、次に述べるように LSI の内部速度に比べて LSI の IO 速度が小さいと言う点は、光結合を持ってきても解決できない。

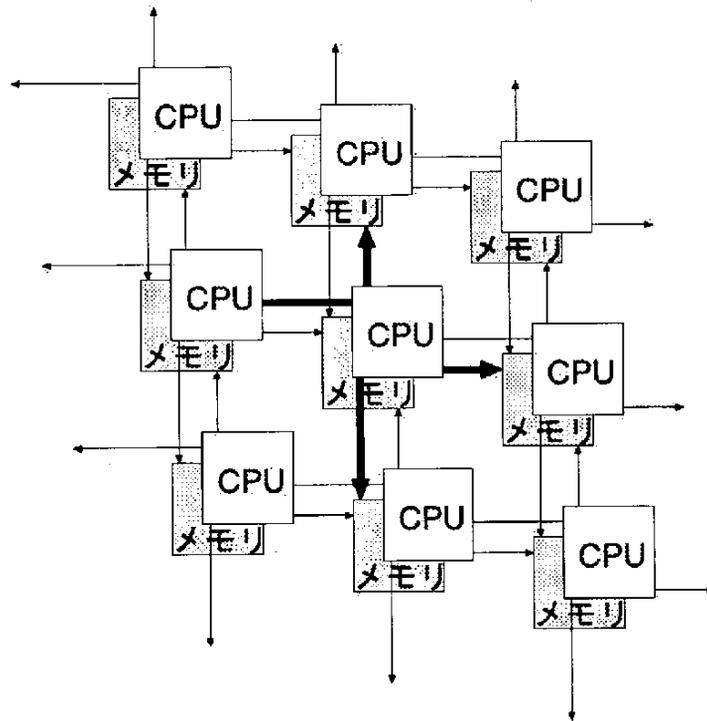


図1 二次元メッシュ部分共有メモリ

### 3.1.2.4 今後の技術開発のポイント

名目上のペタフロップスマシンを作成する場合はともかく、実質的に意味を持つマシン開発のために、最も重要なのは、要素プロセッサよりも結合網と並列計算に適応したアプリケーションプログラムである。ベクトル用要素プロセッサとしては、2Gflops の要素プロセッサが構成されている(1997年の時点で)が、これを10Gflops にスケールアップすることは簡単ではない。

表4にSIAの予測とそれから計算されるチップ内のTr数とチップ内クロック積、及びIOピン数とチップ外に出せるクロック積を示した[4]。

表4 High performance ASIC に利用可能な諸元

製品出荷時期	1999	2001	2003	2006	2009	2012
On-chip clock (MHz)	1,250	1,500	2,100	3,500	6,000	10,000
Logic Tr/cm <sup>2</sup>	14M	16M	24M	40M	64M	100M
clock * tr	18G	24G	50G	140G	380G	1000G
ASIC Chip I/Os	2000	2400	3000	4000	5400	7300
Board clock (MHz)	1,200	1,400	1,600	2,000	2,500	3,000
clock * IO	2.4M	3.3M	4.8M	8M	14M	22M

この表から明らかなように、Clock・Tr積は1999年から2012年に掛けて50倍以上に増加するのに対して、Clock・IO積は9.2倍にしか増加しないと予想されている。チップ面積の増加を考慮すると、Clock・Tr積は1999年からの13年間に100倍に増加するということになる。

結局、演算器数は、チップ内のトランジスタの増加により増やすことが出来るが、CPUに供給するデータの出入口であるピン数はそれほど増加しないことを示している。更に、DRAMのランダムアクセススループット、すなわちアクセス速度がチップ内速度ほどは向上しないことを合わせると、メモリからのバンド幅はチップ内の論理回路の速度向上ほどは増えないということも分かる。

これらを多少とも改善するアーキテクチャ面での工夫はあると思うが、実アプリケーションでの性能を左右するマシンの基礎体力は、高速ロジック、高速伝送技術などのベースとなる技術の性能に強く依存する。特に、通信分野ではあまり重要視されてこなかったCMOSでの電気伝送による高速通信技術は、短レイテンシを要求するシステム内接続網に最も重要な技術である。この技術の開発は、高並列システムには必須の技術であるが、学

会発表を見る限り米国の方が進んでいるため、この方面の開発促進は重要な課題となる。

### 3.1.2.5 国としてとり組むべき課題

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いので、ペタフロップスマシンの実現のためには、国が産業界を牽引する形で実現する必要がある。現時点でペタフロップスマシンの意義は、巨大加速器と同じように科学技術の進展のためと言う側面が大きい。ペタフロップスマシンを構成するいくつかの技術の中で、他への波及効果が期待できる基礎技術が含まれている。ペタフロップスマシンはこのような様々な技術への影響を及ぼす基礎的技術開発を推進するという意味の指標として取り組むことが良いと考える。

### 参考文献

- [1] Hot interconnects V symposium record, 1997, IEEE
- [2] Hot interconnects IV symposium record, 1996, IEEE
- [3] The Synfinity Interconnect Architecture, <http://www.fjst.com/synfin/Hot>
- [4] Table of Overall Roadmap Technology Characteristics, 1998, SIA,  
<http://notes.sematech.org/mcpgs/ovrall4.pdf>

### 3.1.3 高精度予測と高性能計算 (中島 浩委員)

#### 3.1.3.1 はじめに

本節では、将来の高性能計算を支える重要な要素技術として、計算とそれに付随する様々な事象を精密に予測する技術を取り上げる。すなわち、本質的に不可知あるいは可知ではあるが知ることが困難な情報について何らかの予測を行い、その結果に基づいて計算の高速化を図る技術について、以下のような議論を行う。

##### (1) 予測技術の重要性

高性能計算、特に並列計算にとって、高精度の予測とその結果に基づく計算の最適化がなぜ重要かを議論する。

##### (2) 予測技術の実際

現状の予測技術を様々な観点から分類し、並列計算を含む高性能計算のためにどのように用いられているかを議論する。

##### (3) 予測技術の将来

将来の予測技術の方向について、プロセッサやシステムのアーキテクチャに関する大局的見地から議論する。

#### 3.1.3.2 予測技術の重要性

並列計算は一般に、分散と統合という2種類の相反する性質の処理から構成される。このうち分散的（あるいは局所的）処理には計算の本質的部分が多く含まれるのに対し、統合的（あるいは大域的）処理には非本質的操作が少なからず含まれる。

たとえばプロセッサ  $P_x$  がある値  $X$  を、また  $P_y$  が  $Y$  をそれぞれ独立に計算し、その和  $Z$  を  $P_x$  が計算するという、ごく単純な例について考える。これらの計算の中で、分散的に行われる  $X$  や  $Y$  の計算は当然本質的である。一方  $P_x$  が行う統合的処理である  $Z=X+Y$  については、加算自身はもちろん、 $Y$  の値を  $P_y$  から得る操作もやはり本質的である。そこでこれらの本質的な操作に要するコストを最小化するために、 $Y$  を  $P_x$  での加算操作に可視であるようにする、すなわち  $Y$  を  $P_x$  のレジスタやメモリに置くようにすると、 $Y$  の正当性を知るために  $P_y$  での計算完了を確認するために同期操作が必要となる。

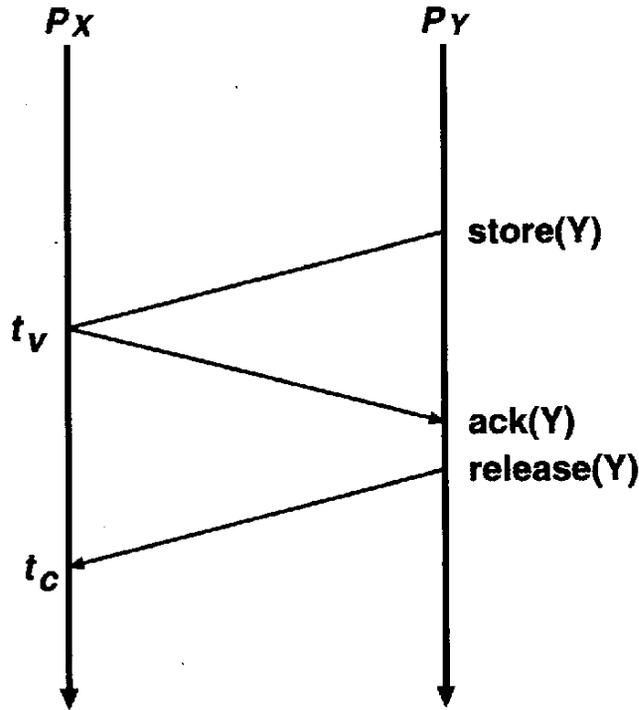


図1  $Z=X+Y$  の並列計算

この同期操作は、 $P_x$ が加算を行おうとする時点で $Y$ が正当な値であれば不要であるということから、非本質的な操作であるといえる。この非本質的な操作のコストを隠蔽するには、 $Y$ がどの時点で（望むらくは $X$ の生成と同時に）正当な値となるかを予測し、その予測に基づき加算を投機的に実行すればよい。

この予測は図1に示すように、 $Y$ が実際に正当な値となる時刻  $t_v$  に行ったとしても、それが確認される時刻  $t_c$  で判明する情報を予測していると考えられる。すなわち空間的に離れた場所に置ける計算状況の認識には、空間的距離を時間に変換したコストを要し、そのコスト隠蔽には本質的に不可知である未来の状況を予測する必要がある。したがって並列計算においては本質的計算コストの削減や、それに伴って生じる非本質的計算コストの隠蔽のために、並列計算がなされる場の空間的距離を時間に変換した未来の状況の予測が重要であるといえる。

また上記の例では、 $X$ の計算が完了すると同時に $Y$ が正当な値となるのが最適である。そのためには $X$ 、 $Y$ の計算に要する時間  $T_x$ 、 $T_y$  と、 $Y$ を  $P_y$ から  $P_x$ に伝達するための時間  $T_c$  が、 $T_F=T_y+T_c$ となる必要がある。あるいはこのような通信時間を加味した微調整は省略す

るとしても、 $T_x \sim T_y$ とすること、すなわち負荷の均衡が重要であることはいうまでもなく、不均衡によって生じるアイドル時間は非本質的計算コストとなる。

ここで  $X$  や  $Y$  の計算が一定の独立性を持った部分計算に分解でき、個々の部分計算を任意に  $P_x$  または  $P_y$  に割付け可能であるとする。このような状況で最も望ましい負荷分散は、計算開始前に個々の部分計算に要する時間（厳密には  $P_x$  および  $P_y$  おける計算時間）をプログラム解析などによって予測し、静的な割付を行って負荷を均衡させることである。一方、静的予測が困難であったり十分な予測精度が得られない場合など、部分計算の動的な割付を必要とする際には

- 動的に生成される部分計算の数
- 個々の部分計算の計算時間
- 他のプロセッサの負荷状況

などを予測する必要がある。逆にこれらの予測を省略する、精度を落す、実際の状況を知る、などの「手抜き」をしてしまうと、直ちにアイドル時間や前述の遠隔計算状況の認識による非本質的計算コストが増加する。したがって、予測自体に要する計算コストを予測精度の向上によって得られる利益でカバーできる範囲で、高精度の予測を静的／動的に行う必要がある。すなわち並列計算においては低コストで得られる情報に基づく一定の局所的計算コストを投資した上での、非本質的計算コストを削減するに足る高精度の予測が重要であるといえる。

### 3.1.3.3 予測技術の実際

ここではまず、予測技術を特徴付ける分類項目を挙げ、続いて現状の高性能計算に用いられている予測技術の代表例を示すとともに、項目ごとの分類を行う。

#### (1) 予測技術の分類項目

**時間的予測と空間的予測** 現状の予測技術では、将来に生じるであろう事象を予測する時間的予測が主なものである。一方並列計算においては、前述のように他のプロセッサの計算状況などの認識に高いコストを要する情報を予測する、空間的予測も用いられている。

**静的予測と動的予測** コンパイル時などの計算開始以前に予測を行う静的予測と、実行時に予測を立てる動的予測とがある。また動的予測には、過去の履歴などに基づいて予測方法を修正する適応的なものと、そうではない単純なものがある。

**予測に関するコスト** 予測に関するコストとして、以下の3項目が挙げられる。

- 予測コスト：予測自体に要するコスト
- 成功コスト：予測失敗に備えるなど、予測が成功した場合でも支払うコスト
- 失敗コスト：予測失敗時に支払われるコスト

また予測コストと成功コストには、計算時間の形で支払われる時間的成本と、ハードウェア投資やメモリ消費などの形で支払われる空間的成本がある。また、ここでの失敗コストは、予測が完璧である場合に比べた時間ロスという時間的成本とする。

**予測精度** 予測が正しい確率。実際に支払われる失敗コストは、予測精度に相関する。

## (2) 予測技術の実例

### キャッシュ [1, 2]

キャッシュ（あるいはTLBなどのキャッシュ的機構）は、あるアドレスに対するアクセスが近い将来に行われるであろうという時間的予測に基づく機構であり、予測による高速化技術の代表例といえることができる。予測の方法は適応的ではない単純な動的予測であり、予測コストは時間的にも空間的にも0、また時間的成功コストも0である。空間的な成功コストはキャッシュメモリそのものであり、かなり大きなコストであるといえる。また失敗コストは、キャッシュライン中でアクセスされずにリプレースされる部分をキャッシュにロードするコストと、1度しかアクセスされないラインによってリプレースされるラインの再ロードコストであるが、予測精度（ヒット率）が高いことから小さいといえる。

キャッシュに関連する予測技術としては、ハードウェア/ソフトウェアによるプリフェッチがある。ハードウェアプリフェッチについては、過去のアクセス履歴によるストライド予測といった、適応的予測を行うものもある。一方ソフトウェアプリフェッチは、通常静的予測を用いる。またキャッシュと比べて、全くアクセスされないラインのロードによる失敗コストが余分に生じ、予測精度も特にハードウェアプリフェッチでは必ずしも高くない。

### 分岐予測 [3、4、5]

分岐予測とそれに基づく投機的実行も、時間的予測による高速化技術の代表例の一つである。予測方法には静的／動的の双方があるが、最近の主流は2ビットカウンタ方式のように一定の適応性を持った動的予測である。予測／成功に関する時間的コストは0であるが、空間的予測コストとして分岐予測テーブルが、また空間的成功コストとしてリオーダーバッファのような投機失敗に対処する機構が、それぞれ計上される。失敗コストは数クロック程度であるが、予測精度がさほど高くなく、50%程度の例もある。

また投機的実行の変形例として、分岐確率を静的に予測した上で、taken/not-takenの双方のパスに関する投機的実行を行うものもある。これは一種のリスクヘッジであり、他の予測技術には見られない方式である。

### 負荷分散 [6、7、8]

現状の並列計算における予測技術の代表例が（自動）負荷分散であり、様々な種類のものが提案／実装されている。負荷分散における予測は、負荷の大きさの予測とプロセッサの負荷状況の予測に大別される。

負荷の大きさの予測は多くの場合静的に行われ、for ループのように比較的単純なものを対象とするものが多い。しかし最近では、たとえばプログラム解析によって負荷の大きさがリストの長さに比例することを求めておき、実際の予測は実行時にリストの長さを測定して行うといった、一定の予測コストを投資する例も現れている。

プロセッサの負荷状況については、拡散型の分散方式のように近傍の負荷状況から大域的な状況を類推する空間的予測、非均質分散計算でのプロセッサ性能をベンチマーク性能などから類推する静的な時間的予測、計算の途中までに要した時間によって以後の負荷状況を推定する適応的な時間的予測などが用いられている。これらに要する予測コストは負荷状況の測定や記録など比較的小さいが、システムの各部の動作状況を積極的にモニターして得た値を負荷モデルに適応して予測するといった、高い予測コストを投資するものもある。

負荷分散では予測の失敗が計算の誤りを導くことはないため、成功コストは通常0である。一方失敗コストは、負荷の不均衡によって生ずるアイドル時間として計上される。

### Time Warp [9、10、11]

分散シミュレーションの代表的かつ原理的な手法の一つである Time Warp は、対象モデルを構成する各プロセスの局所時刻がシステム全体で揃っているという空間的予測に基づく手法である。予測方法は、常に局所時刻と因果関係の不整合が生じないと予測するとい

う意味で静的であり、予測コストも0である。成功コストは、予測失敗による因果関係の逆転に備えた、プロセスの状態遷移履歴の保存コストである。また失敗コストは履歴の巻き戻しと、その他プロセスへの波及であり、かなり大きな値となる。また予測精度はシミュレーション対象に依存するが、50%以下という例も報告されている。

Time Warp に関連する予測技術としては、局所時刻と因果関係の整合を仮定する範囲を限定し（すなわちある範囲の時刻のみを整合していると予測する）、その範囲を種々の方法で適応的に制御するものがある。このような方法では予測精度が向上するが、Time Warp にはない失敗コスト、すなわち不整合を予測することによる不要な待ちが生じる。

以上の実例を前節の分類項目にしたがってまとめると、表1に示すものとなる。

表1 予測技術の分類

予測技術	方式	コスト			精度
		予測	成功	失敗	
キャッシュ	T+D	0	S(cache)	小	高
プリフェッチ (S/W)	T+S	0	T(P/F 命令)	小～中	中～高
プリフェッチ (H/W)	T+A	S(予測機構)	S(SB)	小～中	中～高
分岐予測	T+A	S(BPT)	S(ROB)	小	中
負荷分散	T/S+S/D/A	T/S	0	中～大	低～中
Time Warp	S+S	0	T/S(履歴)	大	低～中

方式 X+Y ... X=T: 時間的予測, S: 空間的予測; Y=S: 静的, D: 動的, A: 適応的  
 予測/成功コスト ... T: 時間的, S: 空間的

### 3.1.3.4 予測技術の将来

過去および現在の回路設計技術の動向に基づいて将来の並列計算を展望すると、局所的計算と大域的通信のコスト差がさらに拡大するものと予想される。したがって3.1.3.2で述べた並列計算を行う場の空間的距離が相対的に大きくなり、高精度の予測がますます重要になる。そこで、予測技術の適用範囲の拡大と、予測精度の向上の2項目について、プロセッサやシステムのアーキテクチャがどのような方向で変革すべきかを議論する。

#### 適用範囲の拡大

3.1.3.3 で述べた現状の予測技術の多くは、予測失敗が計算の誤りを引き起こさない安全なものであるか、あるいは分岐予測のように狭い範囲で巻き戻す程度に留まっている。例外はTime Warpであるが、巻き戻し自体に要する失敗コストだけではなく、その準備の

ための成功コストが大きいことから、その改良の多くは予測の aggressiveness を制限する方向で行われている。

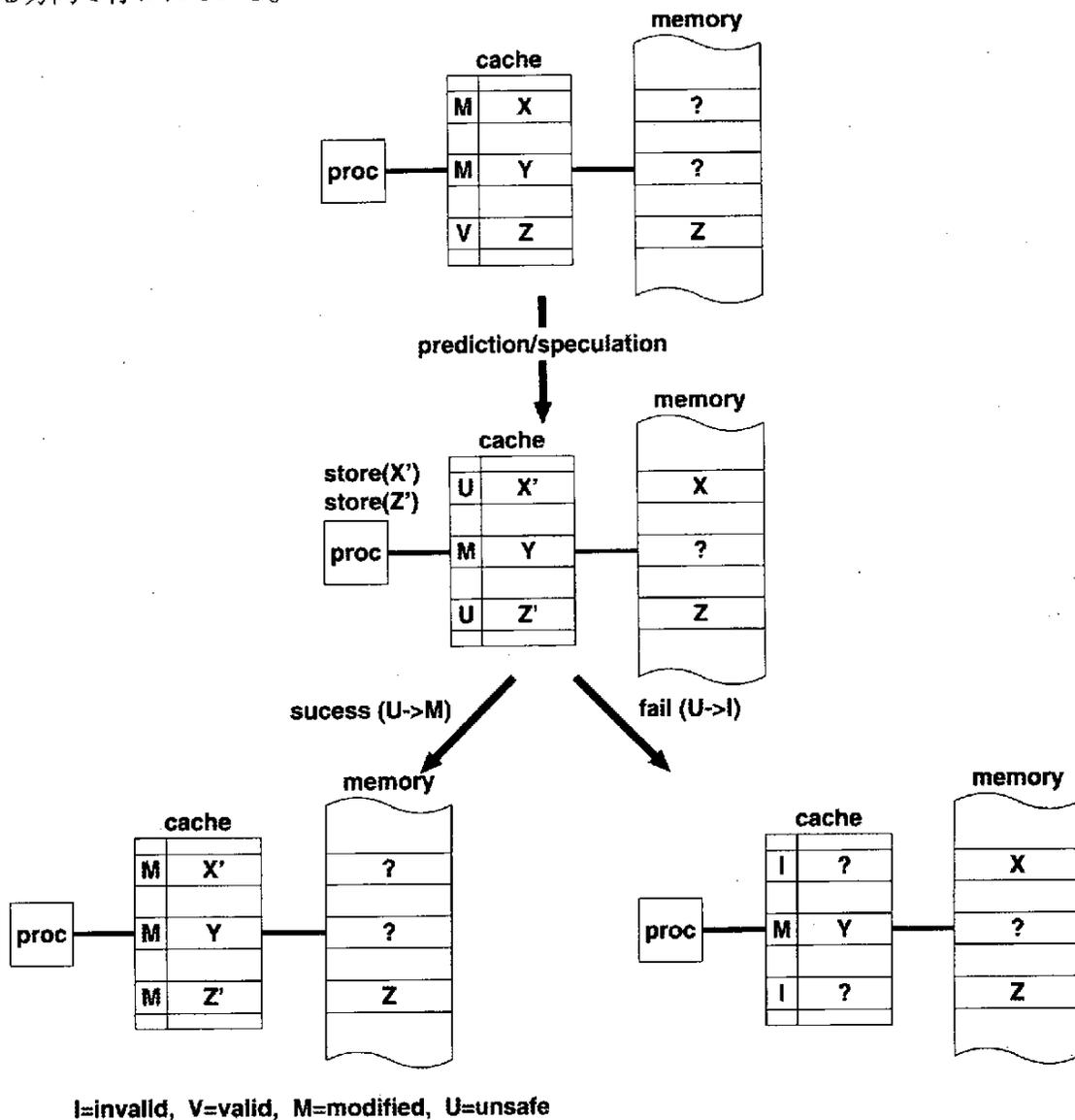


図2 キャッシュを用いた投機的メモリアクセス機構

一方、将来の並列計算では時間的／空間的に広い範囲の予測、すなわち予測してからその結果が判明するまでに長い時間を要するような予測が要求されるため、投機的なメモリ更新のように非限定的な投機的操作を許容するアーキテクチャが求められる。たとえば我々は、同期操作に対するメモリアクセスを投機的実行を行うために、キャッシュを用いて予測点での状態を低コストで保存する機構を提案している (図 2)。この機構では、投

機の開始、成功、失敗に関する操作が全て定数時間で実施できるため、結果の判明に時間を要するような予測を積極的に行うことができる。

なお上記の機構は、投機的実行に関するほとんどの操作をハードウェア化することによって予測に関するコストを最小限にしているが、そのために予測から結果判明までの期間がオーバーラップできないなどの制約が存在する。より一般的な機構とする場合には、成功コストの最小化を目的としたハードウェア機構と、ある程度の失敗コストを許容しつつ一般性を確保するためのソフトウェア機構の組合せが重要である。

### 予測精度の向上

ハードウェアによる予測機構は、時間的な予測コストが小さい反面、テーブルなどを用いた比較的単純な予測を行うため、予測精度の面で問題が生じる。したがってソフトウェアによる静的または動的な予測と、必要に応じてそれを支援するハードウェアを組み合わせるのが適切である。

ここで問題となるのは、現状の命令セット、アーキテクチャが予測のような一種のメタレベル計算に対応していないこと、すなわち予測に関わるハードウェア機構に対してソフトウェアが介入するための一般的な枠組がないことである。もちろん、個々のハードウェアごとに異なるであろう予測機構とのインタフェースを統一するのは不可能であるが、たとえばコプロセッサのような枠組で一般的なインタフェースを提供する必要がある。またこのような枠組をさらに進展させると、メタレベル計算に専念するコプロセッサの導入も考えられる。

### 参考文献

- [1] N. P. Jouppi. Improving Direct-Mapped Cache Performance of a Small Fully Associative Cache and Prefetch Buffers., In *Proc. ISCA'90*, pp. 364-373, May 1990.
- [2] S. Palacharla and R. E. Kessler. Evaluating Stream Buffers as a Secondary Cache Replacement. In *Proc. ISCA'94*, pp. 24-33, April 1994.
- [3] J. K. F. Lee and A. J. Smith. Branch Prediction Strategies and Branch Target Buffer Design. *Computer*, Vol.17, No.1, pp. 6-22, January 1984.
- [4] M. D. Smith, M. Johnson, and M. A. Horowitz. Limits on Multiple Instruction Issue. In *Proc. ASPLOS'89*, pp. 290-302, April 1989.

- [5] H. Ando, C. Nakanishi, T. Hara, and M. Nakaya. Unconstrained Speculative Execution with Predicated State Buffering. In *Proc. ISCA'95*, pp. 126-137, June 1995.
- [6] 佐藤令子, 佐藤裕幸, 中島克人. 疎結合型マルチプロセッサ上の拡散型動的負荷分散方式—LIS-G方式—. In *JSPP'93*, pp. 363-370, May 1993.
- [7] 後藤慎也, 窪田昌史, 田中利彦, 五島正裕, 森眞一郎, 中島浩, 富田眞治. 並列化コンパイラ TINPAR による非均質環境向けコード生成手法. In *JSPP'97*, pp. 205-212, May 1997.
- [8] R. Wolski, N. Spring, and C. Peterson. Implementing a Performance Forecasting System for Metacomputing: The Network Weather Service. In *Proc. Supercomputing'97*, November 1997.
- [9] D. Jerfferson. Virtual Time. *ACM TOPLS*, Vol.7, No.3, pp. 404-425, 1985.
- [10] J. Steinman. SPEEDES: Synchronous Parallel Environment for Emulation and Discrete Event Simulation. In *Proc. SCS Western Multiconf. on Adv. Par. and Dist. Simulation*, pp. 1111-1115, 1991.
- [11] 松本幸則, 瀧和男. バーチャルタイムによる並列論理シミュレーション. *情報処理*, Vol.33, No.3, pp. 387-395, March 1992.

### 3.1.4 ベタフロップスマシンに向けた技術開発のありかた (濱中直樹委員)

'80年代には性能がGigaflopsクラスのベクトルプロセッサ型スーパーコンピュータが商用化された。'90年代には性能がTeraflopsクラスの並列型スーパーコンピュータが商用化された。これに伴い多くの公的研究機関や民間企業がスーパーコンピュータを導入し、数値シミュレーションの技法を用いた計算機実験を通してさまざまな成果をあげてきた。

その一方で、スーパーコンピュータの開発費はその性能とともに膨大になり、多額の開発費を投入できるメーカーのみがスーパーコンピュータ市場に残れる状況になった。また、スーパーコンピュータへの期待感ほどには市場規模が拡大せず、多数のメーカーがビジネス判断としてこの分野から撤退し、結果として少数企業の寡占市場になった。

'90年代までの性能トレンドを単純に延長するなら、'00年代にはPetaflopsクラスのスーパーコンピュータの実現が期待されることになる。しかしこれまでのスーパーコンピュータの市場規模のトレンドを考慮すると、今後もビジネス判断としてスーパーコンピュータ市場からの撤退を決断する企業が出る可能性がある。その結果、企業間競争がなくなれば性能トレンドの維持は困難になる。さらに、当該市場に参入する企業が全くなくなるような事態が生ずれば、スーパーコンピュータそのものがなくなってしまう。

科学技術振興の観点から、このような状況を国策として是とはできまい。そのため、公的研究機関での実用化を意識した先行的研究活動や、複数企業の競争を通じた実用化技術の発展を支えるための国家レベルでの施策が必要になる。国費投入も必要になる。

以下では、Petaflopsクラスをめざすスーパーコンピュータ市場の健全な発展に必要な国家レベルの施策について、報告者の私見を述べる。

#### 3.1.4.1 ベタフロップスマシンに向けた技術開発の指針

最初に、Petaflopsクラスをめざすスーパーコンピュータ(以下、ベタフロップスマシンと呼ぶことにする)を開発していくために必要な技術開発の指針について述べる。

これまでのスーパーコンピュータ向きアーキテクチャ進展の経緯を見てわかる通り、ベタフロップスマシンが並列プロセッサであることは必然と考えられる。並列プロセッサのアーキテクチャには実にさまざまなタイプがあり、どのタイプが本命であるかは今後の研究によって究明されるべき事項である。しかし、どのタイプの並列プロセッサであっても、その実効性能は次式でモデル化できる。

$$P=rnp$$

ただし、

P: 並列プロセッサの実効性能

r: 処理効率(実効性能のピーク性能に対する比率)

n: 要素プロセッサ数

p: 要素プロセッサ性能

である。

PをPetaflopsクラスにまで向上するためには、r、nおよびpのそれぞれの向上が必要になる。そのための主な課題は次の通りである。

表1 ペタフロップスマシン実現のための主な技術課題

課題	応用ソフト	コンパイラ・OS	アーキテクチャ	ハードウェア
r 向上	高並列算法	高効率並列実行制御	高効率協調動作 データ参照局所化	高速データ転送
n 向上	-	-	-	高密度実装
p 向上	高速算法	高効率 CPU 制御	命令レベル並列	高周波数動作

r、n、p が互いに独立であるならば、それぞれの向上の目標を設定し、それに向かってそれぞれの分野が邁進すれば良い。実際、並列プロセッサ研究開発の黎明期には、r、n、p を独立とみなして研究を進めることができた。しかし、並列プロセッサの実用期になると、もはや r、n、p は独立とはみなせなくなった。並列プロセッサ研究者の間では、次のような依存関係が常識として認知されている。

(1) r は n および p に依存する。n が大きくなるほど r の向上が困難になる。

(2) n の向上と p の向上は背反の関係にある。なぜなら、n を大きくするためにはコンパクトな要素プロセッサが有利だが、p を大きくするためには命令レベル並列処理技術等のハードウェア量拡大が必要だからである。並列プロセッサとして見れば、結局は  $n \times p$  を向上すればよいのは確かだが、そのために最適な n、p それぞれの目標は見えていない。

このような場合、r、n、p の依存関係を明確にするために、よりベーシックな独立変数で r、n、p を定式化することが科学的なアプローチかもしれない。ただ、目標自体が「00年代にペタフロップスマシンを実現」という工学的な目標であるため、本稿でも工学的な視点からの分析を進める。しかし上記のような科学的アプローチは極めて大切であり、別途、あるいはペタフロップスマシンの開発の一部として推進する必要があることを忘れてはならない。

現状の技術レベルとして、 $r$ の値はおおむね0.1以上、ときには1に近い値を達成していると考えられる。そのためテラフロップスマシンからペタフロップスマシンへの1,000倍の性能向上は、主として $n \times p$ の向上に依存する。また上述のように $n$ が大きくなるほど $r$ の向上が困難になることを考慮すると、 $n \times p$ の向上においては $p$ の向上を優先する必要がある。以上を考慮し、ペタフロップスマシンを実現するための技術開発の指針を一言にまとめるならば、次のようになる。

#### [指針]

$p$ の向上を最大限に考慮した上で、 $n \times p$ の1,000倍の向上を図る。さらにこの過程での $n$ の増大がもたらす影響を克服し、 $r$ の維持あるいは向上を図る。

#### 3.1.4.2 ペタフロップスマシン開発の指針を遂行する上での課題

ここで $r$ 、 $n$ 、 $p$ それぞれの現状ならびに向上努力の動機について概観する。

$r$ の向上は、並列プロセッサに特有な重要課題であるため、大規模並列処理技術の中心として世界的に研究されている。

$p$ の向上には、大きな利潤をもたらすマイクロプロセッサ向き商用技術としてアメリカを中心に盛んに投資されている。マルチメディア応用等があるためMIPSのみならずFLOPSにも競争があり、商用マイクロプロセッサにも高性能な浮動小数点演算器が搭載された。そのため、商用マイクロプロセッサ技術はペタフロップスマシンのために極めて有用と考えられる。

$n$ の向上の直接的な素地は高密度実装技術にある。高密度実装技術には、主としてバイポーラLSIを多数搭載することで高性能を実現するメインフレーム計算機向き技術として投資されてきた。メインフレーム計算機が大きな利潤をもたらしてきたのは周知の通りである。また、この分野では日本企業に一流の技術があるのも周知の通りである。そのため、ペタフロップスマシンの開発においては日本の貢献が重要である。

しかし、 $n \times p$ の大幅な向上については、科学技術計算向き並列型スーパーコンピュータ実現のために必要とは考えられたが、その市場規模があまりに小さいため大規模並列プロセッサに特化した $n \times p$ 向上のための技術開発への投資は積極的には進められなかった。そのため、 $n \times p$ 向上の究極の姿を考慮した $r$ 向上の研究も進んでいないと言える。思いつきの範囲を越えないのは承知で例示するが、例えば下記のような研究テーマが有り得ないと誰が言えようか。

- (1) LSI の発熱量を抑制し実装密度を高めるためのプロセススケジューリング方法の研究
- (2) LSI 微細化を演算器幅拡大・演算精度向上に利用することで収束性を向上する算法の研究

ペタフロップスマシンのような高い目標を達成するためには、 $n \times p$  向上のための技術にブレークスルーが必要である。ブレークスルーは日常の常識の外に存在するはずである。そのためには、上記のように一見すると日常の常識の範疇にないような、設計の階層を越えたコンピュータ技術のシステム的な総合最適化のためのアイデアが必要と考えられる。学術面と企業の連携も必要になる。アメリカのマイクロプロセッサの成功にしても、カスタム LSI 設計の最適化ということで、コンパイラ技術やアーキテクチャ技術から回路技術までの設計階層を越えたシステム最適化の結果と見ることができる。

コンピュータ技術のシステム的な総合最適化というキーワードのレベルなら学術面でも企業でも共有できる。しかし、これを具体的に推進するための目的ならびに意図は、学術面と企業では一致しない。なぜなら具体的な推進には相応の投資が必要だが、学術面と企業では投資の回収に対する考え方が異なるからである。

しかし、推進するための目的ならびに意図が完全に一致はしないにしても、全く異なっているという訳でもない。そのため、共通点と相違点を浮き彫りにし、それぞれに対する育成計画を具体化することがペタフロップスマシン開発を遂行するための課題になる。

#### 3.1.4.3 商用マルチプロセッサとペタフロップスマシンの共通技術

現在のコンピュータ企業はコンピュータ技術のシステム的な総合最適化というキーワードを、主として商用マルチプロセッサのための技術開発に関連すると受け取る。多くの企業の行動指針には社会貢献が含まれているだろうが、コンピュータ開発関連投資が膨大になった現在、企業の存在基盤確保なくしては、ペタフロップスマシンのような社会貢献的な事業は推進できない。

多くのコンピュータ関連企業は、目下その存在基盤確保のためビジネス用途向きの大型サーバに視点を置いている。その大きな理由は下記のような市場ニーズの変化にある。

近年の商用マイクロプロセッサ技術の急速な進展に伴い、ビジネス用途向きコンピュータの価格性能比が急激に向上した。そのためダウンサイジングということで商用マイクロプロセッサベースの情報機器による分散処理が進み、情報機器数が増大した。その結果、情報機器の管理コストが急激に増大し、これの削減が望まれている。

また、ビジネス用途向きコンピュータ開発への参入が容易になり多数のコンピュータ企業がさまざまな製品を投入する中で、ソフトウェアのポータビリティを確保していくため OS の業界標準化が進んだ。その結果、新しく開発される魅力的なソフトウェア製品も業界標準 OS をターゲットにするようになり、業界標準 OS の搭載を無視したコンピュータ製品の商品価値は著しく低下した。

また、商用マイクロプロセッサベース情報機器の価格性能比の急激な向上に伴い、従来型のメインフレーム計算機の割高感が著しく増加した。そのため、従来型メインフレームを置き換えられるような商用マイクロプロセッサベースのコンピュータが求められている。

これらのニーズを解決するために、商用マイクロプロセッサを利用し業界標準 OS を搭載したビジネス用途向きの大型サーバが注目されている。大型サーバには従来ソフトウェアの継続使用等のニーズもあるため、共有メモリ型マルチプロセッサ方式が主流になっている。これらの大型サーバ製品は性能面・信頼性機能面双方での多数企業間の競争が激しく、著しい進歩を遂げてきたし、この進歩は当面続くと考えられる。

ところで、大型サーバと言っても、並列プロセッサの一種であるため、その性能は、

$$P=rnp$$

でモデル化できる。r 向上のための課題や方法論はペタフロップスマシンとは異なるが、ペタフロップスマシンの要素プロセッサにマイクロプロセッサが有望であることも考慮すれば、 $n \times p$  の向上という観点からは共通点がある。

また、大型サーバは共有メモリ型を基調にする一方、ペタフロップスマシンは分散メモリ型が基調と考えられている点が異なるように見えるが、もしペタフロップスマシンを共有メモリ型マルチプロセッサ構造を内包するクラスタ型で検討するなら、 $n \times p$  向上の共通点はますます高まる。

信頼性機能については大型サーバとペタフロップスマシンと概ね共通と考えられる。

つまり、個々の要素技術開発という面からは、大型サーバとペタフロップスマシンの間に多くの共通点を見い出せるはずである。

#### 3.1.4.4 ペタフロップスマシン開発を成功させるために

ペタフロップスマシンは、個別要素技術の進展のみならず要素技術相互間の総合的な最適化なくしては実現しえないことは既に述べた。そのためには、いわゆる並列処理技術の範疇にとどまらず半導体技術を含めたハードウェア技術開発のための活動も必要になる。

マイクロプロセッサ技術などはアメリカに一日の長があるため、この活動には当然国際的な協力が含まれる。

その一方、ペタフロップスマシンそのものの市場規模は小さく、これ自体を目的にした技術開発活動は企業には困難である。特に企業の貢献が期待されるハードウェア技術の開発には多額の投資が必要なためである。

こういう環境の中で、ペタフロップスマシン実現に利用できるハードウェア技術開発を推進するためには、企業が注目している商用マルチプロセッサとペタフロップスマシンの共通点と相違点を明確にし、それぞれに対する国家レベルの育成計画を立案していくことが必要になる。

共通点については、企業の独創性や自主性を重視し激しい企業間競争を通じた発展を促すことが効率的と考えられる。そのため国家としては、企業間競争を加速させるための規制緩和ならびに、ペタフロップスマシンとの共通技術の開発加速を誘導するための税制面での優遇措置を通じた制度支援を中心にして育成計画を立案すべきと考える。

相違点については、企業には積極的に取り組みにくい項目であるため、国家がリーダーシップを発揮して推進していく必要がある。具体的には国家プロジェクト体制での推進になるだろう。ただ前述の通りペタフロップスマシンの実現には国際協力が欠かせないので、国外企業を含めた体制の確立が必要になる。しかし国家の指導という形では国外企業が参加できる筈もない。それゆえ参加する企業にとってのメリットを熟慮し、積極的に参加できるような魅力あるプロジェクトを世界に提示しなければならない。もちろん、多額の国家経費の投入が必要になる。ただ多額の国家経費の投入にあたっては、その根拠や効能を納税者たる国民にわかりやすく説明する必要があることは言うまでもない。それは誰もが納得する基本コンセプトの提示であり、目標とする夢の提示でもある。つまり、国民にとっても魅力あるプロジェクトを立案する必要がある。

また、共通点の技術開発を推進するための制度支援と、相違点の技術開発を推進するための国家プロジェクトは表裏一体なことを忘れてはならない。そのため両者が互いに良い影響を及ぼしあうようにコントロールするための機構、言わば「メタ国家プロジェクト」の確立と維持も必要である。この機構が「ペタフロップスマシン開発」を手段と認識し、その目的を国民から見てクリアに定義すること、そして時代の趨勢とともに目的を随時更新していき、目的が色褪せてしまわないように機能していくことが、開発を成功させるための最重要課題と考える。

## 3.2 ソフトウェア

### 3.2.1 System on a Chip 化時代のシステムソフトウェア基盤技術

(石川 裕委員)

#### 3.2.1.1 背景

1チップ内集積度の向上ならびに DRAM 混載チップの出現により、家電、オーディオ、車載用コンピュータ、PDA、プラント制御、ポケットゲーム等あらゆる電子製品の分野で、さらなる高度情報処理を可能とするシステムが1チップで実現可能となってきた。1チップでシステムの持つ機能を全て実現できるので、このようなシステムは System on a Chip (以降 SOC と略する) と呼ばれている。SOC の台頭は、さらなる付加価値の高い電子製品が実現できるだけでなく、従来組み込むことが出来なかった製品にも組み込まれることが可能となってきた。このような背景の下、製品開発競争の激化により、短期間に限られたハードウェア資源上で、付加価値の高い電子製品開発を可能とする環境が必要とされている。例えば、MPEG エンコーダを搭載したマルチメディア対応チップの製品開発には、3年間で 60 名規模の人材が必要とされる。延べ 180 人年のマンパワーを要しているが、180 人いれば1年間で開発が完了できるというものではない。設計試作の短縮化のための開発環境の整備、すなわち、CAD、コンパイラ、組込み型実時間オペレーティングシステムの整備が必要である。

SOC は、組み込み用計算機としての用途だけでなく、SMP 型並列計算機や次世代超並列計算機[3]高速ネットワークルータの要素技術として使われる重要な電子産業基盤でもある。このような観点において、SOC は 21 世紀の産業基盤となる重要な領域と位置づけられる。

SOC 関係では、ハードウェアあるいはアーキテクチャの観点から研究開発支援の必要性が言われてきている[1]。ここでは SOC 開発のための支援ツールおよび SOC を使ったシステムを開発するときのシステムソフトウェア技術およびその研究開発支援について筆者の思うところを述べてみる。

#### 3.2.1.2 設計支援環境

従来より、ハードウェア記述言語で記述されたシステムのシミュレータ、論理合成ツール、論理レベルのシミュレータ、レイアウトツール等が CAD として提供されてきている。

これらツールの中には数日から月単位の処理時間を必要としていて、短期間の設計を可能とするためには高速計算機を必要としている。

限られた開発期間内にシステムを開発するためには、チップやボードの試作品を作成する前に、チップ内部のシミュレーションだけでなく、チップを含むシステム全体をシミュレーションし、動作検査できる環境が必須である。このようなシミュレーション環境は製品毎に構成が異なるので、システムを検査するためのツールを構築するためのツールが必要となる。このようなツールは、単にハードウェアの機能検証だけの目的だけでなく、ファームウェアあるいはソフトウェアの開発、改良のために使われ、ハードウェアの開発と並行してソフトウェアの開発も可能となり、開発の短縮化に貢献する。より複雑なシステムチップを短い設計サイクルで実現するために、Virtual Socket Interface(VSI)とよばれる業界標準化の動きがある[2]。VSIでは、チップ内で使用される機能ブロックのIP(Intellectual Property)を保護し、ユーザがIPプロバイダの提供するVirtual Component(VC)を利用するために必要なインタフェースを規定しようとしている。設計支援ツールはこのようなインタフェースを含む形で構築される必要があるであろう。

### 3.2.1.3 Building Blocks 型オペレーティングシステム

現在のオペレーティングシステムは、組込み型オペレーティングシステムとPCやワークステーション用のオペレーティングシステムとで異なっている。今後重要となるのは、オペレーティングシステムの部品化である。組込み型計算機の上でのプロセス管理は、ワークステーション用のプロセス管理とは異なる。プロセス間交信も異なる。システムコールレベルあるいはライブラリレベルでAPIを統一し、プログラマの学習時間の軽減をはかることによって、開発コストを軽減することが可能となる。

従来の組込み型OSは、多くの場合既存のMPUを持ったシステムを想定している。従来自前でランタイムを用いていた製品にもオペレーティングシステムの必要性が増大するだろう。

実時間処理用OSと言う観点からは、組込み用途を主としたITRON[4]、Unixの拡張としてのPOSIX Real-Time Extension、Free OSとしてもっとも多くのプラットフォーム上で稼動するLinuxを拡張したReal-Time Linux[5]等がある。また、組込み型プロセッサ上でのOSとしてuITRON、Embedded OS-9、pSOS、Embedded OS/2 Warp等も存在する。これらオペレーティングシステムは、プロセッサとしては汎用マイクロプロセッサあるいはマイクロコントローラを想定している。今後、SOC向けの機能が限定されたオペレーティングシステムが必要とされるであろう。

### 3.2.1.4 プログラミング言語

組込み型計算機用プログラミング言語に求められているのは、実時間処理記述と適応可能性である。実時間処理記述においては、タスクの実時間制約記述により静的にデッドラインが満たされるかを検証できるような処理系が望まれる。今までの実時間処理記述言語は、時間制約記述は存在する[6]。時間制約記述により制約のあるタスク間でのスケジューリングが可能かどうかの検査は可能であっても、記述された制約自体が与えられた計算機上で制約が満たされるかどうかを静的に検査する処理系は存在しなかった。今後、ソフトウェアの設計およびプログラミング時に時間制約が満たされるか、必要とされるメモリ容量がどのくらいかを予測できる処理系が必要である。

特殊ハードウェアによる製品開発を短期間で成し遂げるためには、特殊ハードウェアを容易に制御できるプログラミング機能が提供されなければならない。新しいハードウェア毎にそのためのプログラミング言語を開発する時間はない。そこで、プログラミング言語自体が新しいハードウェアに応じたプログラミング構文や最適化が提供できる枠組みが望まれる。これを実現する要素技術としてはコンパイル時リフレクションと言う概念がある。

### 3.2.1.5 応用

#### (1) SOC

家電製品、PDA 用組込みチップ SOC を用いた製品における大きな市場の一つは、言わずと知れたコンシューマ向け製品である。マルチメディアのためのデータ圧縮機能とネットワーク機能を搭載した SOC は大きな市場となる。

#### (2) 次世代並列計算機用チップおよび用途

DRAM 混載型チップにより 1 チップで 4 から 8 個のプロセッサとメモリを搭載することが可能となると、現在の Teraflops 級計算機が一つの筐体で実現できるようになる。次世代並列計算機と言ったときに、Petaflops 級計算機ではなく、現在のサーバ計算機のような一つの筐体に納まった計算機が Teraflops 級の処理能力を持つ計算機のことを指すべきである。Teraflops 級計算機は、CAD ソフトウェアや自然環境シミュレーション等の計算科学に使われるであろう。SOC 設計支援環境としての CAD ソフトウェアがこのような計算機上で実現されることにより、論理合成、論理レベルシミュレーション、レイアウト、回路シミュレーション等が短時間で終了することが可能となる。これにより設計のサイクルが早まる。

自然環境に関するシミュレーションでは、地震が起こってからの津波予報、都市火災や山火事における効果的消火活動を支援するためのシミュレーション等では、リアルタイム処理以上の処理による高速シミュレーションを必要とする。

Teraflops の次にくる Petaflops 級計算機の用途の一つは、分子動力学シミュレーションによる計算科学である。例えば計算分子生物学の世界では、タンパク質立体構造予測がある。タンパク質立体構造が予測できると新薬の開発に利用でき、その市場は計り知れないものがある。分子動力学シミュレーションによるタンパク質立体構造予測では、現在の数十 Gigafllops 級高並列計算機において1 ナノ秒のシミュレーションに1日強かかるといわれている。現実的に使用可能なのは1ミリ秒のシミュレーションであり、現在の6桁速い計算機すなわち Petaflops 級計算機を必要としている[7]。

新情報処理開発機構における 64bits プロセッサ開発経験を聞くと、論理部分で 200K ゲートレベルの論理合成に1週間、論理レベルシミュレーションは1秒間に数クロック程度、レイアウトには1ヶ月程かかっていたようである。論理レベルシミュレーションの高速化は、テストパターンを豊富にすることが可能となり早期バグ出しが可能となる。強いては開発サイクルを短くすることに貢献する。

### (3)次世代ルータチップ

2から3年のレンジでは、High End Gigabit Ethernet Switch のための SOC が重要となる。現在、ハイエンドルータは米国企業の独占状態になりつつある。Gigabit Ethernet および将来のさらなる高速ネットワークに対応したネットワークルータの市場は、今後ますます増大するだろう。

#### 3.2.1.6 国に期待するもの

SOC 技術の重要性は企業においては既に認識されている。SOC 開発のための基盤技術について、企業は既存の CAD 製品を利用し、コンパイラはフリーソフトウェアの gnu を利用していたりする。21 世紀における我が国の産業基盤である電子製品の国際競争力を維持するためには、CAD ソフトウェア、コンパイラ、オペレーティングシステムを育成していく仕組みを必要としている。このためには、企業毎にローカルにノウハウを蓄積するのではなく、中立的組織を中心として、産・官・学が協力して基盤ソフトウェアを育成していく必要がある。もし、CAD ソフトウェア、コンパイラ、オペレーティングシステムが共有されれば、大学、研究機関からの最適化のアルゴリズムやその他新しいアイデアを実際のシステムソフトウェアに組み込むことが可能となり、最先端の技術をメーカーが享受でき

るようになるのではないだろうか。情報産業はハードウェア技術のノウハウ蓄積からソフトウェア技術のノウハウ蓄積に比重が変わってきている。ソフトウェア技術の蓄積には、共通基盤としてのソフトウェアが恒常的に保守され、新しい技術を吸収していく枠組みが必要である。

夢のあるプロジェクトという意味では、パーソナルテラフロップス計算機、すなわち、現在、米国の国防関係の研究所でしか利用することが出来ない高性能計算機を個人が持つようにするための技術開発があげられる。これを遂行するためには、SOC素子技術、SOCのためのシステムソフトウェア基盤技術の整備が必要であり、それを利用したプロセッサアーキテクチャ、通信アーキテクチャ、システムソフトウェアを開発する必要がある。このような総合システム開発を国が主導することは、国内の産業を育成し国力をあげるものであると確信する。

#### 参考文献

- [1] 三輪 晴治、「半導体/システム・オン・チップが国際競争の焦点に」、日経ビジネス 2月9日号、pp. 112 - pp. 116, 1998.
- [2] <http://www.vsi.org/>
- [3] Kurt Oeler, 「Clinton touts "supercomputer on a chip"」, NEWS.COM.
- [4] ITRON <http://tron.um.u-tokyo.ac.jp/TRON/ITRON/>
- [5] <http://luz.cs.nmt.edu/~rtl/linux/>
- [6] Yutaka Ishikawa, Hideyuki Tokuda, Clifford Mercer, "An Object-Oriented Real-Time Programming Language," IEEE Computer, Vol. 25, No. 10, pp. 66 - 73, 1992.
- [7] 新情報処理開発機構 秋山氏との私的会話による

### 3.2.2 並列化コンパイラにおける研究課題と展望 (笠原博徳委員)

#### 3.2.2.1 はじめに

本節では、21 世紀の HPC 技術の発展に向けた自動並列化コンパイラ研究開発の重要性を述べ、国内の研究開発状況を海外との比較をまじえながら外観し国内の並列化コンパイラの製品化が遅れていることを示し、今後の対処策について論じる。

また、具体的に、現在直面しているコンパイラによる命令レベル並列性、ループ並列性の抽出の限界を打ち破るために、複数の粒度の並列性を使用することによりプログラム全域の並列性抽出を可能とするマルチグレイン並列化が重要であることを述べる。

さらに、他の重要な研究課題として、データのローカルメモリあるいは分散共有メモリへの自動分散（データローカライゼーション）、データ転送と処理のオーバーラッピングスケジューリング、コンパイラによるキャッシュ制御などについて述べる。

最後に、将来の実効性能が高く、使い易い HPC（スーパーコンピュータ、シングルチップマルチプロセッサを含む）を開発するためには、コンパイラによる並列化を支援するアーキテクチャが不可欠で、ソフト・ハードの協調設計が重要であることを述べる。

#### 3.2.2.2 自動並列化コンパイラ研究開発の必要性

現在日本のスーパーコンピュータ業界は世界トップのハードウェア技術をもっており、現時点でピーク性能は 1Tflops を越え、21 世紀初頭には数 10Tflops 以上のピーク性能をもつマシンが開発されると予想される。

しかし、現在のスーパーコンピュータは、ピーク性能の向上と共に実際にプログラムを実行したときの真の性能（実効性能）との差が大きくなっている、すなわちコストパフォーマンスに問題が生じている。また、使い勝手的にも、ユーザが問題中の並列性を抽出し、HPF、MPI、PVM などの拡張言語、ライブラリを使用して、ハードウェアを効果的に使えるプログラムを作成しなければならず、一般のユーザには使いにくい、あるいは使いこなせないという問題がある。これらの問題にも関連し、世界の HPC マーケットを拡大できないという問題が生じている。

これらのコストパフォーマンス、使いやすさの問題を解決し、スーパーコンピュータのマーケットを拡大するためには、ユーザが使い慣れている Fortran 等の逐次型言語で書かれたプログラムから並列性を自動的に抽出し、ハードウェアを有効に動作させるようにスケジューリングする並列化コンパイラの開発が重要となる。

一方、汎用マイクロプロセッサの分野では、我が国の産業界は世界最高レベルとは言い難い状況にあることが認識されている。しかし、従来我が国の電子産業を支える一つの柱となっていた DRAM 製造による利益確保が将来的に困難と予想される状況では、より付加価値の高い汎用マイクロプロセッサの開発が重要であると思われる。

その際、海外企業が大きなシェアをもち、さらに命令レベルの並列性の限界から将来的な実効性能の向上が難しいと予測されるスーパースカラや VLIW ではなく、21 世紀初頭の有力なアーキテクチャの一つとなると考えられるシングルチップマルチプロセッサについて検討を行うことは我が国産業界がこの分野で一定のシェアを獲得するために重要と考えられる。また、このようなシングルチップマルチプロセッサに関する検討は、上述の HPC の価格性能比向上に対しても重要な役割を果たすものと思われる。

ただ、このようなシングルチップマルチプロセッサの研究開発を行う際にも、単に従来の主記憶共有メモリアーキテクチャで、プロセッサを集積しただけでは、世界に対する国産プロセッサの優位性は得られない。アーキテクチャの独自性、高性能、低価格を達成するためには、プログラム中の命令レベル並列性、ループ並列性、粗粒度並列性をフルに使用できるマルチグレイン並列処理のようなシステムの高性能化及び誰にでも使える自動並列化を可能とする新しい並列処理手法を実現するコンパイラ技術と、それを生かせるような、日本独自のアーキテクチャの開発が重要である。

このような汎用マイクロプロセッサの分野で、世界のシェアを獲得するには、近い将来にスーパースカラあるいは VLIW のように命令レベル並列処理の限界が顕著になり、他アーキテクチャへの移行が進むと考えられる 21 世紀初頭に向けて製品を開発するのが良いと思われる。また、このアーキテクチャ移行の時期にシェアを獲得しないと、その後のプロセッサでのシェアの獲得は非常に難しいものになると危惧される。

### 3.2.2.3 コンパイル技術研究の現状

上述のように、HPC 分野、マイクロプロセッサの将来を支えるためには、自動並列化コンパイラの研究開発が重要である。

この自動並列化コンパイラに関する世界の研究状況をみると、従来自動並列化コンパイラの研究は、米国イリノイ大を中心に行われてきた。イリノイ大は、1970年代前半の ILLIAC IV の開発以来、並列処理コンピュータのキーテクノロジーとして Prof. David Kuck、Prof. David Padua を中心に研究を行い、

- Paraphrase プロジェクト
- 並列化コンパイラ技術を生かすマルチプロセッサスーパーコンピュータ開発の

## ための Cedar プロジェクト

### ● Cedar 並列化コンパイラを拡張した Polaris プロジェクト

などを通じ、現在、世界の大学、SGI CRAY、IBM、Intel、HP を始め、PGI、KAI などのベンチャーで活躍している優れた人材を世界に向けて輩出している。

また、その他では、Rice 大、Vienna 大と企業を中心とした HPF とそのコンパイラの開発を始めとし、Stanford 大、Purdue 大、California 大、Maryland 大、Ohio State 大など、多くの大学でアクティブな研究がなされている。またヨーロッパも Delft 大、GMD 大、IRISA 等 大学、国立研究所、企業で活発な研究が行われている。

これに対して、国内では従来より自動並列化コンパイラの研究をしており、実際に動作するコンパイラを開発しているのは京都大学、早稲田大学等少数の大学、一部のコンピュータ企業のみである。また、商用の自動並列化コンパイラを比較してみると、我が国の製品は、ベクトル化コンパイラでは世界をリードしていたものの、マルチプロセッサ用コンパイラでは米国に比べ5年以上の遅れがあるように見える。また、RISC 系プロセッサ用コンパイラに関しても日本独自の並列化コンパイル技術は極めてわずかである。

また学会レベルの活動からみても、自動並列化コンパイラ及びコンパイラとアーキテクチャの関連が議論される著名な会議は、Int'l Conf. on Supercomputing, Int'l Conf. Parallel Processing, Supercomputing, ASPLOS, PPOPP 等である。しかし、この分野の研究者が最重要視している学会は、コンパイラ関係の著名な研究者のみ論文投稿及び出席を招待するアメリカ、ヨーロッパのクローズなワークショップである。これらのワークショップには米国を中心とした LCPC: Workshop on Languages and Compilers for Parallel Computers, ヨーロッパを中心とした CPC: Compilers for Parallel Computers)があり、約10年前から組織化され、高度な議論が展開されている。しかし、我が国からの参加者(招待者)は早稲田大学、京都大学、一部の企業からの数人のみであり、アメリカ、ヨーロッパと比べ圧倒的に少ない。

このような状況を打破するためには、産・官・学が協力したマルチプロセッサスーパーコンピュータ用及びシングルチップマルチプロセッサ用のコンパイラの開発、及びそのコンパイラの性能評価を活かすことができるアーキテクチャの開発を行う継続的なプロジェクトが必要であると思われる。ここで継続的と言ったのは、世界を断然リードするコストパフォーマンスが良く使い易いスーパーコンピュータ、マイクロプロセッサを一つ開発して終わりではなく、フロントランナーとして継続的に新しい技術を生み出していく力(従来の日本に求められた力とは異なるより難しい力)を備えられるようにしなければならないという考えからである。

特に、このようなプロジェクトでは、最先端の研究を進めるだけでなく、研究者、技

術者の絶対数が足りないことを考慮して、人材の育成にも力をいれるべきと思われる。

#### 3.2.2.4 産・官・学で今後研究開発すべきコンパイル技術

自動並列化コンパイラに関して今後研究を行うべき研究課題としては以下のようなものが考えられる。

##### ① マルチグレイン並列化技術

サブルーティン、ループ、基本ブロック間粗粒度並列処理（マクロデータフロー処理）、ループ並列処理、プロセッサ間（近）細粒度処理、SCM 内マルチグレイン処理、ベクトル処理を階層的に融合する技術で、プログラム全域の並列性を使用できる日本独自の新コンパイル手法であり、関連して以下の基礎技術要素の研究も含まれる。

- インタプロシージャ解析の高度化
- データ依存解析技術の高度化
- マルチグレイン処理における階層的プロセッサクラスタリング手法

##### ② データローカライゼーション技術（データ分割配置）

従来はHPFなどによりユーザが最適化しようとしていたが難しかったデータの分割及びローカルメモリへの配置技術を、以下のような技術を用い自動化しようとする技術であり、以下のような要素技術から構成される。

- ループ間グローバルデータ依存解析
- データ整合分割、マクロタスクタスク融合
- パーシャルスタティックスケジューリングを伴うダイナミックスケジューリング技術

##### ③ 処理と通信のオーバーラッピングスケジューリング

これは、データローカライゼーションなどによりデータを最適配置しても削除できなかったデータ転送オーバーヘッドを、タスク処理とデータ転送をオーバーラッピングさせて（同時に）行うことにより、データ転送のオーバーヘッドを隠蔽しようとする技術であり、タスク実行およびプレロード、ポストストアを用いたデータ転送タイミングを最適に定める、スタティックスケジューリング、ダイナミックスケジューリング及びその混合モードのアルゴリズムの開発が重要である。

#### ④ キャッシュのソフトウェア制御技術

上述のプレロード、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スケジューリング技術、データの分割配置技術の融合することにより、キャッシュのようなプロセッサ直近の高速メモリをコンパイラに制御させ、プログラム中の情報をフルに使うことにより、ハードウェアキャッシュにおけるミスヒットを劇的に減らそうとする技術であり、シングルチップマルチプロセッサの高速化の際に特に重要になると考えられる。

#### 3.2.2.5 まとめ

以上のようにマルチプロセッサ用自動並列化コンパイラは、我が国が現在世界最高のハードウェア技術をもっている HPC 分野において、ハードウェアの高い性能を有効に引き出しプログラムを実行した時の真の性能（実効性能）を高め、価格性能比を改善するとともに、使いやすさを向上させるために重要な技術である。

またこの価格性能比の向上と使いやすさの向上は、従来比較的小さいと言われていた世界の HPC マーケットを拡大させ、我が国の HPC 産業をさらに発展させることができると期待される。

さらに、マルチグレイン並列処理のようなコンパイル技術は、21 世紀に日本の情報産業を支える技術となりうる高性能なシングルチップマルチプロセッサの開発のためにも重要である。

また、上記のようなベタフロップを睨んだ、コストパフォーマンスに優れた使い易いスーパーコンピュータの開発は、コンピュータ産業の発展のみならず、高速計算を必要とする省エネルギー・新エネルギーを始めとした地球環境保全、原子力安全利用、電力安定供給、自動車・航空機設計等各種関連産業の発展にも大いに貢献できるものと期待される。

### 3.2.3 並列プログラミングインタフェースの現状と課題について

(妹尾義樹委員)

現在のデバイステクノロジーの延長を仮定し、今後10ないし15年のレンジでペタフロップスマシンを開発しようとする、少なくとも1万~10万プロセッサから構成される分散メモリ型の並列システムとなる。このようなシステムの構築に際しては、デバイス、アーキテクチャ、応用ソフトウェアが重要なのはもちろんであるが、プログラミングインタフェースに関する技術がとりわけ重要となる。高並列マシンにおいては、高いピーク性能を引き出す並列プログラミングが非常に困難であるからである。本節では並列プログラミングインタフェース技術について、現状と課題を述べる。また米国の主要な関連プロジェクトを概観し、我が国の今後のプロジェクトのあるべき姿を考える。

#### 3.2.3.1 並列化記述言語の現状

現在の並列システム上の並列プログラミングとしては、メッセージパッシング(MPI: Message Passing Interface[1]など)、データパラレル言語(HPF: High Performance Fortran[2]など)、共有メモリ用並列化指示行インタフェース(OpenMP[3]など)、スレッド、並列化記述インタフェース、自動並列化インタフェースが代表的なものである。これらそれぞれについてその概要と現状について概観する。

##### (1) メッセージパッシング

メッセージパッシングは、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(Single Program Multiple Data Stream)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。

MPI(Message Passing Interface)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。仕様はMPIF(MPI Forum)と呼ばれるプライベートなフォーラムで検討され、1994年6月にMPI1.0がまとめられた。その後、並列I/Oやプロセスの動的生成・消滅、1方向通信(One Sided Communication)などの機能をさらに追加したMPI2.0[4]の言語仕様が1997年7月に定められた。MPIの詳細についてはホームページ<http://www.mcs.anl.gov/mpi/>を参照されたい。

MPIは、現在、大半の商用並列マシン上でサポートされている。これらのMPIにはArgonne国立研究所とMississippi州立大学との共同で開発されたMPICH[5](<http://www.mcs.anl.gov/mpi/mpich/>参照)をもとにしたものが多い。MPICHは仮想デバ

イスの概念を導入することにより、システム依存部がコンパクトに切り離された構造になっており、移植性が高い。ベンダは、システム依存部を対象ハードウェアに適するように開発することで、効率の良いMPIを実装することができる。

MPI1.0について、1対1通信を行う限りにおいては、並列システム固有の最適化がかなり行われており、高い通信性能が期待できる。一方で、集団通信などのそれ以外の機能については、まだ満足できる性能レベルにあるとは言い難い。それぞれのシステムにおいて、さらなるマシン固有のチューニングが必要である。MPI2.0については、そろそろ最初のインプリメントが登場しようかという段階である。いずれにせよ、チューニングおよび、利用経験の蓄積と、これをシステムヘフィードバックし、安定して高速なシステムとするには、あと1、2年はかかると思われる。

また、MPIの利用経験の蓄積やプログラミングツールの開発においては、まだ日本と欧米でかなりの開きがある。日本でMPIを駆使した並列プログラムを開発できる技術者が欧米に比べて非常に少ないことが主な原因と考えられる。このような技術者、研究者の育成は日本のHPC技術分野の振興にとって最も大きな課題の1つである。

## (2) データパラレル言語

大規模な数値シミュレーションでは、配列として記述された大規模なデータについて、要素ごとに同様の処理を施すことが多い。データパラレル言語とは、このような処理に内在する並列性の抽出を主要目的として考案された言語インタフェースであり、商用システムとしては、コネクションマシンに実装されたCM-Fortranが最初である。並列化においては、処理の分割と対象データの分割が重要であるが、データの分割に重点をおいて並列プログラムを記述する。このため、分散メモリマシン用の並列化記述インタフェースとして適している。

現在のところ、HPF (High Performance Fortran) [1]が、デファクト標準として認知されている。HPFは、Rice大学のKen Kennedy教授を中心とするプライベートな組織HPFF (HPF Forum)で定められた言語である。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、1997年1月にHPF2.0[6]の仕様がまとめられた。HPFFのホームページが、<http://www.crpc.rice.edu/HPFF/>にあり、HPF言語の仕様や関連研究プロジェクトの情報が得られる。

HPF処理系(コンパイラ)のインプリメントは、APR、PGI、NUGなどの欧米のベンチャー企業が先行した。現状では、PGIの処理系がシェアのかんりの部分を押さえている。最近では、日米の大企業によるHPF製品化が増えてきた。IBM、DEC、日立、富士通、NECなど

が自社並列システム用の HPF 処理系を製品化済み、もしくは製品化アナウンスしている。まず HPF1.1 をサポートし、HPF2.0 の公認拡張仕様を順次開発中というところが多い。

HPF の問題点としては、言語仕様、処理系が未成熟であることや、利用経験の蓄積の不足があげられる。特に、不規則なデータ参照パターンを持つ応用プログラムで効率の良いプログラムが書けない、あるいは書きにくいという HPF 言語の制約が大きな問題となっている。また、欧米では、Rice 大学、Vienna 大学、Syracuse 大学、GMD、Indiana 大学など、多数の大学や国立研究機関が HPF コンパイラの大規模な開発を行っているのに対して、日本では、このような活動が非常に少ない。一方でベンダは、安定したシステムの開発が最優先のため、新規機能の開拓や、効果が不明な機能への挑戦は優先度が下がる傾向にある。これらの補完には、大学や国立研究機関によるコンパイラ技術の研究開発を活性化する必要がある。

また、MPI と同様に、HPF を利用した並列プログラミング経験の蓄積、並列プログラミング技術者、並列プログラミング支援ツールなどの研究開発も重要である。1996 年に JAHPF (Japan Association for High Performance Fortran) [7] というプライベートな検討会が結成され、HPF の機能拡張、実用コードの並列化/評価、ベンチマークプログラム開発、日本語ドキュメントの整備などが行われている。このような活動をさらに広げていく必要がある。

### (3) 共有メモリ用並列化指示行インタフェース

分散メモリマシンに比べて共有メモリマシン上の並列化は、データの分散配置を考慮する必要がないため簡単である。構造化され、並列化対象のはっきりしたプログラムなら、自動並列化で十分な並列化効果が得られる場合もある。しかし、一般の場合には、並列化ループの選択、粒度の調整、リダクション演算の認識、手続き呼び出しを含むループの並列化、DO Across 型ループの並列化、同期の最適化、共有データ/ローカルデータの切り分け、スケジューリングなどをコンパイラがすべて効率良く行うことは困難である。そこで、指示行の形式でこれらの最適化をコンパイラに指示するための言語拡張が、共有メモリ用並列化指示行インタフェースである。CRAY、SX、Alliant、Convex といったベクトル並列マシンを対象に開発された「マイクロタスキング」が商用機では最も成功した例である。また、1991 年に ANSI の X3H5 で Parallel Fortran [8] の標準仕様案が作成されたが、当時 MPP が並列マシンの主流だったこともあり、正式標準として成立しなかった。しかし最近になって、SMP と呼ばれる RISC ベースの並列マシンが多く製品化されるに伴い、OpenMP (<http://www.openmp.org/>) という仕様がデファクト標準を目指して検討されている。

OpenMP は SGI や IBM、KAI (Kuck & Associates Inc.) などが中心となってフォーラムを

結成し、共有メモリ用並列化指示行インタフェースの標準化を目指しているもので、最初の仕様案がまとめられ、すでに KAI などから OpenMP の処理系(トランスレータ)が製品化されている。

また、共有メモリマシンには、DSM(Distributed Shared Memory)と呼ばれる方式を採用するシステムが増えている。論理的には共有メモリであるが、物理的にはデータの格納場所によってアクセス速度に差が生じるシステムである。このようなシステムにおいては、高い性能を得るためにデータアクセスローカリティの抽出が重要である。DSM 用の言語機能、コンパイル方式の研究も今後重要性が増すであろう。

#### (4) スレッド並列化記述インタフェース

スレッド並列化記述インタフェースとは、複数の手続きを別々のスレッドで非同期に実行することにより並列処理を実現する枠組みである。並列化制御ライブラリを用いた Fork ~Join のプログラミングスタイルとなり、共有メモリマシン用のインタフェースである。ある意味では、データ転送を共有メモリアクセスと同期制御で行うメッセージパッシングととらえることもできる。Unix 上では、pthread ライブラリがこのインタフェースの代表である。また、CRAY や SX などの共有メモリベクトル並列マシンでも「マクロタスキング」として、同様の機能がサポートされている。

#### (5) 自動並列化インタフェース

現状の自動並列化コンパイラは、共有メモリマシンを対象とするものが中心であり、前記(3)の共有メモリ用並列化指示行インタフェースの発展形ととらえることができる。現在の技術レベルでは、ユーザが全く指示行を挿入しない場合の自動並列化の適用範囲は、かなり狭い。(3)でも述べたが、並列化ループの選択や、DOALL 以外のループの並列化、手続き呼び出しを含むループの並列化その他の要因により、コンパイラにすべての判断を任せて効率の良いプログラムを生成できるケースが限られるからである。また、分散メモリシステムにおいては、データレイアウトの自動化がネックとなり、自動並列化コンパイラの実現が共有メモリマシンを対象とするものに比べて一層困難になっている。

### 3.2.3.2 ベタフロップスマシンと言語関連機能

前節において、科学技術計算分野における並列言語処理系の種々のインタフェースを概観した。ベタフロップスマシンが、1 万プロセッサを超える高並列システムとなり、メモリアーキテクチャが分散メモリ形態になると仮定すると、これらのうちで(1)のメッセー

ジパッシングまたは(2)のデータパラレル言語のどちらかのインタフェースを採用せざるを得ない。また、ペタフロップスマシンが、SMP クラスタのように共有メモリシステムをノードとして、これを多数結合したような、階層構造を持つことも考えられる。このような場合には、(3)、(4)、(5)の共有メモリ用の並列化パラダイムもノード内並列化の手段として重要となる。この際には、分散メモリ～共有メモリの階層構造をもつ並列化パラダイムの統合も重要な課題となる。

また、ペタフロップスマシン上の言語関連機能としては、コンパイラだけでなく、ツールが重要である。並列プログラミング、チューニング、デバッグなどは、逐次システムを対象とするのに比べるとはるかに複雑で困難であるからである。

今後5年から10年で重要となる研究課題は以下のものが挙げられる。

- ・ MPI 技術のさらなる研磨 (性能チューニング、MPI2.0、ツール整備)
- ・ HPF 実用化 (言語仕様の改良、適用範囲拡大、安定した処理系の開発、コンパイラ最適化強化)
- ・ 応用プログラムの並列化の実践と言語インタフェースへのフィードバック
- ・ 種々の並列化インタフェースの統合
- ・ 並列化ツール (並列化支援、チューニング支援、プロファイラ、デバッガなど)
- ・ 並列アルゴリズムの研究と、特定アルゴリズムに特化した並列化支援

### 3.2.3.3 米国の関連プロジェクト

並列化記述言語については、これまで常に米国が主導的役割を果たしてきた。3.2.3.1で述べた標準言語インタフェースについても MPI、HPF、OpenMP など、いずれも米国メンバを中心として結成されたフォーラムで仕様が検討されている。並列化ツールに関する、Ptools コンソーシアム(<http://www.ptools.org/>)についても同様である。この要因を分析してみると、以下のようにまとめられる。

- ・ 産・官・学が一体となって活動している。
- ・ 実応用を持つ HPC ユーザが活動に多く貢献している。MPI などは、どちらかといえば、ユーザのコミュニティーが中心となって組織されている。
- ・ 世界に対して、完全にオープンな環境で活動が行われている。
- ・ CRPC(Center for Research on Parallel Computation)や DOE などの、これら活動への安定したスポンサリング。
- ・ 活動の直接的、間接的結果が論文化されることによる参加研究者の積極的参画。
- ・ 活動が実用性を重視。また主要ユーザの参画により、ベンダが成果を無視できない。

この他に、実応用を見据えて並列化技術の振興を図るプロジェクトや活動として、以下のものがある。

(1) NHSE(National HPCC Software Exchange: <http://www.nhse.org/>)

米国 CRPC が中心となり、HPC 分野のソフトウェア技術の振興を目的として、世界中の並列処理関連ソフトウェアの収集と配布を行っている。また、関連プロジェクトのホームページへのリンクも充実させている。

(2) Caltech Concurrent Computation Program

(<http://www.netlib.org/utk/lsi/pcwLSI/text/BOOK.html>)

Geoffrey Fox 教授が中心となって、Caltech で開発された並列プログラムを以下の 5 つに類型化し、それぞれに必要な並列化パターン、並列アルゴリズム、HPF や MPI での記述性などを論じている。

- ・ Synchronous Applications
- ・ Loosely Synchronous Applications
- ・ Embarrassingly Parallel Applications
- ・ Asynchronous Problems
- ・ Metaproblems

(3) Enabling Technologies for Peta(FL)OPS Computing ワークショップ

60 人以上の並列処理の種々の領域の専門家が 3 日間、ペタフロップスコンピューティングの実現に必要な技術をサーベイしたもの。応用、デバイス、アーキテクチャ、ソフトウェアそれぞれについてワーキンググループに別れて議論し、報告書をまとめている。( <http://nhse.npac.syr.edu/roadmap/petaflops/peta.html> )

(4) PTOOL コンソーシアム (<http://www.ptools.org/>)

並列化ツール技術の振興を目的として設立されたコンソーシアム。並列化ツールとして必要な要件の洗い出しや、ポータブルな並列化ツールの開発、技術交流などを行っている。

### 3.2.3.4 我が国で行うべき施策

前節で、米国の並列化ソフトウェア関連研究を概観した。これらを踏まえて、我が国の並列化ソフトウェア関連のプロジェクトのあるべき姿を考えると、以下の要件が重要であると思われる。

#### (1) 応用指向であること

米国のプロジェクトは日本に比べて非常に応用指向である。並列化の対象とすべきアプリケーションを入念に検討し、これらの効率の良い並列化の実現を目標として、並列言語インタフェース、コンパイラ、ツールなど並列化ソフトウェアの研究開発活動を行っている。また、こうして開発された並列化ソフトウェアが実応用の並列化に使われ、利用者に並列化技術を蓄積するとともに、再度結果がソフトウェアの開発にフィードバックされる。このような良い研究の循環を生み出すことが重要である。

#### (2) 世界に開けたプロジェクトであること

米国の活動は、非常にオープンな形で行われている。HPF、MPIなどの言語標準作成活動は、米国政府がスポンサーとなって研究者の旅費や会議場の提供、成果の出版など種々の支援をしているが、一方で、欧州や日本からの研究者の参加を許している。しかも、研究成果の世界への公開にも積極的である。このことが、世界のすぐれた研究成果を取り込み、しかも世界の関連研究を先導していく原動力になっているように思える。

#### (3) 産・官・学がバランス良く参加したプロジェクトであること

米国では、大学、国立研究所の研究者が中心となってプロジェクトを推進し、これにベンダが積極的に参加する形のプロジェクトが多い。このため、基礎理論中心の研究にならずに、しかも新たな課題に挑戦的に取り組むことができる。

#### (4) 複数のプロジェクト間の協調

米国の非常に多数のプロジェクトが並行して実行されているが、これらが HPCC という枠組みの中で方向づけされ、相互に密に連携しているように見える。大学や国立研究所の研究者たちの仲介により、研究成果がタイムリーに他のプロジェクトに利用されている。また、CRPC による大学連合、NASA、LLNL、LANL、ORNL などの国立研究機関がプロジェクト間連携に大きな役割を果たしているように思う。

## (5) 議論の場

米国では、フォーラムや、ワークショップの形態で全米の研究者が集まって深く議論する場が多い。たとえば、ホテルや会議場に3日程度泊まり込んで、絞り込んだテーマを深く議論し、その結果を報告書としてまとめる機会が多く設けられている。日本でも、このような活動を広く展開すべきであろう。

プロジェクトを成功させるためには、研究テーマの選定もさることながら、横と縦の連携が重要である。ここで横とは、産・官・学のことであり、縦は、応用からデバイスにいたる研究領域のことである。また、研究の成功には応用指向で研究の目的をはっきりと絞り込むことが重要である。日本でも最近、このようなプロジェクトが増えてきた。RWCPで研究されているPCクラスタ、航技研のNWT、原研モンテカルロ装置、筑波大学CP-PACS、地球シミュレータなどは、応用を絞り込んだプロジェクトの良い例である。また、PDCや本ペタフロップスマシン技術調査ワーキンググループなど、研究交流を行う機会や、今後の研究のあり方について討議する場も以前に比べると増えてきているように思う。今後も、このような流れをさらに継続するとともに、プロジェクト間の連携、方向づけをさらに強化していくことが重要であると考えられる。

## 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] OpenMP Architecture Review Board, OpenMP: A Proposed Industry Standard API for Shared Memory Programming, Oct. 1997.
- [4] MPI Forum, MPI-2: Extensions to the Message-Passing Interface, July 1997.
- [5] William Gropp and Ewing Lusk, User's guide for mpich, a portable implementation of MPI. Technical Report ANL-96/6, Argonne National Laboratory, 1994.
- [6] High Performance Fortran Forum: High Performance Fortran language specification, version 2.0 (1996)

[7] Yoshiki Seo, Hiroshi Ohta, Hidetoshi Iwashita, Kenji Suehiro, HPF/JA: An Extended HPF for Parallelizing Real-World Numerical Simulations, IWIA Workshop, HAWAII, Oct. 1997.

[8] ANSI X3H5, X3H5 Fortran-77 Language Binding Version 3, July 12, 1991.

引用 URL 一覧

- (1) The Message Passing Interface standard (<http://www.mcs.anl.gov/mpi/>)
- (2) MPICH: A Portable Implementation of MPI (<http://www.mcs.anl.gov/mpi/mpich/>)
- (3) High Performance Fortran Forum (<http://www.crpc.rice.edu/HPFF/>)
- (4) OpenMP (<http://www.openmp.org/>)
- (5) The Parallel Tools Consortium (<http://www.ptools.org/>)
- (6) National HPCC Software Exchange (<http://www.nhse.org/>)
- (7) Caltech Concurrent Computation Program  
(<http://www.netlib.org/utk/lsi/pcwLSI/text/BOOK.html>)
- (8) Enabling Technologies for Peta(FL)OPS Computing  
(<http://nhse.npac.syr.edu/roadmap/petaflops/peta.html>)

### 3.3 アプリケーション

#### 3.3.1 計算科学と産業基盤ソフトウェアの研究開発の課題 (関口智嗣委員)

ここ10年間における計算手法と計算機の両方の著しい進歩は、これまで到底達成不能であったような複雑な計算シミュレーションを可能とし、今や現実の工学的問題においても解析のツールとしてだけでなく、材料・原理探索の強力な武器としてその地位を固めてきている。こうした計算科学ならびに産業基盤ソフトウェアは、大規模科学技術計算として今後とも高性能計算機の第一義的な対象領域であることは間違いない。ペタフロップスという現行の1万倍程度高速な最高性能を持つシステムが登場するとき、開発するための最も重要な原動力としてこれらのアプリケーションが想定されることは間違いない。ここでは、アプリケーション技術の現状とロードマップについて述べる。

##### 3.3.1.1 アプリケーションのキーワード

今後のアプリケーションを高度化するにあたって必要なキーテクノロジーとして重要なものが「マルチスケール計算技術」とか「シームレスカップリング (ズーミング)」と称される複数の計算モデルを融合させるための計算技術である。これは、従来の単独モデルで極限操作を行ったとしてもそのスケールでは扱いきれない現象を対象としている。ひとつの例として地球環境シミュレーションがあり、そこでは「大気循環モデル」「海洋大循環モデル」「極地モデル」「陸モデル」などを総合し、全体として取り扱う必要がある。この問題ではそれぞれに取り扱う時間単位が違うこと、離散化のスケールが違うこと、計算量が異なることから単純にプログラムを足しあわせるだけでは計算を構築できない。個々のモデルを別個に高速システムで実装するだけでは、計算時間の短縮がはかれない。そこで、相互に計算中途のデータを補間・外挿したり、時間の間引きを行うが、これらを統一的に扱うための手法が今求められている。

別の例としては、ミクروسケールとマクروسケールの両方からのアプローチをモデル化し、計算科学的手法により解明する問題領域がある(図1)。この領域をメゾ領域と呼ぶが、この領域以下のスケールにおいては粒子的な取り扱いでモンテカルロ法や分子動力学法によるシミュレーションを行ったり、量子的な取り扱いにより第一原理計算に基づく電子状態の計算を行ったりすることが可能であった。また、メゾ領域以上のスケールにおいては連続体、構造体の取り扱いとして差分法、有限要素法によるマクロ量として計算することが可能であった。ミクروسケールからの延長上でメゾスケールを取り扱くと、系の粒子数

を2桁以上増加させる必要があるが、計算時間、メモリ量との関係で実際の計算技術としては困難を伴う。一方、連続体からの極限的な分割による近似では系のメッシュ数などを増加させる必要があり、これも計算時間、精度、メモリ量の関係から一般には取り扱いが困難である。こうしたメゾ領域はマクロな性質が現れるための素現象の過程を見るために非常に重要であり、破壊や変形などの材料的な性質を決定しているとも考えられている。このためマクロないしミクロの領域からのアプローチによる計算手法を高密度に連携させ、より高精度かつ高速に計算する要請がある。計算技術的な観点から見ると、それぞれのアプローチで実行に用いられる高性能化技術は異なっており、単純には統合することができない。このために新たな計算手法のモデル化と適した計算機システムの開発が重要となる。

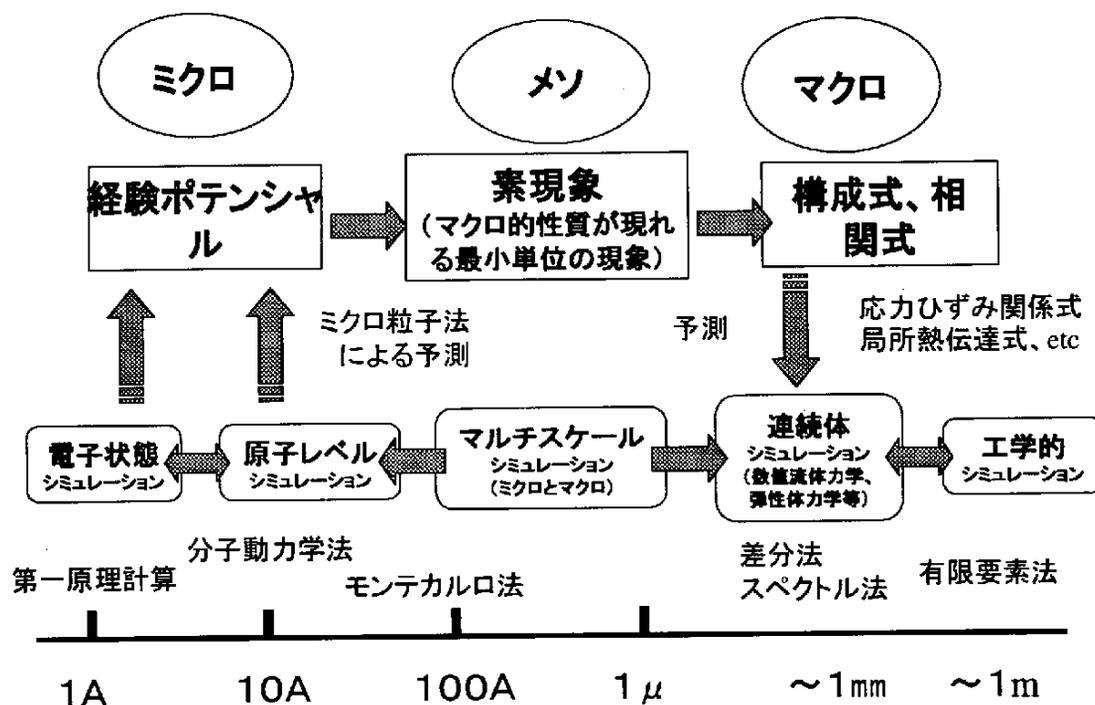


図1 物理・化学のマルチスケールモデリング

### 3.3.1.2 ベタフロップス用計算アルゴリズムの開発

従来の計算科学での計算手法の代表的なものとして、古典分子動力学法と、量子力学的電子状態計算に基づく計算がある。それぞれは、ほぼ独立して研究が進められてきた。古典分子動力学法は、系内の原子について古典力学的に運動を求める方法であり、熱力学的

性質や動的性質の研究には適しており、現在では100万個の原子を扱うことも可能である。しかし、この方法の最大の欠点は、解析的な原子間のポテンシャルを用いるために、原子間の結合が生成されたり切断されたりするような過程を正しく記述することが困難であることである。また、電子が直接に関与するような現象については全く無力であり、その意味で化学反応を扱うことには無理がある。一方、量子力学的電子状態計算に基づく計算は、古典分子動力学法と全く相補的である。正確な記述が可能であること、電子が関与する現象を扱うことが可能であることなどの利点がある一方、計算が複雑であるために大規模の系を扱うことは困難である。

計算アルゴリズム開発の例としてマルチスケールな物理・化学の問題を考える。ここでは電子が関与する過程は、系の一部分で生じており、系の残りの部分はその中心部を構造的に支えていたり、電場を及ぼしたりしているに過ぎない。このような場合には、古典分子動力学法だけでは問題を扱うことはできないが、量子力学的手法で全系を扱うことも計算量が膨大になってしまって不可能になる。そこで、その活性部位は量子力学的に扱い、残りの部分を古典的に扱うという融合的な計算手法が確立されれば、このような従来は困難であったような問題が扱えることになる。メゾスケール問題は現実の世界には、酵素反応、触媒反応、破壊、粒界、複合材料など非常に多く存在している。

この計算手法により試行錯誤を減らすことがトータルコストの低減につながる。例えば触媒の開発と改良は、化学産業の効率化に寄与するだけでなく、環境・エネルギー問題としても重要である。酵素反応の解明も同様である。計算科学的手法による触媒作用や酵素反応の原子レベルでの解明は精力的に進められており、この方向の研究に大きい寄与をすることが必要である。従ってこうした研究は情報産業のみならず材料関連、環境関連、エネルギー関連新規産業創造に貢献することが期待されている。また、これからの計算機システムを中心となるであろう専用計算システムや並列計算システムの融合手法は、情報産業にも重要な寄与をする。

さらにいくつかの応用例を記載する：

- 蛋白質のダイナミクス：酵素反応においては、その一部の反応中心では、化学結合の切断・再結合が起きており、量子力学的計算が必要である、その周辺部分および蛋白質を囲んでいる溶媒については、古典分子動力学で十分である。
- 材料の破壊：実際に破壊が生じる部分では原子間の結合の組み替えや切断が生じるので、正確な記述には量子力学的計算が必要である。一方、その周辺での系の変形は破壊現象に重要ではあるが、古典的な取り扱いで十分であろう。粒界や、接合面についても同じように扱えるであろう。
- 複合材料：材料機能の発現は材料構成要素の一部が主に受け持っており、他の部分は

構造を支えていたり、熱の伝達を受け持っているというような場合が多々ある。従って、ここでも古典的扱いと量子的扱いの融合が有効である。

- 溶液内での化学反応：反応する分子とそのごく周辺の溶媒分子については、量子的扱いが必要であるが、より離れたところにある溶媒分子は古典的な扱いで十分である。

### 3.3.1.3 各省庁施策における位置付け等

科学技術会議諮問第25号「未来を拓く情報科学技術の戦略的な推進方策の在り方について」において高性能コンピュータとネットワークを駆使した新たな地のフロンティアを目指した研究開発の重要性が指摘されている。また科学技術会議諮問第14号「物質・材料科学技術に関する開発基本計画について」に対する答申のフォローアップ報告書でも計算機シミュレーションの基盤技術の重要性が謳われている。

大規模な計算科学技術が、材料設計や環境・エネルギー問題などの現実世界の広い分野において、今後ますます重要な役割を果たすことが期待されている。材料設計では、試行錯誤を回避することによるコストの低減、環境・安全問題などでは、シミュレーションによる危険なために実際の実験が不可能な過程の解明などが典型的な例である。しかしながら、これらの問題に対応するには、少なくとも数百万個の原子についての古典力学的な運動と量子力学的な電子の振る舞いの組み合わせさった複雑な現象を解明する必要があり、従来の計算手法では対応しきれないことが多い。こうした現状を打破するには、計算機システムとアルゴリズムの両面から効率的な計算手法の開発が必要である。すなわち専用計算システムと並列計算システムの融合と、古典シミュレーションと量子シミュレーションの融合が必要である。この目的のためには、単一の分野のグループでは不可能であり、参画研究機関それぞれの分野の専門家の持っている高いポテンシャルを融合することで、初めて上記の手法を開発できる。こうした研究開発は、広範囲の問題の解決を可能とするために必須であると考えられる。

### 3.3.1.4 今後の課題例

大規模な、いわゆるグランドチャレンジ的な問題はすでにその研究の緒に付いているが、それに加え、今後は新たな展開が期待される。例えば工業技術院が保有するアプリケーションには、超伝導現象の理論的解明、新機能材料創製を目指した結晶成長シミュレーション、微細構造を持つ新デバイス設計、慣性核融合プラズマ解析用シミュレーション、生体複雑系の振舞いの解明、遺伝子情報の解析など、基礎理論から実用的なシミュレーション

を実施する項目だけではなく、並列・分散手法による高性能アルゴリズム、産業基盤ソフトウェアの開発や複雑な計算機システムの性能を解析評価するという情報技術としてスーパーコンピュータを活用するような超高速計算機利用技術がある。これらの計算手法としては分子動力学、分子軌道法、差分法による対流・拡散・伝熱計算、粒子シミュレーション、変分モンテカルロ計算、巨大行列の対角化、大規模連立一次方程式の解法などが主に用いられている。

これらのアプリケーションからは速度、メモリといった点への要求は際限がない。スーパーコンピュータは、並列処理機能を具備するのがすでに標準的な構成となっており、今後のシステムも、より台数の多いシステムとなるであろう。しかし、安易に並列化のみに依存して最高性能をたたき出すシステムは、現行のアーキテクチャでは単体での性能が制限されるため、大きく飛躍することは困難となる。並列化による高速化をプログラムの上では努力、工夫はしているものの、多数ユーザが同時に使用する共同利用センターでは並列化よりも逐次プログラムによる最適化と高速化が実行時間の短縮には大きく貢献する。したがって、単体性能が極力高く、かつ可能な限り並列化が実現できるようなシステムの導入に期待する。また、メモリも単に容量ではなく、高速なアクセス速度を持ち、かつ演算装置に十分なデータ供給が可能となるアーキテクチャを期待する。最低でも1,000Gflops、256GB、かつ単体性能が10Gflops以上という能力と容量が今後3年程度を見越した場合には必要であると考ええる。ベタフロップスはそのまた先にある。

### 3.3.2 大規模並列処理に期待のかかる応用について (中島克人委員)

従来の大規模並列応用は科学技術計算分野に限定されて来た。しかし、高性能ハードウェアの低廉化により、ビジネス分野や設計・製造などの産業応用分野にも並列処理が浸透しつつある。

ここでは、ビジネス分野で近年重要な位置を占めつつある情報系のデータベース応用システムに着目し、そこで用いられる OLAP(Online Analytical Processing) やデータマイニング等のための処理方式と、設計・製造分野での重要な情報処理技術である最適化のための探索手法の2分野を取り上げ、その並列処理の規模やスケーラビリティに関する展望・考察を述べる。

また、これらの応用での処理内容の考察に基づき、大規模並列マシンのアーキテクチャに対しての要件を述べる。

そして最後に、大規模並列処理技術の発展と実用化のために国の推進すべき施策について、私見を述べる。

#### 3.3.2.1 データベース応用システム

ディスクの低廉化と大容量化、および、企業の情報化の進展により、大量のデータが蓄積されるようになり、これを情報系データベースとして有効活用することを目的として、大容量データを扱う技術が近年脚光を浴びるようになった。

大容量データ活用技術は、その処理の複雑度に応じて表1のように分類出来よう。これを並列処理の観点から整理する。ただし、近年爆発的に利用が進んでいるインターネット上のWebページ等をデータベースに見たてた分散型のデータベース応用システムは除外し、ローカルなデータベースに対する並列処理のみを対象とする。なお、表中の「計算の濃度」とは、ディスクへの入出力とプロセッサ内での計算量の比を相対的に表わしたものである。

表1 データベース応用システムにおける処理の内容と特徴

	計算の並列度	計算の濃度 (*1)	基本処理例
(1)データ検索・集計	小	薄い	選択・合計
(2)データ統計・解析	大	中位	平均・分散・散布集計
(3)データマイニング	大	中位	相関・分類・クラスタリング
(4)データ利用プランニング	大	濃い	探索・事例ベース

(\*1) ディスクアクセスに対する計算量の比

### (1) データ検索・集計

情報検索における特定のキーワードからの文書検索、データベースにおけるインデクスを用意していない属性を用いたレコード検索、集計作業などは、ディスクの走査が少なくとも1回必要となる。このため、検索の並列度は $O(N) \sim O(\log(N))$ 程度得られるものの、ディスクアクセスの所でどうしても直列化が発生してしまい、ボトルネックとなる。一方、直列化を軽減すべくディスクを並列化すると、当然ながら管理システムを並列化する必要性が生じる。並列データベース管理システムを例にとると、その効率的な実現には様々な課題がある。例えば、通常のデータベース関連の応用プログラムはディスクアクセスが多いため、プログラムを丸ごと分配しないと効率が悪くなる。この時、エラー制御のプログラムでラップするなどしてから分配しないと、システムがエラー制御を行いにくくなるという問題点が知られている。いずれにせよ、複数の検索・集計を同時に行うとしても、計算能力に対する要求はそれほど大きくないと言える。

### (2) データ統計・解析

種々の集計結果を求める処理であるが、OLAP と呼ばれる対話性・瞬時性をセールスポイントとした技術が近年注目を集めている。あらゆる属性に関する集計処理をリアルタイムで求めるのは今のハードウェア技術では流石に困難であることから、予め主な属性に関する集計処理を行い、これを多次元テーブルもしくはグラフに表示出来るような形式（多次元 DB 形式または多次元データキューブと称する）に変換してメモリに格納しておく前処理（ビューの実体化とも称する）と、ユーザの指示に従い、それを3次元表示するための処理がある[1]。

勿論、前処理されていない集計結果を求められた場合には、実体化済みのデータキューブの情報の組合せもしくは元のデータベースから即座に集計することも要求される。

前処理の代表的なものは2属性または3属性に関する頻度分布表（もしくは散布図）を求めることであるが、属性数の多いデータの場合、その組合せは膨大となり、全ての頻度分布表を前処理で求めておくことは現実的ではない。従って、計算の手間およびメモリの容量を勘案し、予め参照されそうな組合せについてのみ前処理で求めておき、それ以外のはオンデマンドで作成することになる。

プロセッサの高速化とメモリの大容量化により、より多くの属性組合せに関して前処理（実体化）を済ませておくことが出来るようになったこと、そして、グラフィック表示技術の進歩により3次元散布図等のリアルタイム表示が容易になったことが、OLAP の実現・普及に結び付いている。

データの大容量化と更なる高速化への要求から、並列処理への期待が高く、しかも、上

記のように前処理においては高い並列度を有する。並列化においては多数のディスク上にレコードを分散配置し、それぞれのプロセッサで出来るだけ独立して集計を行う。ただし、プロセッサごとの小計値の報告のために通信が必要となる。分散メモリ型計算機においては通信量とメモリ効率はトレードオフの関係になる。即ち、それぞれのプロセッサで集計し、最後に一回だけ小計値報告をするためには、頻度分布表のコピーを全てのプロセッサで保持しなくてはならなくなるため、メモリ効率が悪い。頻度分布表のコピーを持たず、逐一オリジナルの頻度分布表を持つプロセッサに報告すると通信が膨大となる。

リレーション（テーブル）を縦割りにしてディスクに分散配置しておき、特定属性の値が1つのプロセッサで取り出せるようにする等の工夫も考えられる。これにより、一部の頻度分布表作成は完全にプロセッサ単位に独立に行える。しかし、組み合わせる属性がプロセッサ/ディスク境界をまたぐこともあり、属性単位の通信が避けられない。

なお、1回のディスクの走査で複数の頻度分布表作成を行うことにより、ディスクアクセスのボトルネックを回避できるため[2]、スケーラビリティの確保には(1)と異なり通信性能がより重要となる。

### (3) データマイニング

大量データからの知識発見をデータマイニングと称する。OLAPと同様、ユーザの意志決定支援のツールとして用いられるため、迅速な応答性が要求される。

データマイニングには、属性間の相関発見、クラスタリング、分類・予測等の種々の目的にそって様々な手法があり、用いられる手法や処理の内容は多岐に渡るが、いずれもOLAP以上に計算負荷が高い。また、データ蓄積量が増大しても応答性能を確保したいため、スケーラビリティを持った並列処理方式が求められる。

ここでは、最近注目を集めている相関ルール発見の代表的アルゴリズムである apriori アルゴリズムを見てみよう[3]。基本的には属性値（アイテム）同士の共起頻度のカウントであるため、属性数ではなく、各属性の取り得る値同士の組合せ数だけの計算量があり、本質的な並列度も高い。あるサポート値（出現頻度比率）以上のアイテムの組（ラージアイテムセットと称する）を効率良く発見するため、 $k$  個組のラージアイテムセットの候補は  $(k-1)$  個組のラージアイテムセットから生成し、各候補が所与のサポート値を超えるかどうかのカウントを行う。即ち、組数の増加に関しては逐次であり、各組数におけるカウントのみが並列処理の対象となる。ただし、多くの実用問題では2～3個組のラージアイテムセットを求める処理が計算の殆どを占めるため、並列処理による速度向上が大いに望める。

$k$  個組のカウントの並列処理手法として、候補アイテムセットのカウントをハッシュに

より分散配置し、各レコードから得られたk個組に対応するカウントアップ信号の送り先もやはりハッシュにより求める方法が知られる[4]。データ自身が移動することもなく、カウンタの複製も不要のためメモリ効率も良いが、カウントアップ信号の通信頻度が高く、また、一部のプロセッサに通信が集中して負荷のアンバランスを来す恐れがある。これを解消するため、頻度の高そうな候補アイテムセットに関してのみ全プロセッサで別々にカウントするという工夫も提案されている[4]。この他に様々な並列化手法が提案されつつある[5]が、数十プロセッサ以上での並列処理はこれからの課題である。

数値属性間の相関発見の場合は、意味のある相関ルールとするために、離散化、即ち、連続値の区間割りを行わなくてはならないが、たとえば、あるサポート値以上となるように区間割りを自動決定する場合は、一種の最適値探索問題となる。2つの数値属性を条件部に持つ「 $(XY) \in R \rightarrow Z$ 」という相関ルールを求めたい場合は、 $XY$ の2次元平面上の領域決定問題となり、計算量は更に増大する[1]。候補領域をピクセルという微小領域に分割し、それらのマージを行う手法の場合は、ピクセル数に比例した並列度があるが、実際の並列化は今後の研究成果を待つことになる。

#### (4) データ利用プランニング

データマイニングでは「知識発見」とは言うものの、仮説を生成・提示するだけに留まり、その採用・棄却の判断や知識としての利用はユーザに任される場合が多い。即ち、ユーザの意志決定を側面支援するものが殆んどである。

そこで、マイニングされた仮説を知識として蓄積し、それに基づきユーザの行動をプランニングする次の段階が考えられる。データマイニングとの境界ははっきりしたものではないが、例えば、「売り上げの最大化を目的に、クラスタリングされた顧客情報に基づいて宣伝投資の配分を立案する」などは、データ利用プランニングと称して良いであろう。また、(3)で触れた連続値属性の自動区間割りや領域決定も利用目的によってはこの範疇に入れても良いであろう。前述のように、このようなプランニング問題では次節の最適解探索技術を必要とするものが多い。

事例ベースを活用したシステムも考えられる。例えば、適切な過去の事例を提示して立案に代えたり、評価の高い類似事例を組み合わせることも考えられる。事例ベースの場合は基本的には事例数だけの並列度が存在する。分類を目的としたものではあるが、事例ベースの並列処理研究も進められており、16台で15.8倍の台数効果が報告されている[6]。

## 3.3.2.2 組合せ最適解の探索問題

組合せ最適化問題に分類される多くの実用問題では、多目的問題であったり、目的関数が厳密に定義出来ない等の理由により、実際には厳密な最適解を求めることよりも、準最適解を高速に求めることが要求される。そのため、準最適解の求解速度の優れたモダンヒューリスティックス探索が近年脚光を浴び、種々の実用問題でも成功を収め始めている。

組合せ最適解の厳密探索解法の殆んどは指数オーダーの手間がかかるため、N台のプロセッサによる並列処理によって計算オーダー $O(N)$ を削減しても、その効果は低いと言えたが、計算の手間を大幅に省略して実用時間で求解するモダンヒューリスティックス探索においては事情が異なり、並列処理によるN倍の効率向上に期待がかかる。設計分野などでは開発スピードが製品としての成功を決定づけるため、経済効果が大きいからである。

探索木を用いた探索解法の並列化では、通常、探索木の分割という形で行われるが、探索木の深さや枝の広がりアンバランスによる粒度の調整、プロセッサ稼働率確保のための動的負荷分散、これを制御するための通信のボトルネック解消等、様々な課題を解決せねばならず[7]、十分なスケーラビリティを確保するのはなかなか困難と見られた。しかし、モダンヒューリスティックス探索の中でも特に多点型の確率的な探索では、探索範囲をN倍にすることが比較的容易であり、これにより最適解もしくはある閾値を越える満足解の求まる確率をN倍にすることも不可能ではない。

遺伝的アルゴリズムにおける交配のような局所探索のための操作を、もし大域的な情報を元に行う場合には、N倍あった探索範囲を縮めてしまう。そして、プロセッサ同士でのいわゆる重複探索の可能性を高めてしまうため、探索範囲の再拡大等の難しい課題が生じる。しかし、パラメータ数が十分に大きい場合には重複探索の回避はそれほど困難ではない。

例えば多点ランダムウォーク探索という単純な方法においても、歩き出す方向にちょっとした工夫を施すことによって重複探索が殆んど生じないように出来る。そこでは、近傍点をランダムに一定数選択し、その中の最良点へと探索を進める方法をとる。近傍点をランダムに取ることにより、複数の探索プロセスが、万一交差したとしても、同じ航跡をたどる確率を十分低く抑えることができ、探索範囲の広さが保証される。これはタブー探索を実現するために共有メモリに保持すべき短期経路記憶を省略し得ることを示し、大規模並列処理に大変適した性質と言える。遺伝的アルゴリズムにおいても個体集団の多様性が確保される「島モデル」は大規模並列処理向きと言えるであろう[8]。最適解もしくは局所最適解に漸近するための「局所探索」と「多様性もしくは探索空間の維持」のバランスの取り方に技術課題は残るものの、探索手法とそれを元にした応用問題は大規模な並列処

理が活用出来る重要な領域と言える。

### 3.3.2.3 応用から見た大規模並列マシンアーキテクチャ

上記で見た情報系データベース関連の応用ではディスクへの並列アクセスが前提となるが、殆んどはレコード走査、即ち、逐次アクセスであるため、強力なプリフェッチ機能を持つディスクキャッシュが効果的である。また、(2)の頻度分布表作成や(3)の相関発見などでは多数の、あるいは、大きなテーブルを用いたカウント処理が行われるため、従来のキャッシュメモリではヒット率が高くないので、何らかの工夫が必要である。更に、多くのカウントアルゴリズムでは、小計の通信先である最終集計プロセッサをハッシュで求めることにしており、この部分のハードウェアサポートも検討に値する。

ディスクの走査回数を削減するために複数の頻度分布表を同時作成する場合にはレコード情報をブロードキャストする方法が結局は有効であるという報告もあり[2]、また、相関発見の並列化でもラージアイテムセットのブロードキャストが必要であることから、ハードウェアによる高速なブロードキャスト通信機構も有用であるかも知れない。

また、多くのアルゴリズムにおいてメモリ容量と通信量とにトレードオフが存在する。この性質は従来の科学技術計算ではあまり考慮され無かったため、現状の商用並列計算機では価格性能比の良いシステムを実現することが困難かも知れない。

一方、3.3.2.2の「探索問題」における多点探索アルゴリズムでは入出力スループットやメモリ容量に対する要求は大きくない。従って、従来の科学技術計算を想定した並列計算機でも高い性能とスケーラビリティが期待出来る。ただし、タプル探索における長期記憶、即ち、極大値の比較等のためにブロードキャストや効率の良いグローバル演算がサポートされている必要はありそうである。

### 3.3.2.4 国の推進すべき施策について

昔から言われていることであるが、応用サイドの者にとって、並列処理は高速化の手段であって目的ではない。そのため、単一プロセッサの実行速度の著しい進展が続く中、それでも並列処理による高速化を敢えて目指すのはスケーラビリティをセールスポイントとしている場合か、ぎりぎりの競争を行っている場合である。

今回調査検討した情報系データベース応用システム分野ではスケーラビリティが非常に重要であることから、応用サイドからの並列処理研究が例外的に進展したとも言えるが、この分野でさえ、並列計算機アーキテクチャとして、これを十分考慮したものはまだ出現

していない。

最適解探索やシミュレーションをベースにした設計ツールの高速化は開発速度の向上に非常に重要であるにも関わらず、それらの並列処理化研究は、少なくともわが国では盛んではない。利用者側からは設計ツールの並列化に手が出せないか、手が回らないという状況である。

応用からアーキテクチャまでの幅広い知識を必要とする並列処理技術は学際的であり、かつ、基礎的研究開発テーマと言える。故にその開発は、直近の開発に追われる産業界にとって大きな負担となっている。有能な人材を比較的多く抱えている大企業の研究所でさえ、景気低迷・競争激化のために、中長期的な研究から新製品の直接開発へウェートをシフトしている。この研究から開発へのシフトは日本企業同士の競争激化を増進すれども、基礎技術に裏打ちされた米国等の製品を凌駕することにはならず、国際競争力の長期的な維持・向上の観点から心配が多い。

並列処理技術をはじめとする、学際的かつ中長期的な研究を企業にモチベートするための速効性のある国の施策としては、研究開発に対する思いきった税法上の優遇措置が考えられる。また、必要な人材を素早く組織化できるように、人材の流動性を高めるための施策や、産学共同の支援施策なども拡大すべきである。学校教育の見直しも必要であろう。複数専門制を推進するなどにより、学際研究を促進し得る人材の早急な育成が望まれる。

## 参考文献

- [1] 森下 真一: データマイニングと最適化技術について, 信学技報, AI97-50, DE97-83, pp.31-38, Dec. 1997.
- [2] 松澤 裕史, 福田 剛志: 複数の集約計算のための並列アルゴリズム, 信学技報, AI97-54, DE97-87, pp.57-62, Dec. 1997.
- [3] R. Agrawal and R. Srikant: Fast Algorithms for Mining Association Rules, In Proceedings of the 20th VLDB Conference, pp.487-499, Sept. 1994.
- [4] 新谷 隆彦, 喜連川 優: データマイニングにおける相関関係抽出の並列処理方式, 並列処理シンポジウム JSPP'96, pp.97-104, 1996.
- [5] 喜連川 優: データマイニングにおける相関ルール抽出技法, 人工知能学会誌, Vol.12, No.4, pp.513-520, July 1997.
- [6] 松本 和宏, 前田 一穂, 柳沼 義典, 池坂 守夫, 萩原 純一, 隠田 博之: 並列データマイニングアーキテクチャ, 信学技報, AI97-39, DE97-72, pp.33-38, Dec.

1997.

[7] M.Furuichi, N.Ichiyoshi and K.Taki:A Multi-Level Load Balancing Scheme for OR-Parallel Exhaustive Search Programs on the Multi-PSI, In Proceedings of the Second ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP), 1990.

[8] E.Cantu-Paz:A Survey of Parallel Genetic Algorithms, IlliGAL Report No.97003, Illinois Genetic Algorithms Lab., May 1997.

### 3.3.3 アプリケーションの過去・現在・未来

——未来に向かってやるべきこと —— (福井義成委員)

#### 3.3.3.1 はじめに

これまでのアプリケーションの大きな流れから未来の方向を予測する。各アプリケーションで詳細は異なるであろうが、大枠は次のようになっている。

- ・ 1次元            ---> 2次元    ---> 3次元
- ・ 線形            ---> 非線型
- ・ 定常解          ---> 非定常解 (時間的変化)
- ・ 1ケースの計算 ---> 多くのケースの計算 (パラメトリックスタディ、最適化)
- ・ 粗いモデル      ---> 詳細なモデル
- ・ 計算能力の向上により、それまでは不可能であった計算方法・新分野が出現する

また、計算機を使って問題を解くという原点に戻ると、計算そのものだけではなく、モデル化の問題、アルゴリズム、計算結果を正しく認識させる技術も重要である。基本的なモデリングについては、企業では難しいため、大学や政府の研究所での研究が必要である (図1 計算機で問題を解く手順)。

日米比較では多く (ほとんど) のアプリケーションが欧米製であり、日本はお寒い状況である。上記のモデリングについても、米国、欧州での研究に遅れをとっている。米国の大学の場合、モデルを開発し、それをアプリケーションにのせるまでの仕事を行っている。日本でもそこまでやれるような体制 (仕事の評価まで含めて) が必要である。また、すべての分野で欧米と対抗することは不可能であり、日本が得意とする分野で自主性を発揮すべきである。

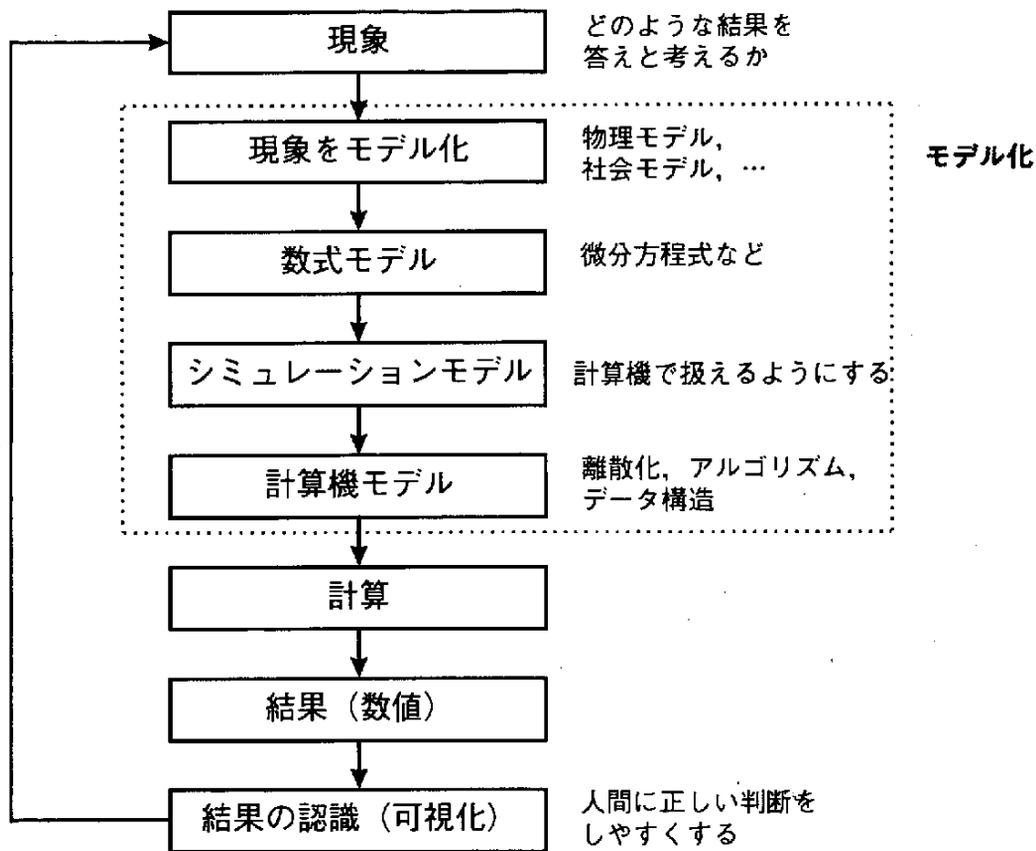


図1 計算機で問題を解く手順

### 3.3.3.2 これまでのアプリケーション

電子計算機（コンピュータ）のアプリケーションは、新たな有用な道具が提供されたことにより、その当時の先進的な科学者、技術者により、ただちに色々な分野への適用の努力が始まった。今のプログラム電卓よりも劣る計算機ではあったが、その当時としては画期的なことであった。いま、科学技術分野に限ってみても、流体、構造、回路、デバイス、材料、化学、天文、気象、地震、プラズマ物理、・・・と計算機が利用されている分野は非常に多く、計算機のアプリケーションの全分野に渡って述べることは不可能なことである。ここでは、あえて独断と偏見に陥ることを恐れず、筆者の関係してきた分野での限られた範囲でアプリケーションの方向と問題点について述べることにする。以下でいくつかの例について述べる。

## (1) 共役傾斜法 (CG法) の変遷

例えば、科学技術計算で重要な位置を占めている連立一次方程式の解法の1つである共役傾斜法の変遷をたどってみる。連立一次方程式の解法の場合、ガウスの消去法のような直接法は有限回の演算で計算が終了する。しかし、反復法の場合、条件の良い場合は少ない反復回数で収束するが、条件が悪い場合は反復回数が非常に多くなってしまふ。普通、反復法の場合、有限回の演算で計算が終了する保証がない。共役傾斜法が出現したころ、反復法で有限回で収束と評判になった。しかし、色々な問題に適用してみたところ、有限回の演算で収束しないケースがあることが判明した。共役傾斜法が有限回の演算で終了するためには、無限桁の計算が必要である。そのため、一時、下火になった。その後、解くべき行列に前処理を施すことにより、収束性が大幅に改善された場合があることが分かり、前処理+共役傾斜法で復活した。偏微分方程式を離散化して得られる行列に前処理を施すことは、もとの問題の性質を利用して、大幅な収束性の改善を得ていることになっている。その結果、計算速度も速くなった。

これは、問題の範囲を限定し、問題の性質を利用することにより、解法の能率を向上させている。いわば、解法 (アルゴリズム) の専用化の方向である。

## (2) 制御系のシミュレーション (連続系シミュレーション言語)

制御系ではモデルをブロックダイアグラムで表現することが多い。ブロックダイアグラムでは機能ボックスを接続するだけで制御モデルを構築することができる (図2 ブロックダイアグラムの例)。いわばオブジェクト指向あるいは関数型的にモデルを構築することができる (非手続き的な表現になっている)。ブロックダイアグラムの結線はアナログ計算機の配線 (デジタル計算機のプログラミング) に対応している。計算機が出現したころは、ブロックダイアグラムで表現されたモデルも離散化し、アセンブラ、FORTRANなどで手続き的に計算されていた。その後、IBM 7040/7090の上で問題向き言語 (連続系シミュレーション言語と呼ばれる) DSL-40/90が開発され、非手続き的に計算をすることが可能となった。連続系シミュレーション言語はその後、CSMP-III (IBM 360/370 シリーズ)、CSPL-6 (ACOS-6)、ACSL (多くの機種で稼動) 等に発展している。モデルの変更が多い分野では、手続き的表現に比較して、大幅 (2~3桁) な時間節約になる。さらに、最近では機能ブロックのアイコンを選択して、機能ブロック同志を結線することにより、モデルを構成できるアプリケーションが出現している。これは、図形的入力 (ブロックダイアグラム) への回帰ということができる。

この例は、間違いにくい記述方法、高いレベルでのモデル記述の重要性を意味している。

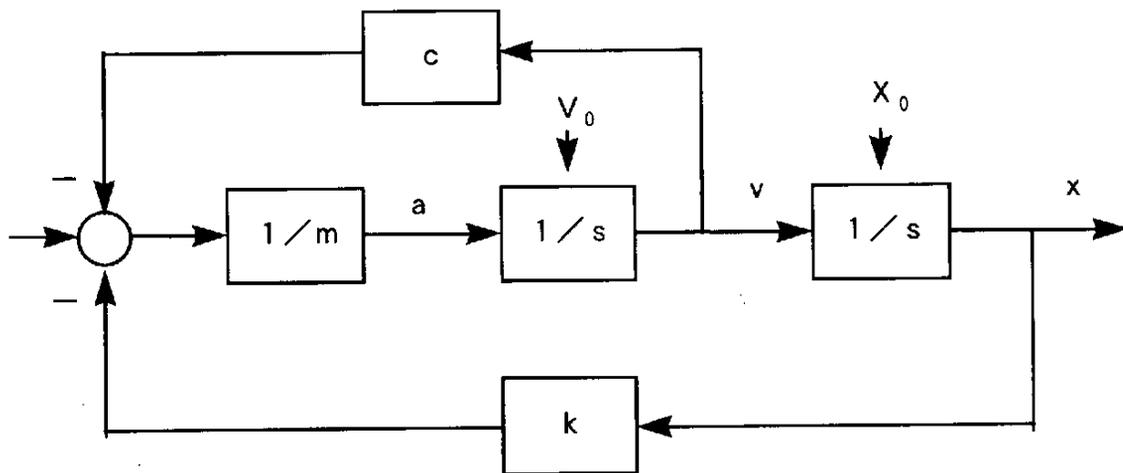


図2 ブロックダイアグラムの例

### (3) モデルの詳細化

モデルの詳細化という面から考える。計算機の性能が低かった時代には、もとの問題をそのまま解くのではなく、何等かの形でパラメタライズされた方法で問題を扱っていた。例えば、半導体の中で、電子の移動度を導入することにより、電子の平均的な動作は扱うことができるが、電子と原子核の相互作用のような効果は扱うことができない。しかし、移動度という概念を導入することにより、計算量を大幅に削減でき、外からの電圧・電流関係が必要な場合は十分な近似になっている。

また、半導体のデバイスシミュレーションは、内部動作に関する電子の数がまだ非常に多いので、主流のソフトでは電子などを流体モデルで近似して、計算を行っている。しかし、構造が微細化してくると、内部動作に関する電子の数が少なくなり、流体モデルでは実際の現象を十分に近似できなくなる。実際の現象を表現するためには、個々の電子レベルでのモデルを構築する必要がある（粒子モデル）。

最近の材料の設計などでは、第一原理からの分子・原子相互の関係から結果を求めることが行われている。このような計算は、非常に大きな計算能力を必要とするが、ある仮定を入れた計算では現れることのない効果を予測することも可能となる。

今後は、研究の高度化や設計の最適化を行うためには、より詳細なモデルを必要とする場面が多くなる。

### (4) 次元の変遷

電子計算機（コンピュータ）の多くのアプリケーションは偏微分方程式を数値的に解くという観点から始まっている。偏微分方程式を離散化して解く場合、1次元から始まり、2次元、3次元と進んでいる。今後も、詳細な結果が必要な分野では3次元の取り扱いが

不可欠となる。3次元の場合、各次元の分割数を増やすと必要となるメモリーが急激に多くなる（3乗に比例）。多くのメモリーを実装した計算機システムを如何に効率的に開発するかが問題となる。

(5) 線形モデルから非線型モデルへ

近似モデルも線形で済んでいた（我慢していた）時代から、非線型が必要な（扱える）時代へと変化している。線形モデルの場合、ある1ケースの結果が分かれば、入力異なる計算結果も全て予測可能となる。しかし、非線形モデルの場合は、入力と出力が比例関係にはないので、入力異なると、その入力に対しての計算を再度行わなければならない。非線形計算は1回だけでも線形計算よりも計算量が多いが、その上に、入力と出力の比例関係がないので、より多くの計算が必要になる。

(6) 時間の扱い

時間的な面も定常解ですんでいた（我慢していた）時代から、非定常解が必要な（扱える）時代へと変化している。

(7) パラメトリックスタディ、最適化

計算機の能力が低かった時代には、1ケースだけでも結果が得られることが大切であったが、計算機の能力が大きくなるにしたがって、多くのケースの計算が可能になり、研究・開発の最適化が可能になる。パラメトリックスタディ、最適化、統計的な解析が可能となり、1ケースの計算に比べて多くの知見を与えてくれる。

これからは、人口の減少の時代を迎えると、研究者・技術者の不足という時代となり、多少無駄な計算をしても、人間の時間を減らす方向に進むであろう（計算機に金を投資し、時間をお金で買う時代となろう）。

(8) 問題解決の重要度の変化

計算の重要部分も変化している。計算機を使って問題を解くということを考えると図1（計算機で問題を解く手順）のようになる。計算機が出現された当初は計算すること自身が時間がかかり、最大の問題点であった。計算機が高速化し、計算手法も高度化することにより、問題解決のフローの中で計算の占める比重が段々低下している。自動車工業での車体の強度解析（衝突解析）では、計算よりもメッシュ作成の比重のほうが大きくなっている。計算は数日あれば終了するが、メッシュ作成には数ヶ月が必要になっている。このような部分にも計算機パワーを利用して短縮することも重要になっている。計算の高速化

により、モデル作成（メッシュ）に比重が移る。今後は、ますます、問題解決の全体の時間が重要になり、問題解決の手順の中で最も時間の掛かる部分を速くすることが重要となる（モグラたたき）。そのためには、数値的計算だけではなく、数式処理、人工知能等の非数値的処理も組み合わせて、お互いの良いところを利用し、問題解決の全体の時間を短縮することが重要となる。いわば、問題解決のシームレス化であろう。

### 3.3.3.3 これからのアプリケーション

今後は、前節のように、多くのポイントでシミュレーションの高度化が進展し、それに応じてより大きな計算能力が必要とされる時代になろう。

計算機的能力が大きくなるにしたがい、より良い結果が得られるようになるであろうが、その結果が現実の研究・開発・設計に与える影響は見かけ上、計算機的能力向上と比例しないと予測できる。これまで、計算機的能力・計算手法の進歩により、段々と詳細で精度の良い結果が得られてきているが、その効果はある閾値を越えたところで変革を起こしているようである。計算機的能力が段々大きくなるに従い、定性的結果だけであったものが、定量的結果が得られるようになってきた。定性的結果でも研究・開発・設計の方向を決める知見となり、定量的結果が得られるようになるとシミュレーション段階での最適化が可能となる。まず、定性的には正しい結果が得られるようになった時点で第1の変革が起り、その後、定量的に正しい結果が得られるようになった時点で第2の変革が起きるであろう。現在は、定性的に正しい結果は多くの分野で得られるようになり、定量的に正しい結果が得られる分野が出現してきたところであろう。これからの課題は、まだ、定性的結果も得られていない非常に難しい分野でのモデル化やアルゴリズムの確立と、定性的結果が得られている分野でも定量的結果を得られるような努力であろう。定性的結果も得られていない分野はまだ多く存在し、五感に依存している分野の技術化であろう。更に、第六感が必要といわれている分野も解明したいものである。

また、計算機的能力が飛躍的に向上すれば、これまで不可能と思われていた分野、手法も実現可能となり、新たな応用分野が必ず出現するはずである。一方で、使い易さの点も重要であり、飛躍的に向上した計算機的能力を計算以外の部分に振り分けることも重要である。

### 3.3.3.4 並列化について

今後の高性能の計算機を構成するには並列化は不可欠である。そのため、並列化を推進するには、現在、高価な並列計算機の価格を安くする必要がある。並列計算機を普及させて、価格を低減させることが重要である。そのためには並列化が行い易い分野から、並列化を行うべきである。パラメトリックスタディなどは並列化しやすい分野である。現実的には、疎粒度の問題から手をつけることも必要である。(これは細粒度の並列化を否定しているのではない。並列処理を普及させると細粒度の並列化がより重要になる。まずは並列処理の普及という意見である。)

### 3.3.3.5 要望

政府の支援は下記のような基礎的な分野での研究投資が重要である。

- ・ 計算機を構成する素子の基礎
- ・ 基本的なアーキテクチャ
- ・ 大規模計算のためのアルゴリズム (数値計算を含む)
- ・ 詳細な現象を表現できるようなモデル化技術 (物理的面が大きい)

また、シミュレーション分野ではソフトの重要性が大きく、それも理論的なものだけではだめで、実際に利用可能なレベルまでのソフトの開発が重要である。しかし、日本では大学等でのソフトに対する評価がされていないのが問題である。米国との比較においても最大の問題である。

技術開発にあたり、米国優位となる日米の差はソフト開発に対する評価と新しいソフトに対する先取性の有無であろう。これは、これまで日本が欧米に追いつくことだけに重点をおいてきたことの弊害であろう。

日本が米国に対等もしくは優位の部分の1つは回路シミュレーションのエンジン部分である。ベクトル計算機でのベクトル化は日本では約10年前に3社で成功したが、米国ではやっと1~2年前に成功した状況である。

これまで米国に任せて、日本が手をつけていない分野はほとんどの分野であるとも言える。これまで、日本では手をつけたといっても、欧米の後追いが多く、自主的に手がけた分野は残念ながら少ない。

### 3.3.4 計算科学のための超高速計算機開発に向けて (横川三津夫委員)

科学技術分野の「数値シミュレーション」という言葉には、「誰もが扱うことのできる計算機は、理論解析を補強する道具または実験解析を行う道具であり、道具以上のものではない」という計算機を利用した研究に対する負のイメージが含まれていたように感じられる。

しかし、ここ数年のスーパーコンピュータの飛躍的發展、及び超並列計算機の出現等によって、数値シミュレーション技術を駆使すれば未知現象の解明や実験不可能な現象の解明さえも可能になると考えられており、数値シミュレーションに基づく研究手段が「計算科学」という一つの研究分野を形成しつつある。理論、実験につぐ「第3の科学」と称する計算科学は、さまざまな研究対象について、横断的かつ普遍的な研究手段が提供できるように考えられている。

この過程において、数値シミュレーションに対する負のイメージは払拭された。しかし、計算科学の現状を考えると、理論、実験科学に対抗する手段に到っていない。計算科学がさらに発展するためには、現状よりもさらに高い計算能力を有する超高速計算機が必要なることは明らかである。本稿では、計算科学をより発展させるための超高速計算機開発について考えてみたい。

#### 3.3.4.1 テラ、ペタフロップス級の計算機は大衆のツールか？

本ペタフロップスマシン技術調査ワーキンググループの平成8年度の報告書「ペタフロップスマシン技術に関する調査報告」においては、「将来のペタフロップス級のコンピュータは超並列計算機にならざるを得ない」という共通の認識がまとめられている。計算科学に用いられる超高速計算機は、当然分散メモリ型並列計算機であろう。しかしながら、ここ1年間を見ても、分散メモリ型並列計算機の利用技術はさほど進展しておらず、依然として利用技術の研究に留まっており、応用分野の研究者自身が一人で使いこなせる状況にない。これは、単一プロセッサや共有メモリ型計算機などの従来の計算機と同様に、分散メモリ型並列計算機についても広範囲のユーザに提供することを目標にしているからではないだろうか？

並列計算というプログラミングモデルの変革は、従来の計算機利用技術と根本的に異なる技術を要求している。すなわち、従来の逐次計算機やベクトル計算機などの単一プロセッサにおいては、数値アルゴリズム等の研究がフォンノイマン型計算機による処理モデル

に基づいており、局所的な処理の流れ及びデータの流に注目する局所プログラミング技術によって計算機の高効率な利用ができる。一旦単一プロセッサ用の計算技術が開発されれば、普通のソフトウェア技術者がその計算技術を利用し高度な応用ソフトウェアを開発することができた。

しかし、分散メモリ型並列計算機上の並列計算では、局所的プログラミング技術だけでは高い実効性能が得られず、応用ソフトウェア全体を考慮した大域的プログラミング技術が要求される。プログラムの構文的な解析だけでは分散メモリ型並列計算機で高い実効性能を得ることは困難であり、応用ソフトウェアを開発する上で数値アルゴリズムの選択のフェーズ、さらには物理モデル化のフェーズに遡った応用分野の数値シミュレーションの意味を知る必要が生じ、ソフトウェア技術者のレベルでは手に負えない領域となってしまった。

例として、分散メモリ型並列計算機用の Fortran プログラムのための並列数値計算ライブラリを利用する場合の問題点を指摘しよう。

数値シミュレーションでは、そこで用いられる数値アルゴリズム及びプログラミング方法によって、計算機での実効性能が大きく左右される。したがって、数値アルゴリズム開発、及びプログラミング方法の研究が重要性を増していった。特定の計算機アーキテクチャ上で極限までその計算性能を引き出すプログラミング能力は、一種の職人芸（匠）である。このため、数値シミュレーションを研究手段としている多くの研究者（ユーザ）は、数値アルゴリズムの開発、あるいは特定の計算機アーキテクチャ上での効率よいプログラミング方法の開発を、数値解析の研究者、数値アルゴリズムの研究者に委ね、ユーザは数値計算ライブラリとして使用することを是としていた。

数値計算ライブラリでは、さまざまな数値アルゴリズムをユーザが簡単に使用できるインタフェースをもつ Fortran のサブルーチンとしてユーザに提供している。ユーザはサブルーチンを使用する場合、サブルーチンで指定されるデータ構造の規定を守るだけで良く、また必要であれば作業用領域（array）も局所的に確保できる。すなわち、単一プロセッサ上の Fortran 言語を使った応用ソフトウェアでは、呼び出し側のサブルーチン内で局所的に変数宣言を行うことができ、応用ソフトウェア全体の構造を大幅に変更する必要がない。ユーザ作成の応用ソフトウェアにおけるデータ構造とサブルーチンが指定するデータ構造が異なる場合には、ライブラリを呼び出す直前にデータ構造の入れ替え処理を簡単に追加でき、その修正だけで高い実効性能の数値計算ライブラリが使用可能である。仮にさらに実効性能のよい数値アルゴリズムが開発されたとしてもライブラリとして提供されれば、呼び出し部分を変更するだけで良く、この変更作業は機械的に行うことができる。つ

まり、ユーザに対して使用上の強い制約を課していなかったと云える。

しかし、分散メモリ型並列計算機上の応用ソフトウェアを開発する場合や単一プロセッサ用の応用ソフトウェアを並列化しようとする場合には、局所的なプログラム構造の把握では性能向上が望めず、プログラム全体を大域的に把握し、データ分割による再配置と並列実行可能な処理の抽出を検討しなければならない。特に、並列数値計算ライブラリを用いる場合、実効性能の高い応用ソフトウェアにするためにはデータ配置が重要なポイントであり、並列数値計算ライブラリのデータ構造と応用ソフトウェアのデータ構造の整合性をとる必要がある。このため、並列数値計算ライブラリを利用する上で、ユーザが作成するプログラムのデータ構造にかなり強い制約が課され、応用ソフトウェアの局所的な変更だけでは対応しきれず、一般のユーザが利用可能な理想的なライブラリにはなり得ない。共有/分散メモリ型並列計算機のようにメモリ階層をもつ計算機を使用する時の応用ソフトウェアのデータ配置問題も深刻である。したがって、一般のユーザが対処できる局所的なプログラム構造の変更では高い実効性能を得ることはできない。

並列計算機の利用技術が進展しないという問題は、「並列計算機をパソコンと同様に誰もが使えるものにならなければならない。」との呪縛から脱却することによって解決できる。すなわち、テラ、ペタフロップス級の超高速並列計算機を用いた大規模並列数値シミュレーションを行うユーザはほんの一握りであり、従来の計算機と同様に一般ユーザが超高速並列計算機を使いこなせる必要はない。超高速並列計算機は、高度な専門性を必要とする計算科学のための実験装置であるとの認識を持ち、ユーザによる実験装置の利用では、それを熟知している研究者と共同で大規模数値シミュレーションを行うことが効率的であると思われる。例えば、粒子加速器のような大規模な実験施設を運用しているのは、高度な専門知識を有する研究者と施設を熟知した研究者及び技術者であろう。超高速並列計算機も同様に考えれば気が楽になる。この装置を設置した共同利用施設を作り、物理モデルや数値アルゴリズムレベルで応用ソフトウェアの開発能力をもつ並列応用ソフトウェア研究者をそこに配置し、計算科学研究者との密接な関係の下で優れた応用ソフトウェアを開発できる体制を構築したらどうだろうか。ただし、このような高度な知識が要求されるソフトウェア研究者の仕事を民間のソフトウェア技術者に求めることは困難であり、また社会的需要の少ない計算科学分野のソフトウェア技術者の育成、維持を民間に求めることは得策ではない。

### 3.3.4.2 実験装置として超高速並列計算機

ここでは、大規模な実験装置である超高速並列計算機の開発目的について考えることにしよう。

米国においては、既に ASCI プロジェクトにおいて、スーパーコンピュータ開発のための潤沢な国家予算が計算機産業界に投入されている。しかし、このプロジェクトの主目的は、スーパーコンピュータ開発そのものではなく、それを利用した応用分野の課題解決である。1998年2月3日米国ロスアラモス国立研究所におけるクリントン米国大統領のスピーチにおいても、ASCI 計画において開発されたスーパーコンピュータは、包括的核実験禁止条約の下での核兵器のシミュレーションに限らず、地球温暖化の影響予測や省力化エンジン等の開発にも利用されるとの認識を示した。このための 30Tflops のスーパーコンピュータを 2001 年までに 517 百万ドルを投入して開発することになっており、さらに 2004 年までに 100Tflops のスーパーコンピュータの開発が計画されている。ここで開発されたプロセッサ技術は、民生品としてすぐに利用可能であることが特徴である。

日本においては、科学技術庁が推進している「地球シミュレータ」計画において、地球変動予測を行うための応用ソフトウェアの開発とともに、ベクトル計算機を要素計算機とした地球環境シミュレーション専用の分散メモリ型並列計算機を平成 13 年度に完成させることになっており、その要求仕様は実効性能 5Tflops 以上、主記憶 4TB 以上としている。この計画の主目的も高速計算機の開発ではなく、地球上の複雑な諸現象に対する大規模かつ高精度な数値シミュレーションによる地球変動の解明にある。たとえば、局所的な気象現象では高分解能モデルが必要であり、その計算のためには現在のスーパーコンピュータの計算能力でも不十分であるため、より高性能な計算機の開発が求められている。この計画の推進により、地球変動に関する理解が飛躍的に進展するものと期待される。

以上の状況のように、科学技術計算のための超高速並列計算機は、既に応用分野に対する専用計算機という位置づけで開発されている。過去、国主導の計算機開発は計算機産業の育成に重点がおかれていたが、今後次世代の超高速並列計算機開発においては、計算機産業の維持、発展のために開発資金を投入するのではなく、ある特定応用分野の計算科学技術を発展させるための専用計算機開発を国主導で実施することが肝要ではないだろうか。その開発過程において半導体製造プロセスの高度化や実装技術の高度化、普及品としてのプロセッサ開発や計算機開発への波及効果が間接的效果として期待される。

超高速計算機が必要な応用分野の選択過程においては、そのニーズに関するさまざまな応用分野の研究者へのヒアリングも必要であるが、むしろ超高速並列計算機を用いた数値シミュレーションが社会に与えるインパクトを評価することが重要である。すなわち、数値シミュレーションによる実質的な開発コスト削減率等について正確に把握すべきである。たとえば、ボーイング社の航空機製造では、開発コストの削減を目的に、設計に必要な大量のパラメータサーベイを数値シミュレーションによって行い、模型を用いた風洞実験に替えることにより大幅なコスト削減を達成したとの事実がある。産業界において数値シミュレーションを適用した場合のコスト削減の具体例を調査し、そのコスト削減分を計算機開発に投入すれば、かなり大規模な超高速並列計算機が開発できるのではないだろうか。今後、超高速並列計算機開発のために大量の国家予算を投入するとすれば、実験代替をした場合の損益分岐点を明確にすべきであろう。もちろん、数値シミュレーションによって得られる成果は、一般大衆に直に反映され利益をもたらす応用分野のものである必要がある。

運用においては、巨額のコストを投入して開発された超高速並列計算機は一極集中の共用実験装置とするべきである。ネットワークコンピューティング技術はここ数年で格段に進歩することが期待され、開発された計算機を分散して設置することも考えられるが、マシンタイムの確保など全体の計算機資源を占有する運用上の調整が困難となり、全体の計算性能を発揮できなくなる可能性があるためである。共同利用施設には並列応用ソフトウェア研究者を配置し、計算科学の研究者と連携を取りながら、大規模な数値シミュレーションを実施する利用体制が望ましい。

### 3.3.4.3 計算科学発展のために

超高速並列計算機の開発が可能であるとした時、その実験装置の能力を極限まで引き出すことのできる応用ソフトウェア研究者の確保が次の課題である。

日本には計算機開発に対して高い能力があるのに、優秀なソフトウェア研究者、技術者が輩出しにくく、たまたま優れたソフトウェアが開発されても世界的に普及していない。以下の理由が考えられる。

- ・優れたソフトウェア開発者が突出することを社会が容認しない。
- ・ソフトウェア開発の協力体制が欠けている。計算科学で言えば、応用分野の研究者と情報科学分野の研究者の交流が不足している。また、横断的な計算科学の交流がな

- く、計算科学技術の普及の障壁となっている。
- ・開発されたソフトウェアの維持、管理、改良する体制が組織として成立しない。  
また、その作業に携わる研究者、技術者の処遇が悪い。
  - ・日本語の特異性により、世界に向けたソフトウェアのドキュメントの整備が不十分である。
  - ・プレゼンテーション技術不足により世界的なイニシアチブが確保できない。

これらを解決するために省庁間の壁を越えた国としての政策を実施しない限り、日本のソフトウェア技術は育っていかないとと思われる。特に、計算科学の並列応用ソフトウェア研究者の育成を怠れば、超高速並列計算機の需要は見い出せないであろう。また、超高速並列計算機が計算科学のための専用計算機として位置付けられても、それを利用できる応用ソフトウェアの開発、維持体制が整備されて行かなければ開発したハードウェアはただの箱になってしまう恐れがある。

応用ソフトウェアを含む計算機ソフトウェアはある種の芸術品である。したがって、その成熟は一朝一夕に達成されるものではなく、ソフトウェア職人（匠）の待遇を改善し、気長に輩出を待つ以外に手はない。計算科学分野に限らず、安易に高度な自動化ソフトウェアを開発することはソフトウェア職人の衰退をもたらし、ソフトウェア技術が消滅してしまう可能性を秘めている。近い将来並列化のプログラミング技術をもつ人間が無形文化財の指定を受けるなどの状況には陥りたくない。特に、計算科学のための応用ソフトウェア研究者、技術者は、高い専門能力を備えている必要があり、その確保、育成に努力すべきである。給与面、福利厚生面での優遇も効果があると思われる。

最後に本稿における提言を、以下のようにまとめておこう。

- (1) 超高速並列計算機は、一般ユーザにとっても簡単に使用可能であるべきとの幻想は捨て去る。
- (2) 超高速並列計算機（ペタフロップスマシン）は、計算科学のための専用実験装置として開発されるべきである。多額の開発経費が必要ならば世界的な共用実験施設として国際協力によって開発することも考えられる。
- (3) 専門性の高いソフトウェア職人、特に計算科学の応用ソフトウェア研究者を優遇し、養成する体制を構築する必要がある。

## 第4章 技術ロードマップからみた研究開発の方向性

### 4.1 アーキテクチャ

#### 4.1.1 シングルチップマイクロプロセッサのロードマップ (天野英晴委員)

1980年代の後半、マイクロプロセッサはRISCの提案による命令の簡素化によるクロックの向上とパイプライン処理の利用により高速化を果たしたが、90年代になって、命令の動的スケジューリング、複数命令の同時発行、キャッシュの高機能化、投機的実行など、高速化のテクニックはますます発達し、それにつれてマイクロプロセッサの構造はますます複雑化した。複雑化した構造は、それを実装するための半導体の面積を必要としたが、高機能マイクロプロセッサの実装に見合う分の面積は、半導体技術の発達によって常に供給が保証されてきた。このようにして高性能マイクロプロセッサの性能は向上し続けてきたが、1996年に入って、シングルプロセッサとして的高速化テクニックが限界に達する一方、半導体技術の発達が衰えぬペースで続いた結果、半導体の面積を有効に利用できない可能性が生じた。この問題を解決するためのアプローチが2つ提案された。

(1) シングルチップマルチプロセッサ: 巨大な高機能プロセッサに代わって、比較的簡単なRISCプロセッサを複数搭載する。

オンチップマルチプロセッサ、マルチプロセッサチップ、1チップマルチプロセッサと呼ばれる場合もあるが、ここではシングルチップマルチプロセッサと呼ぶことにする。

図1に最も基本的なシングルチップマルチプロセッサの構成を示す。

この構成では、4つのプロセッサが大規模な共有L2キャッシュと共に1チップ上に搭載されている。

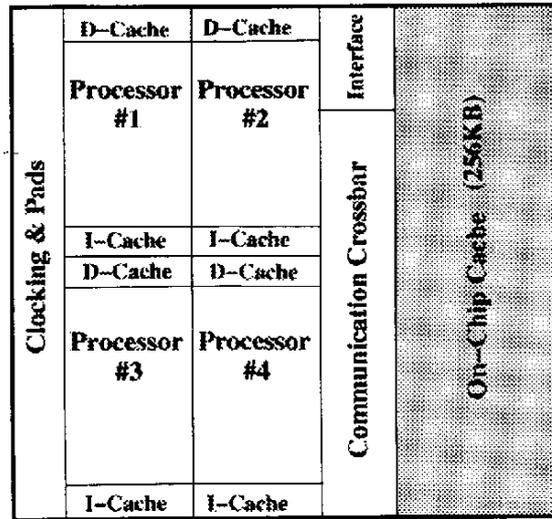


図1 基本的なシングルチップマルチプロセッサの構成

(2) DRAM 混載: 主記憶の一部である Dynamic RAM を同一チップ上に搭載する。

後者の考え方には、チップ外のメモリへのアクセスの壁(Memory Wall)による性能の低下に対する解決法のひとつとしての意味も持っている。

もちろん、この両方を用いる、すなわち、シングルチップマルチプロセッサに DRAM を混載するアプローチも有力である。今回のロードマップは、半導体技術の発展に対して、この2つのアプローチがそれぞれの年代で取り得る形式を検討する。

まず、SIA のロードマップ[1]を参考に、各要素の面積の予測をざっとまとめると表 1 のようになる。

表1 各要素の面積

年	1998	2001	2004	2007
Process	0.25	0.18	0.13	0.10
DRAM ( $\mu m^2/bit$ )	1	0.4	0.2	0.08
SRAM ( $\mu m^2/bit$ )	6	2.4	1.2	0.48
Logic ( $\mu m^2/gate$ )	14	8	4	1.8
Max die area( $mm^2$ )	300	360	430	520

表2 可能となる構成例

年	1998	2001	2004	2007
DT R5000	1+30MB/ 4+0	4+75MB	-	-
DT Alpha	1+20MB/ 2+0	1+88MB/ 2+64MB	4+173MB	-
LT R5000	3+0/1+12MB	1+37MB/ 4+0	4+50MB	-
LT Alpha	1+0	1+23MB	4+0	4+101MB

表1を基に、それぞれの年代で、どのような構成が可能になるかをざっと予測してみた(表2)。予測には、2種類のプロセッサを用いた。比較的軽量の組込み用プロセッサとしてはR5000(3.6Mgates、64KB Cache)を選び、WS/デスクトップPC用のプロセッサとしては、現在最も面積が大きく性能の高いDEC Alpha 21164(9.3Mgates、112KB Cache)を選んだ。シングルチップマルチプロセッサ構成にした場合、キャッシュの要求量が問題だが、問題を簡単化するために、各プロセッサ毎に現在のキャッシュ分だけ持たせると仮定した。すなわち、Single Portの共有キャッシュなら、現在のプロセッサ数倍、プロセッサ数分のポート数を持たせるならば、現在と同量を搭載することになる(これはちょっと少ないかもしれない)。

シングルチップマルチプロセッサ構成は、プロセッサ数は最大4とした。これは現状の技術の延長で汎用目的では、これ以上あっても活用が難しいと考えたためである。

また、ここでは、WS/デスクトップPC用の高性能プロセッサ(DT)と、ラップトップあるいは移動、制御目的用軽量省電力プロセッサ(LT)の2種類を分けて考えた。前者は、予測された最大面積を使うとしたが、後者は、年代にかかわらず150mm<sup>2</sup>とした。年代が進んでも、この辺の面積を越すと価格がぐっとアップするという状況はあまり変わらないのではないかと考えたためである。表中で4+56MBと表記した場合、4プロセッサのシングルチップマルチプロセッサで56MB DRAM搭載可能を示す。表中は、一応バランスが取れているとみなせる場合のみを示し、プロセッサに対して極端にDRAMが少なくなったり、多くなるようなアンバランスな構成になる場合は示していない。

この表自体大変怪しい代物だし、将来Alphaを上回る巨大チップが確実に出現するので、1ランク上のプロセッサに対する予測が必要であるが、少なくとも以下の点を読みとることができる。

(1)余った面積をシングルチップマルチプロセッサにつきこんだ方が有利になる時期は、2回来るだろう。

1回目は、現在から近い未来(1998年)である。この時期は、DRAM混載技術がまだ立ち

上がり段階で、余った面積を DRAM につぎこむと、価格的に有利な面積のチップでは十分な量搭載することができない。これに対し DRAM を混載しないシングルチップマルチプロセッサは、容易に数を増やすことでピーク性能を上げることができる。しかし、シングルチップマルチプロセッサにしても、並列プログラムでピーク性能を上げたり、マルチジョブをするだけでは、商業的に採算は合わないと考えられるので、実験的なシステムに留まる可能性が高い。しかし、この時期に実験システム上で、シングルジョブの高速化技術を確立しておく必要がある。

2 回目は、シングルチッププロセッサ+DRAM 構成にしても面積が余るようになる 2004 年以降である。もちろん Alpha を上回る面積の高性能プロセッサは今後も出てくると思うが、それでも 2004 年を越えると上記の傾向は出てくると考えられる。最初の時期に、シングルジョブの高速化技術をきちんと確立しておけば、これ以降シングルチップマルチプロセッサ+DRAM がマイクロプロセッサの王道になることが期待される。

#### (2) 先にブレークするのは、DRAM 混載、LT 用であろう

DRAM 混載型プロセッサは、①実装面積が小さくなる、②電力消費量が減る、の 2 点で携帯用、移動用、制御用に圧倒的に有利である。DT 用では、これからも面積の大きい強力なチップが登場するであろうから、DRAM 混載がブレークするのは、まず移動用分野で、それが可能になるのは 150mm<sup>2</sup> 程度のサイズで搭載が可能になる 2001 年前後であろう。それから、どんどん DT 用に普及が広がっていくであろう。

#### (3) DRAM 混載型は必然だがシングルチップマルチプロセッサはわからない

前項のように DRAM 混載は、絶対的に有利な点と将来性のある分野を抱えているがシングルチップマルチプロセッサはそうではない。ここ数年のうちにシングルチップマルチプロセッサ上で並列化されていない単一ジョブを高速化する技法が発達することが重要である。NEC の提案する MUSCUT[2]、早稲田大学の笠原研の提案するコンパイラ主導のアプローチ[3]は、有望かどうかは別としてどんどんやらなければならない。2007 年以降、余った面積を使うためにのみシングルチップマルチプロセッサ化が行われるとしたら、その頃は WS/PC 自体が、今のメインフレームのように計算機利用の主体からはずれているのではないと思われる。普及しはじめた SMP の使われ方に注目する必要がある。また、マルチメディア用チップ、FPGA を用いた Custom Computing Machine、通信用チップ等専用チップと考えられているチップに注目する必要がある。

参考文献

- [1] SIA Roadmap, 1997, SIA
- [2] 鳥居、近藤、本村、西、小長谷: "On Chip Multiprocessor 指向制御並列アーキテクチャ MUSCAT" JSPP97 論文集 pp.229-236, 1997.
- [3] W. Ogata, A. Yoshida, M. Okamoto, K. Kimura, H. Kasahara, "Near Fine Grain Parallel Processing without Explicit Synchronization on a Multiprocessor System", Proc. of Sixth Workshop on Compilers for Parallel Computers (Aachen, Germany), Dec. 1996.

#### 4.1.2 「付加的」論理のロードマップ

(中島 浩委員)

表1は、過去6年の代表的なマイクロプロセッサについて；

- $t$ ... トランジスタ数 (100万 Tr 単位)
- $c$ ... オンチップキャッシュ容量 (KB 単位)
- $i$ ... 整数演算ユニット数
- $f$ ... 浮動小数点演算ユニット数

をまとめ、それに基づいて将来動向を予測したものである。1992~94年と1995~97年にはトランジスタ数とキャッシュ容量が約3倍程度増加したのに対し、演算ユニットの数はさほど増加していない。そこでこの傾向を説明するために、「付加的」論理によるトランジスタ消費量  $a$  を、下式によって定義する。

$$a = t - 0.05c - 0.5 - 0.05(i-1) - 0.1(f-1)$$

すなわち；

$$\begin{aligned} \text{キャッシュ1バイトあたりのトランジスタ数} &= 50 \\ \text{「基本的」論理による消費量} &= 50 \text{万} \\ \text{整数演算ユニットの増分あたりの消費量} &= 5 \text{万} \\ \text{浮動小数点演算ユニットの増分あたりの消費量} &= 10 \text{万} \end{aligned}$$

と考える。なお「基本的」論理による消費量は、比較的単純な構成であるR4000のキャッシュを除くトランジスタ数である。

最近の2世代については、この「付加的」トランジスタ数が大きく増加しているが、これは本格的なスーパスカラアーキテクチャの採用、すなわち out-of-order の命令発行や完了、分岐予測や投機的実行といった高度なプロセッサアーキテクチャが採用されたことによるものである。またこのようなアーキテクチャは、3.1.3 で述べた予測技術と密接に関連している。

そこで将来の動向を以下のように予測する。

- $t$  ... SIAによる予測

- *c*... SIAによる製品レベルDRAMのチップあたり容量の1/256ないし1/128
- *i/f*... 3年あたり2倍

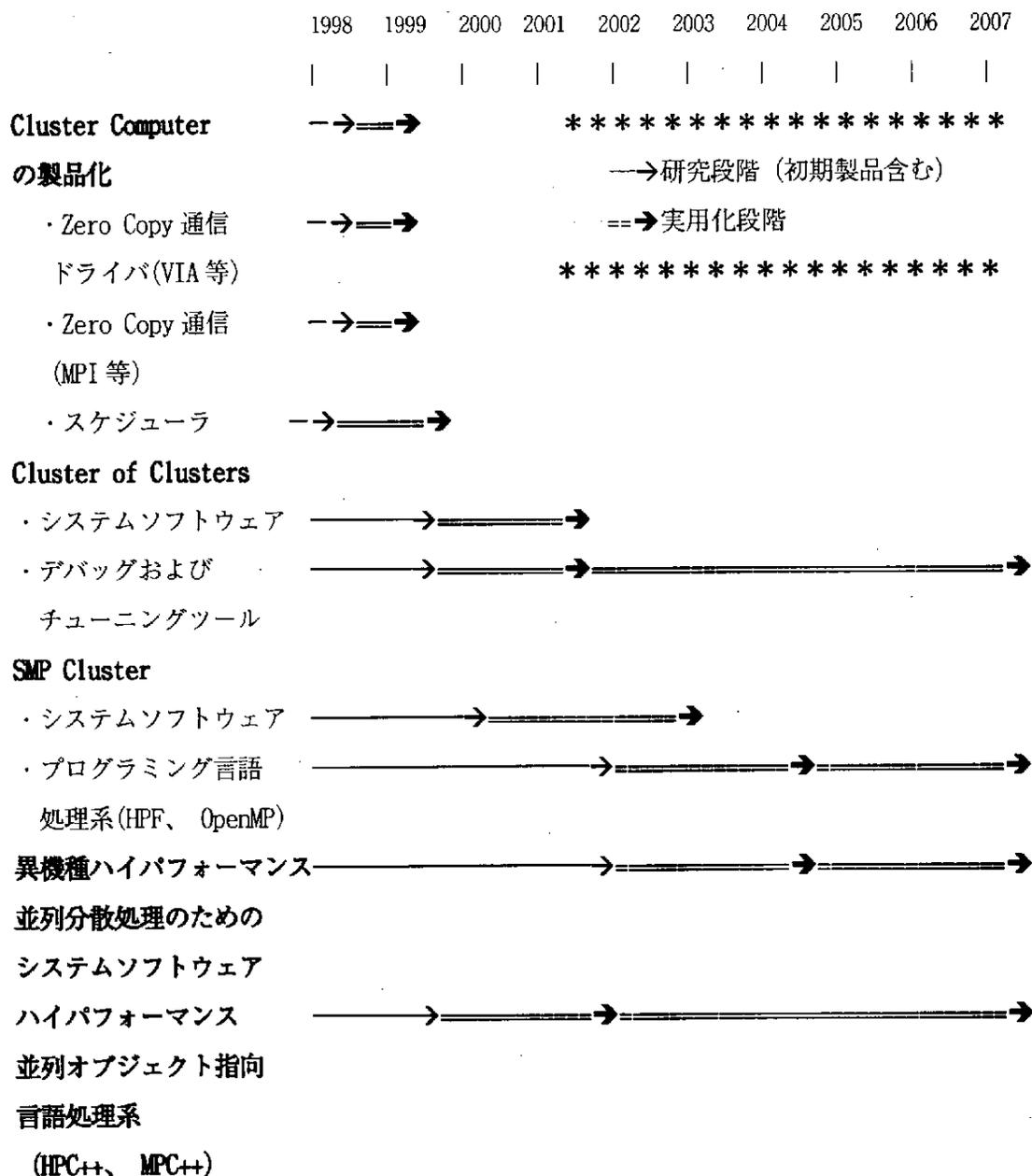
その結果、今後10年間では100万から1000万のオーダーのトランジスタが、「付加的」論理に費やされるという結論が導かれる。すなわち単純に命令を実行する以外の、予測技術などを含めた種々の機構（あるいは「付加的」プロセッサ）の一層の高度化が予測される。

表1 「付加的」論理のロードマップ

year	MPU	<i>t</i>	<i>c</i>	<i>i/f</i>	<i>a</i>
92-94	PA-7100/7200	0.9	0	2/1	0.4
	R4000/4400	1.3	16	1/1	0.0
	21064	1.8	16	1/1	0.5
	PowerPC 601	2.8	32	1/1	0.7
	SuperSPARC	3.1	36	2/1	0.8
	Pentium	3.3	16	2/1	2.0
95-97	PA-8000/8200	3.8	0	2/2	3.2
	UltraSPARC-I/II	5.0	32	2/1	2.9
	PowerPC 604e	5.0	64	2/1	1.3
	R10000	?	64	2/2	?
	Pentium Pro	5.5	16	2/1	4.2
	PowerPC 750	6.4	64	2/1	2.7
	Pentium II	7.5	32	2/1	5.4
	21164	9.6	118	2/1	3.2
98	21264	15.2	128	4/2	8.1
	PA-8500	120?	1536	2/2?	42.6?
98-00	?	20	128-256	4/2	6.5-12.9
01-03	?	60	512-1024	8/4	7.7-33.3
04-07	?	200	2048-4096	16/8	0.0-95.7
08-10	?	520	8192-16384	32/16	0.0-106.9

## 4.2 ソフトウェア

### 4.2.1 分散並列システム (石川 裕委員)



説明

○ Cluster Computer の製品化

既に Compaq は SC'97 において WindowsNT をベースにした PC Cluster の発表をしている。しかし、現在のソフトウェア構成では、性能面において、研究レベルで実証されている通

信性能が達成されていない。1999年度中には、現在の研究レベルの通信性能を持つ製品が出始めるであろう。

○ Cluster of Clusters のためのシステムソフトウェア

Cluster Computer が製品化され広く行き渡るようになると、Clusters をさらに Cluster する話が現実的になってくる。このためのシステムソフトウェアは、既に研究されているが、Cluster Computer が製品化された後の1年半後には製品化されるだろう。

○ Cluster Computer 上のデバッグおよびチューニング環境

技術的に Cluster Computer 上のデバッグおよびチューニングツールは従来の MPP におけるデバッガやチューニングツールと大差はない。特殊なハードウェアの存在を仮定せずに実現する必要があるという点である。デバッガ及びチューニングツールは、ハードウェア、プログラミング言語の進化に応じて進化していく必要がある。

○ SMP Cluster のためのシステムソフトウェア

ここで言う SMP Cluster とは、ノードが SMP 構成のクラスタを指すこととする。SMP 上でのマルチスレッドと他のプロセッサとの通信に関する実行モデルおよび効率良い処理系の開発が必要である。

○ SMP Cluster のためのプログラミング言語処理系 (HPF、OpenMP)

SMP Cluster 上で、マルチスレッドおよび通信プログラミングをしなくても並列化されるプログラミング言語処理系が必要である。HPF や OpenMP の言語仕様を基に開発されるだろう。

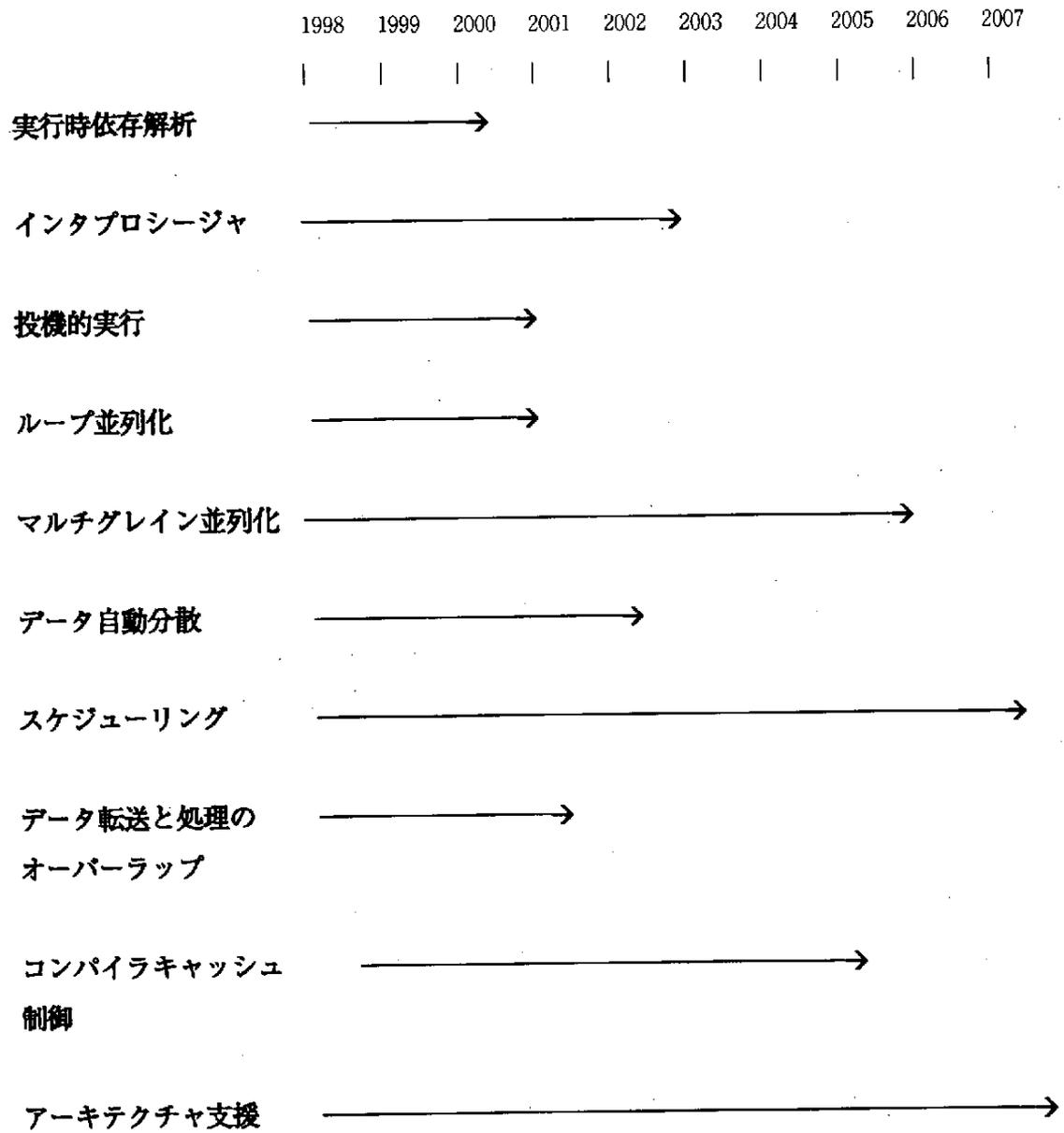
○ 異機種ハイパフォーマンス並列分散処理のためのシステムソフトウェア

プロセッサアーキテクチャ、single processor/SMP、異なるオペレーティングシステム等異機種環境上において並列プログラミングを可能とするシステムソフトウェアの開発。

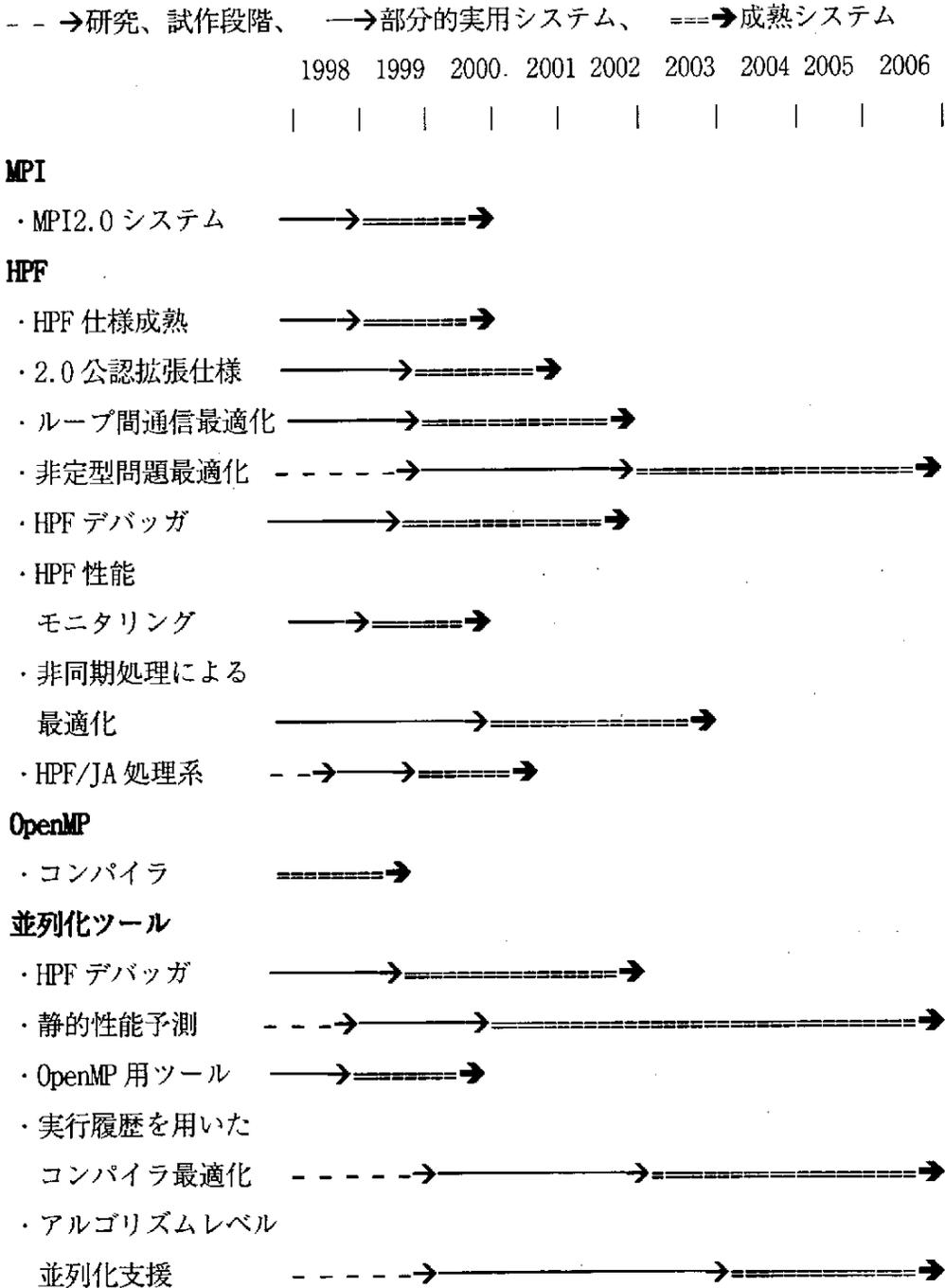
○ ハイパフォーマンス並列オブジェクト指向言語処理系 (HPC++, MPC++)

オブジェクト指向特に C++ の持つ演算子の多重定義機能を用いることにより、可読性に富んだプログラムを記述することができる。さらに、MPC++ のように並列化記述をライブラリに持たせると共にライブラリに特別化した最適化機能を提供する処理系開発が進められている。

#### 4.2.2 自動並列化コンパイラ研究ロードマップ (笠原博徳委員)



4.2.3 並列言語インタフェース/ツール(HPC分野) (妹尾義樹委員)



### 4.3 アプリケーション (福井義成委員)

前にも述べたように、アプリケーションのロードマップを一般的に示すことは、不可能に近い。そこで、ここではアプリケーションを構成する色々な軸をあげ、各軸での方向を示すだけに止める。アプリケーションを構成する軸には

- ・ 1次元 ----> 2次元 ----> 3次元
- ・ 線形 ----> 非線型
- ・ 定常解 ----> 非定常解 (時間的变化)
- ・ 1ケースの計算 ----> 多くのケースの計算 (パラメトリックスタディ、最適化)
- ・ 粗いモデル ----> 詳細なモデル
- ・ 汎用解法 (アルゴリズム) ----> 専用解法
- ・ 低レベルのモデル記述 ----> 高いレベルでのモデル記述 (間違いにくい記述方法へ)
- ・ 問題解決の重要度の移動 (全体の時間を短縮することが重要)
- ・ 計算能力の向上により、それまでは不可能であった計算方法・新分野が出現

が考えられる。ある時点のあるアプリケーションはこれらの要素を軸とする多次元空間上の点として表現される。計算機、計算手法の発達により、アプリケーションの位置は各軸の原点から離れた方向に発展している。どのような位置を移動するかはアプリケーションの性質に依存する。あるアプリケーションは速く3次元化するであろうし、他のアプリケーションは3次元化をする前にモデルの詳細化が先行するというようなことになるであろう (図1)。

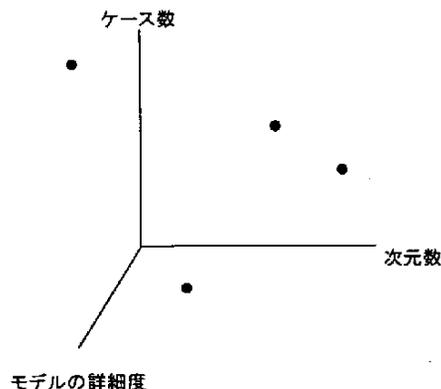


図1 アプリケーションの多次元空間 (3次元の例)

## 第5章 あとがき

本ワーキンググループでの今年度の議論の中心は、ペタフロップスという超高性能計算機の開発に向けた技術やその利用形態を遠くに見ながら、これから5年後程度の情報処理技術を高めるために今何をなすべきかというものであった。本報告書における各委員の見解は、ワーキンググループにおける議論を踏み台にはしているが、昨年と同様に、各委員独自の見解を展開しており、必ずしも整合性がとられているわけではないのでその点をご了承願いたい。また、ある程度意見を整理をするために、各委員の見解をアーキテクチャ関係、ソフトウェア関係、アプリケーション関係という3つに分類して掲載した。ただし、委員の意見はそのような分野にとどまらず、もっと幅広く情報処理技術全般に渡っているものもある。

各委員の見解をもとに、私見を交えて技術課題をまとめてみると以下のようなになる。

### (1) アーキテクチャに関して

- ・プロセッサによるインテルの優位を変革するには、現在の枠組みを変える必要がある。その可能性のあるものとして、Reconfigurable machines、システムオンチップ、メモリ混載プロセッサ、マルチプロセッサオンチップなどがあげられる。
- ・並列処理を効率的に実行するためには、プロセッサの実効性能を高める必要があり、そのための技術として、計算を高い精度で予測する技術が必要。

### (2) ソフトウェア（システム技術を含む）に関して

- ・システムオンチップに関する産業のポテンシャルを高めるために、CAD、組込み用コンパイラ、組込み用実時間並列オペレーティングシステムを開発する必要がある。
- ・並列処理を一般化し実効性能を高めるためには、自動並列化コンパイラを研究開発することが重要である。また、分散共有メモリへの自動分散などの技術やプログラミング支援ツールの果たす役割がさらに重要になってきている。
- ・広域ネットワークを利用したいわゆるメタコンピュータに対応した計算の枠組みやそれらの応用を開拓する必要がある。
- ・MPI インタフェース、データパラレル言語 HPF、OpenMP などの動向の把握およびそれらのコンパイラの開発あるいはアプリケーションの適応などが重要な課題である。

### (3) アプリケーションに関して

- ・高速の計算を必要とする応用分野は大規模な数値計算に基づく数値シミュレーションとビジネスアプリケーションに2分される。大規模な数値計算においてはペタフ

ロップス級の計算機も必要とされているが、全般的な計算需要の割合としては低下していくであろう。一方、後者に関しては、ビジネスデータの多角的な分析業務を効果的にサポートするツールやデータマイニングなど多くの計算量を必要とするアプリケーションがあり、このような分野でのポテンシャル向上が重要である。アプリケーションに関しては、計算そのものだけではなく、モデル化の問題、アルゴリズム、計算結果を正しく認識させる技術も重要である。基本的なモデリングについては、企業では難しいため、大学や政府の研究所での研究が必要である。日本は、このようなモデリングについても、米国、欧州での研究に遅れをとっている。

これらを総合すると我が国は、情報処理産業の各分野で米国に水をあけられているという認識が共通している点が指摘できる。ただし、半導体メモリや液晶といった情報処理機器の部品ではなく（むしろこの分野ではアジア諸国からの追い上げが問題となっている）米国に大きく遅れをとっているものは、プロセッサや基盤ソフトである。アプリケーションはどうであろうか。これはアプリケーションの分野によってまちまちである。アプリケーションといっても、ミドルウェアと呼ばれているような基盤ソフトに近いものは、やはり米国にリードを許している。また、CADなどの一般ユーザを対象としないソフトも同様である。ゲームソフトやワープロソフトなどでは、日本も対等以上に闘っているが、上述のまとめにあるように基本的なモデリングの面では弱い部分が多い。このように考えてくると、日本の技術開発で欠けているものが自ずと明らかになってくる。それは、共通した仕様や共通のインタフェースの抽出とそれをいかに効率よくインプリメントするかといった戦略である。これは、いわば構成力（一般的には構想力と呼ばれるかも知れない）の差であると言い換えられる。したがって、これから我が国における情報処理技術を活性化し、米国に対抗するためには、情報処理技術における構成力を高める必要があるのではないだろうか。21世紀に向かって、このような構成力に基づいた戦略的な技術開発がより重要になると考えられる。

昨年来、我が国のシステムが金融業界を代表として国際的な競争に太刀打ちできなくなっていることが盛んに指摘されている。これに対して特に米国のシリコンバレーを中心とした情報処理産業はベンチャー企業を中心として活況を呈している。米国のこのような企業は専門家集団として活動し、様々なネットワークを通じて情報をすばやく取り入れるとともに、他の企業と提携することによって多様な組み合わせが可能となるという特長を持っている。もうひとつの米国企業の強みは、オープン性を最大限利用してグローバルスタンダードを早めに手に入れてしまうという戦略が効を奏しているという点である。我が国においても、情報処理のハイテク産業のポテンシャルを高めるためには、企業がすばやく行動できるような身軽さや環境が必要であろう。このような点は、我が国と米国とを比較すると、現時点では明らかに大きな差がある

と考えられるが、日米の個々の技術的なポテンシャルはそれほど大きな差があるとは考えられないので、構成力と機動性と多様性を生かす工夫があれば、我が国の情報処理のハイテク分野をより活性化することが可能であろう。

さらに、米国ではシリコンバレー的な企業の活動を補完する政府の役割も戦略的に考慮されており、それに加えて研究予算の配分や公募するプロジェクトなどの査定におけるオープン性も発揮されている。後述の SC97 の報告の中で述べる例を引くと、米国では 1985 年に米国内に 5 つのスーパーコンピュータセンタを設置したが、その時 10 年後に見直すということになっていたそうである。実際に 10 年が経って、ダウンサイジングやネットワークの発展などにより、センタの役割も異なってきているはずであり、米国では、この見直しを数年かけて行った。センタの存在意義は認めたものの、結局 2 つのセンタに集中投資をするという方向を打ち出した。しかも、これらの議論の過程はある程度オープンに行われている。このように、米国では、米国の先端技術分野での優位性を保つために、NSF や DOE などの国家機関が中心となって様々な施策が講じられており、それがオープンに議論されることによってより実のあるものとなっている。

したがって、これからの我が国の施策としては、個々の企業の機動性を高め、競争によってポテンシャルを高めることを可能とする施策を行うとともに、政府としては日本全体の先端技術、特に情報処理技術の競争力を高めるために必要となる施策を講ずるべきではないだろうか。その第 1 は、情報処理研究や情報処理技術の進展に必要なとなるインフラの整備であろう。ただし、この場合に、施設だけを整備するという通り一遍のものではなく、その中身まで考慮した計画ときめ細かなチューニングが不可欠である。例えば、高速のネットワーク網を張り巡らすにしても、どこにどのようにネットワークを張ればそれを有効に活用できるかを策定することも重要であり、また米国の HPCC 計画に見られたようにそのネットワークを用いてソフト、教育、通信など広範囲の応用的な研究が多省庁間で横断的に行われたことは参考にすべきであろう。その第 2 は、21 世紀前半における情報処理技術において、我が国が主導的な役割を果たすために、現時点で解決しなければならない、いくつかの点について戦略的な観点からの技術推進プログラムの設定であろう。この点に関しては、本報告書での各委員の意見が参考になるであろう。例えば、ハードウェア的には、DRAM 混載などを含んだシステムオンチップあるいはオンチップマルチプロセッサあるいは、大規模な FPGA の開発とその応用技術などへの潮流があり、このための技術開発を進める必要がある。私見では、このような技術開発に向けた重要な視点としては、ただ単にハードウェア素子の技術開発よりむしろ、このようなチップを開発するための、CAD システムや IP(Intellectual Property)と呼ばれる、機能ブロックの標準化を図りそのインタフェースを規定することなどがより重要になると思える。ソフトウェア分野においては、

並列処理が一般化してくるという潮流の中で、並列処理コンパイラや最適化技術、あるいはクラスタ計算機上での処理技術などの果たす役割はさらに大きくなっていくため、それに備えた施策が必要である。また、今後、技術開発における構成力や組織力に基づいたデファクトスタンダード化がますます重視されると考えられる。このようなことを米国と対等に行うことのできる人材を育てる必要もあろう。そのためには、構成力のある技術に対する資金的なあるいは制度的な支援を行い得るスキームを確立することや技術的な構成力をつけるための、教育やこれをエンカレッジするための仕組みに援助を行うことも重要な課題であろう。

(山口喜教主査)

## 付属資料1 気象シミュレータ (講演)

東大気候システム研究センター 木本助教授

### 1. 研究目的

人類の生命、財産、経済活動などに非常に大きな影響を与える気候現象の解明は、近年の研究活動によって、より必要不可欠なものとして認識されるようになってきている。具体的な解決すべき課題として、日々の生活行動の指針となる天気予報の精密化、現在および将来の経済活動への警鐘を与える地球温暖化現象の解明、さらに現在、人類を含む動植物系に大きな影響を与えているエルニーニョ現象や異常気象の解析・解明がある。

一方、本物の地球相手には不可能なさまざまな「実験」を実現させてくれる、計算機モデルの確立もまた研究目的となっている。

### 2. 研究対象と物理モデル

地球の表層部である大気と海洋に存在する空気と水の状態、運動を数値計算によって定量的に把握する。図1に地球表層の気候システムで起きる現象を示すが、一言で言えば太陽から受けたエネルギーが大気・海洋の運動によってどのように再分配されてゆくかを表したものである。

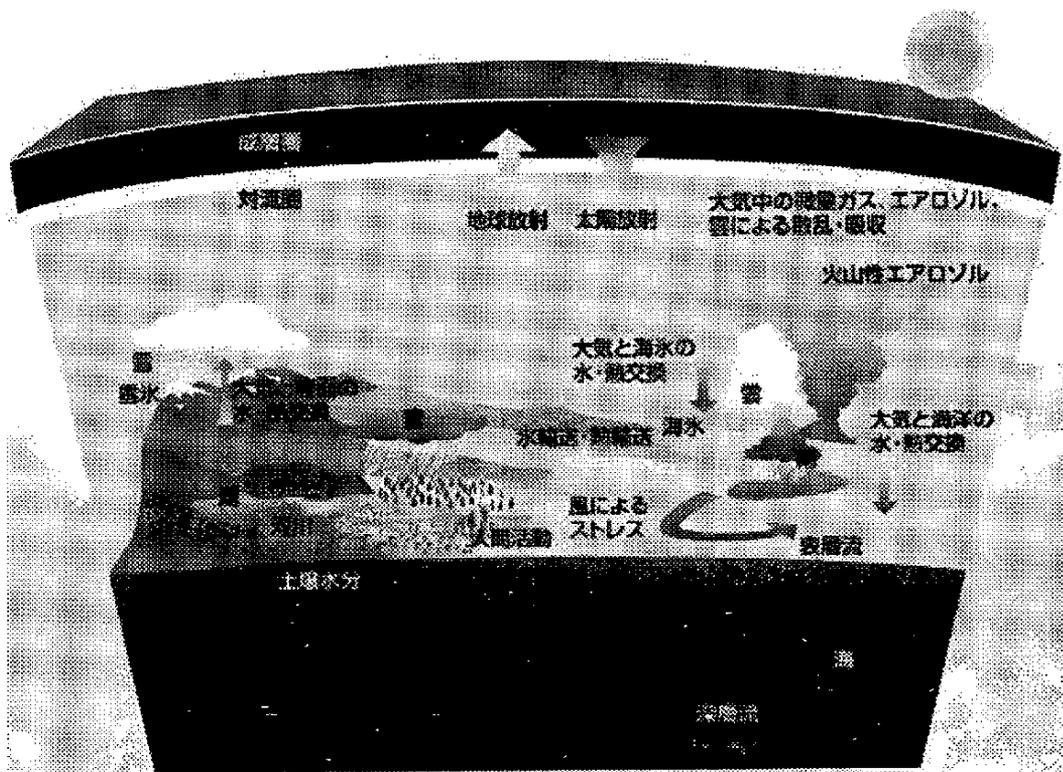


図1

このエネルギーの流れには、水蒸気、雲、雨と形態を変えてシステム内を遍歴する「水」の流れが、凝結熱の解放などを通して、大きな役割を果たしている。大気の運動や海洋の流れは流体力学計算を行い、それに加え、上記の水の変化など気象特有の物理過程計算を行う。シミュレーションにかかる労力は流体力学計算が 40%、物理過程計算が 60%程度である。

図2に、植物も含めた地表面付近での物理過程モデルの一例を示す。

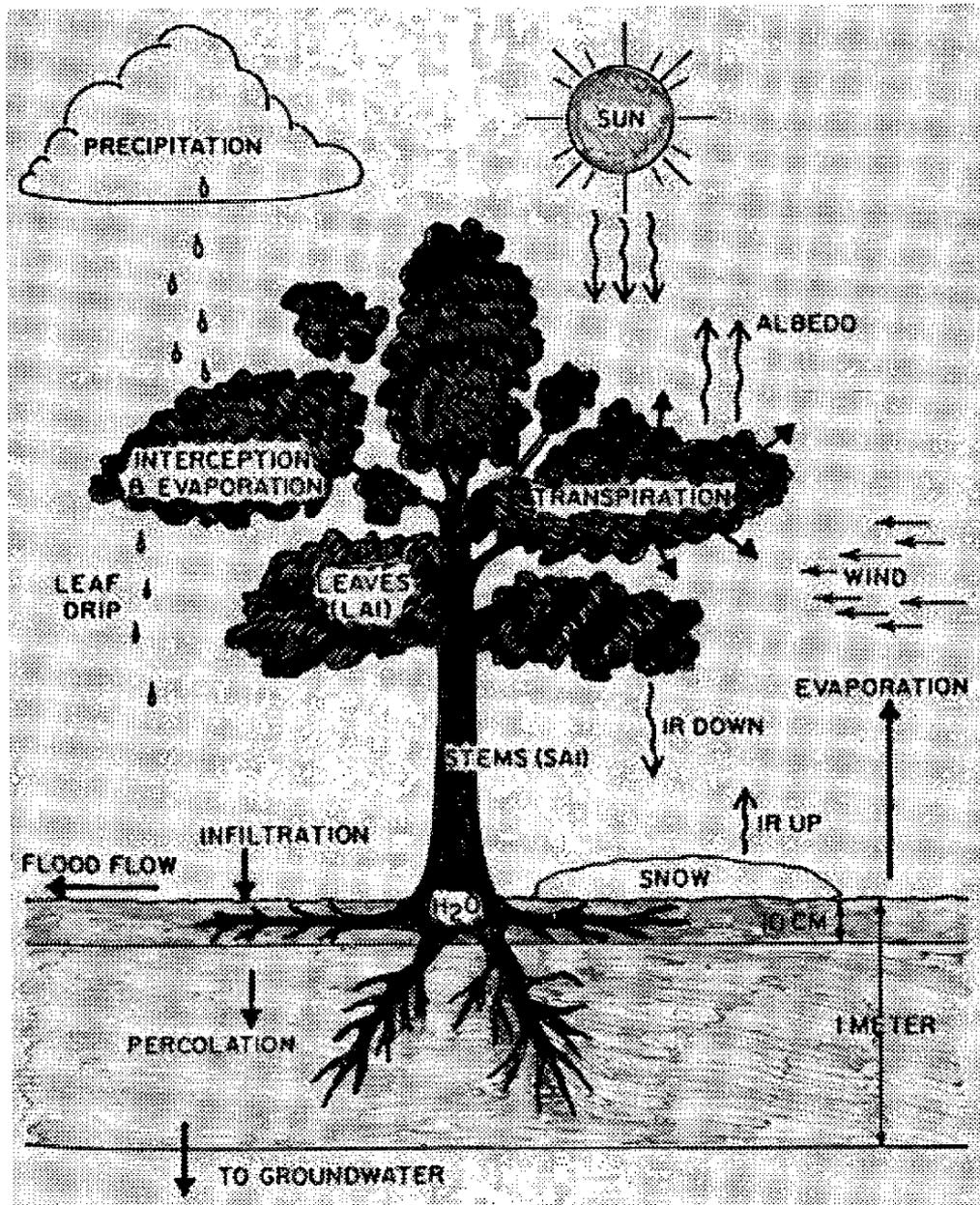


図2

3. 計算方法

流体力学計算は流体の運動量変化を記述するナビエ・ストークス(Navier-Stokes)方程式とエネルギー保存則である熱力学第一法則を連立して解くことによって行われる。物理現象は水の蒸発、凝結等の相変化や太陽エネルギーの吸収・散乱等を定式化する。図3に流体計算と水の相変化などの物理過程計算を含んだ大気循環の計算モデル構造を示す。

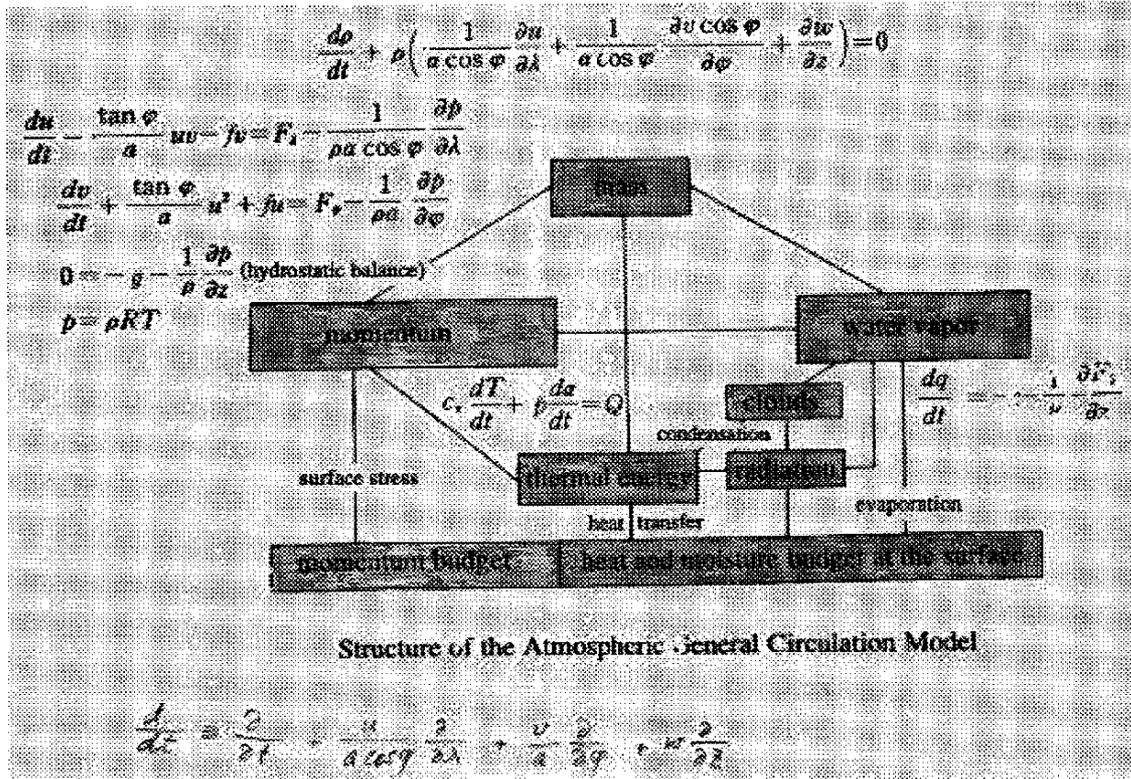


図 3

このようにして出てくる方程式は、常微分方程式、偏微分方程式が混在したものであり、これらを連立して解かなければならない。またナビエ・ストークス方程式など非線型なものも含まれており、解析的に解くのは不可能である。

したがって、解法は数値的なものとなり、対象とする空間領域や時間発展を離散化して取り扱わなければならない。空間の離散化には3次元メッシュが用いられるが、東京の天気予報では地表の水平方向で20km間隔、エルニーニョ現象など地球規模の現象解明には数百km間隔が用いられる。地球全体を離散化した例を図4に示す。

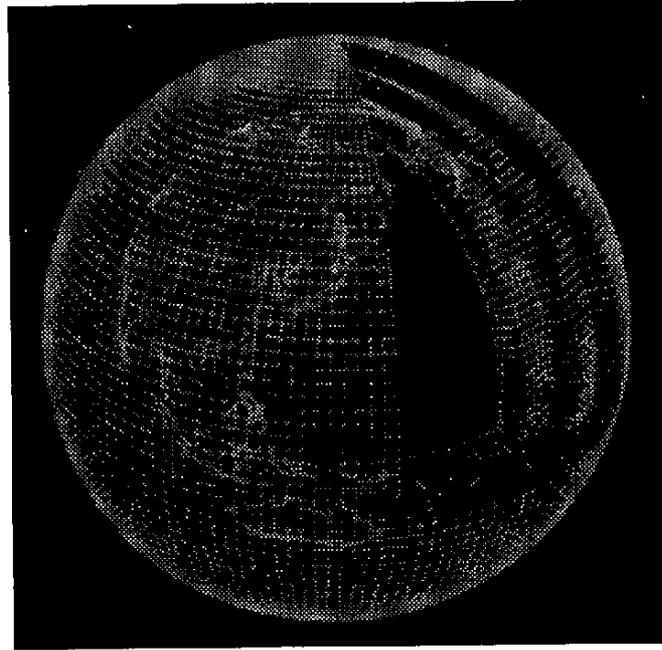


図4

方程式の解法は流体力学計算では関数展開法を用い、物理過程計算では差分法を用いる。関数展開法では、大気の流れが経度方向（赤道と平行）の周期性を利用して球面調和関数展開し(式1)、その係数空間（波数空間）での解を探索する。ナビエ・ストークス方程式は音波からエルニーニョ現象まで、非常に広範囲の波長領域の現象を解として含んでおり、波数空間での取り扱いは、解を限定する上で非常に都合が良い。

波数空間での物理量は、経度方向ではフーリエ変換、緯度方向ではルジャンドル変換によって求められる(式2)。関数展開法には、物理量の保存やエイリアシングエラーがない、また時間積分で陰解法が使いやすいなど、計算上の利点がある。

$$\psi(\lambda, \mu) = \sum_{m=-M}^M \sum_{n=|m|}^{N(m)} \psi_n^m Y_n^m(\lambda, \mu)$$

式1

$$\psi_n^m = \int_{-1}^1 \frac{1}{2\pi} \int_0^{2\pi} \psi(\lambda, \mu) e^{-im\lambda} d\lambda P_n^m(\mu) d\mu$$

式2

実際の計算では図 5 に示すように緯度計算部分は赤道を挟んで対称に分割して解き、経度方向は高速フーリエ変換を用いて波数空間での問題に変換して解く。

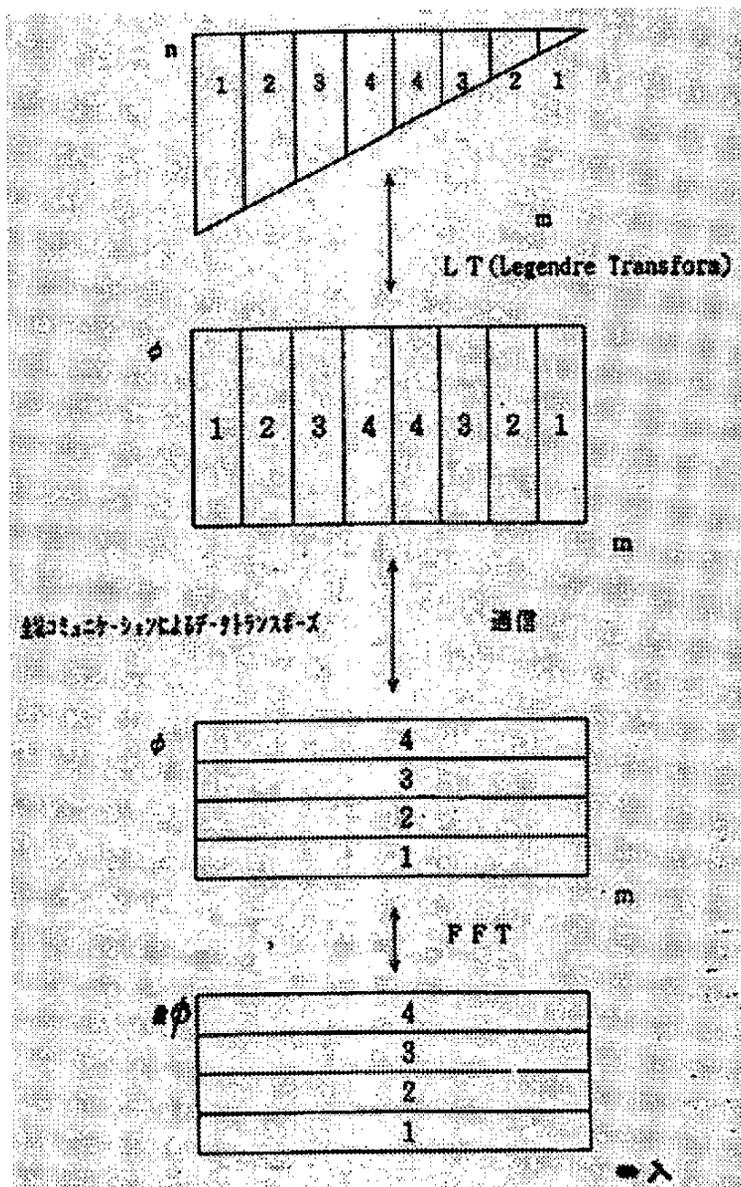


図 5

物理過程を表す方程式は、各メッシュ毎に計算するので複数プロセッサに分割して並列化が容易である。

代表的な計算では、グリッドの大きさ 70km に対し、使用する波の成分の数は (式 1 の  $m, n$  の総数)  $\times$  (高さ成分) で  $213 \times 213 \times 30$  波を用い、5 変数、アップデートする変数を数百万個とる。計算時間は 1 ステップあたり 10 秒程度、24 時間後の結果を出すのに 20~40 分かかる。プログラムは 4 万行程度の規模である。

#### 4. 並列処理

気象関連の計算は、計算量が多く、結果の即時性が求められるため、現存の最速計算機の使用を必要としており、現在は、並列化されたパイプラインベクトルプロセッサによって計算を行っている。

高速化の要は並列化であるが、偏微分方程式を陰解法で解くため、並列化に対応容易とされる格子点法でも全格子データを用いた計算が必要で、関数展開法の並列化も同レベルの労力でコード生成が行えることが判っている。また、ベクトル並列、スカラー並列を問わず、同様に並列化が実現でき、並列化コード生成は比較的容易である事が明確となっている。並列化並列計算機メーカー担当者もしくは専門家なら理論値の 70%程度の効率を持ったコードを生成できる。

並列化コードの研究は、欧米が盛んだが、日本では RIST（高度情報科学技術研究機構）のプロジェクトで研究されている。

#### 5. シミュレーション結果

過去においては、計算過程で経験値を用いた数値解析が行われ、解の真実性、安定性を保っていたが、現在では、純計算を行っても現実とかけ離れない計算結果を得ることができるようになってきた。例えば、以下で詳しく述べるが、エルニーニョの発生時期や規模の予測や、海洋での黒潮の蛇行まで再現できるようになってきている。

##### 5.1 エルニーニョ現象

エルニーニョ現象は短期気候変動のうち最も重要なものの一つであり、数値解析もさかんに行われている。図 6 に赤道東太平洋の海面水温の時間変化を示す。1995 年までは実測値であり、2000 年以降はシミュレーション結果である。ただし、2000 年以降の結果は予測ではないことに注意。エルニーニョに伴う海面水温変動の振幅や周期性がよく再現されている。

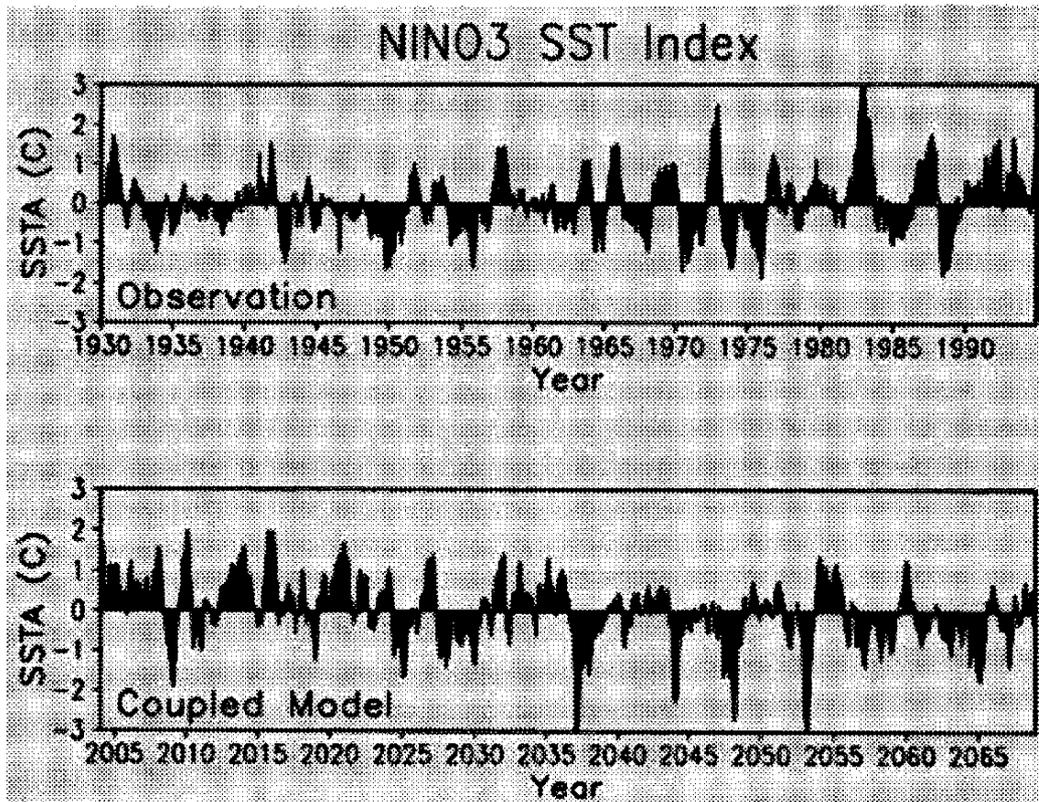


図 6

図 7-1、図 7-2 に、太平洋全体の海面水温偏差の分布の実測値とシミュレーション結果を示す。ニューギニア近辺の水域を除くと水温分布を良く再現していると考えられる。

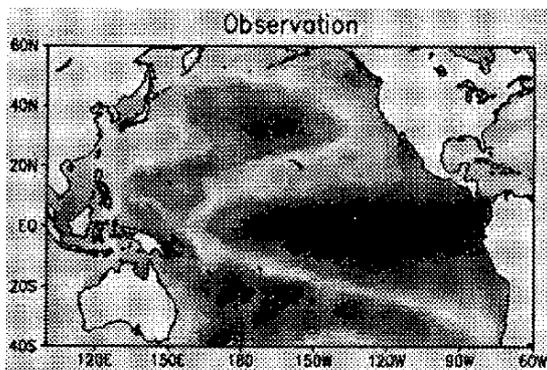


図 7-1

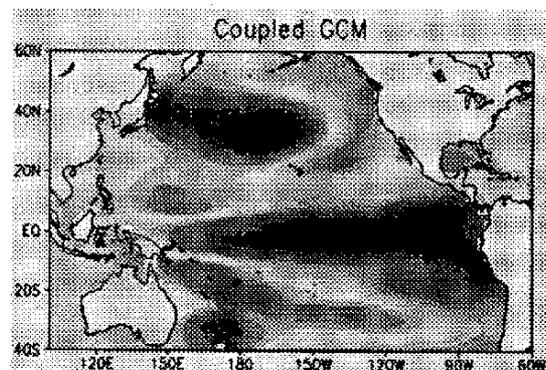


図 7-2

図 8-1 から図 9-2 に、よりグローバルな地球全体の気象シミュレーションの結果を示す。図 8-1 は海面温度の実測値、図 8-2 は降雨の実測値、図 9-1 は海面温度のシミュレーション結果、図 9-2 は降雨のシミュレーション結果である。

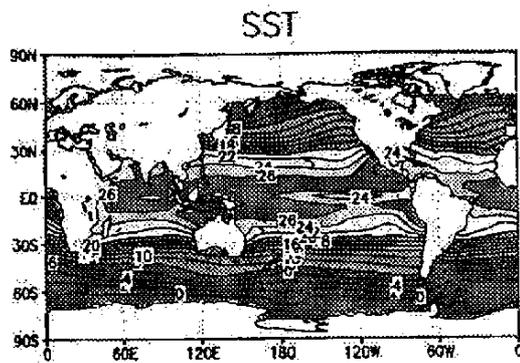


図 8-1

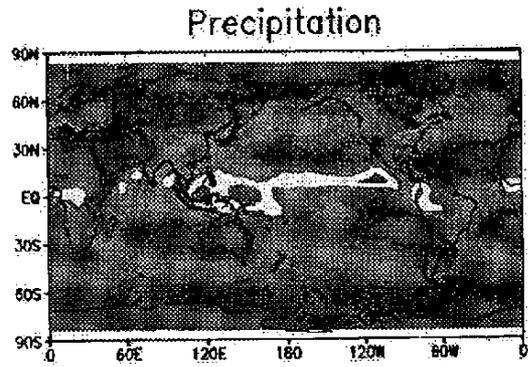


図 8-2

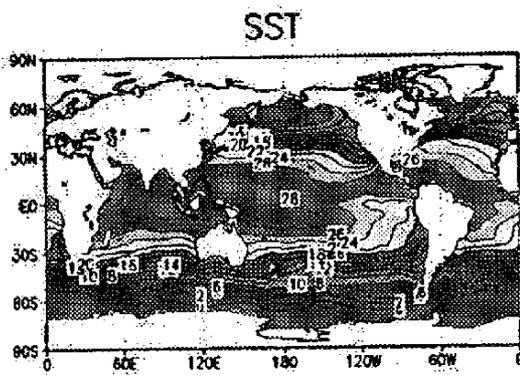


図 9-1

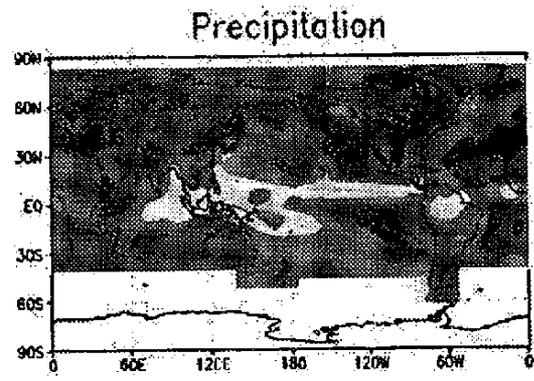


図 9-2

海面温度、降雨ともに、実測をよく再現していると考えられる。

## 5.2 二酸化炭素濃度上昇による地球温暖化

昨年京都での地球温暖化会議の開催でも明らかなように、地球全体の温暖化は現在最も議論されている課題の一つである。特に、化石燃料の使用などによる人為的な二酸化炭素濃度の増加がその原因としてあげられている。そこで、二酸化炭素濃度上昇による気温上昇を見積もることが重要な研究課題となっている。

図 10 に大気中の二酸化炭素濃度を現在の実測値 345ppm に固定した場合(黒丸)と、さらに年率 1%ずつ上昇させた場合(白丸)の地球平均気温の変化を示す。70 年後、その差は 2 度程度となり、海面上昇等、問題が予想される有為な結果となった。

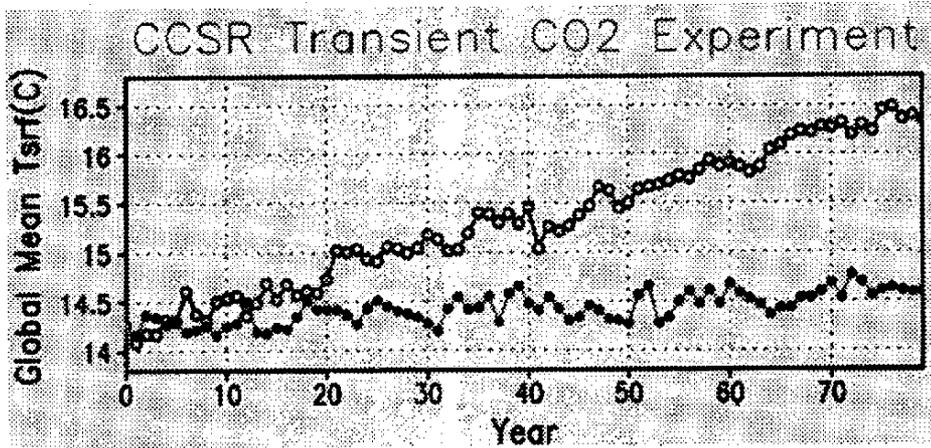


図 10

図 11-1、11-2 には図 10 で濃度上昇が起きる場合の 70 年後の世界を現在との差で示す。  
 図 11-1 には年平均温度分布を、図 11-2 には降雨分布を示す。

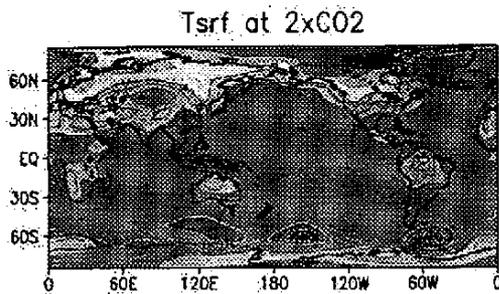


図 11-1

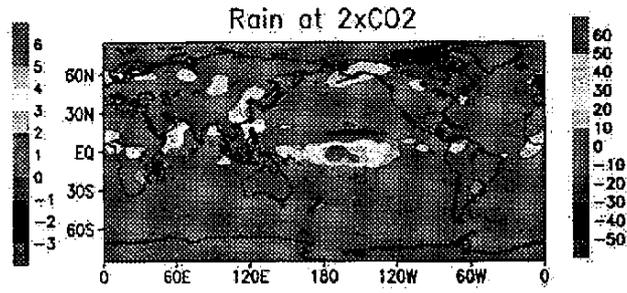


図 11-2

温度上昇、降雨量変化とも一様ではなく、変化の大きい地域と小さい地域が生じる。高緯度地域や大陸での温暖化が進み、低・中緯度で豪雨と干ばつが並立するという状況がみられる。

### 5.3 海洋シミュレーション

海流の流れ、温度分布などは漁業などの産業活動を大きく左右するため、その予想結果の重要性は議論を待たないが、潜水艦関連の軍事情報としても重要な意味を持っている。

我々の結果ではないが、図 12 に米国で行なわれた全球モデルとしては最も高解像度な海水温度のシミュレーション例を示す。ここで注目すべきは、日本近海で温度の高い水域の曲がりくねった様子が見出され、黒潮の蛇行を再現できる事が示された。

## Global ocean model simulation



図 12

### 6. まとめ

気象シミュレーションの場合、細かいメッシュの使用は結果の精度向上に有用である。当面のメッシュ細分化の希望は、大気で 30 から 40km、海洋で 10 から 20km で、この範囲では現在のアルゴリズムを使って力づくで計算できる。これより小さいメッシュでは物理現象をより細かく表現しなければならなくなり、例えば、1 個の雲の形状まで考慮しなくてはならなくなる。また、空間分割を細分化すると時間間隔も小さくしなければならず、膨大な計算量が必要となってくる。

しかしながら、気象特有の物理過程の中味まで踏み込んだ特殊用途の計算機開発はあまり現実的でない。なぜならば、物理過程のモデル化は研究機関、時代によって変化するものであり、単一機能では実用とならないからである。

計算機実験の現状は、24 時間で 10 年を計算している。将来的には、24 時間で 100 年、かつパラメータを 100 種類変えたい。つまり、現在の 1,000 倍の能力が必要である。

計算能力の向上は並列化によって行われるのは仕方がないが、素人にも簡単にプログラ

ム変更ができるような言語環境を整えて欲しい。私自身は、HPF が出てくるまでは専門家にお願いしようと思っている。

ところで、地球温暖化問題では各研究機関での計算結果の比較結果のエラーバーが大きいと指摘されているが、現在解明されていない物理現象もあり、モデルが違うことが原因の一つである。そもそも、計算実験回数の不足という経済的問題もあり、研究者としては不本意である。

## 付属資料2 SC'97(Networking and Supercomputing'97)参加報告

電子技術総合研究所 山口喜教

1997年11月15日から21日までシリコンバレーの中心 San Jose (サンノゼ)市の convention center で開かれた、SC'97 に参加したので、その報告を行う。この会議は、従来は Supercomputing 国際会議という名称で、9年間行われてきた。1997年の第10回目からは、High Performance Networking and Computing と名称を改めて、ネットワークの分野にも重点を置いて開催されたが、略称は従来の会議名を踏襲して SC'97 となっている。この会議のこれまでの経緯を見てみると、米国の東と西で交互に開かれており、その開催地は以下のようになっている。

- 1988年 第1回 Orlando, フロリダ州
- 1989年 第2回 Reno, ネバダ州
- 1990年 第3回 New York, ニューヨーク州
- 1991年 第4回 Albuquerque, ニューメキシコ州
- 1992年 第5回 Minneapolis, ミネソタ州
- 1993年 第6回 Portland, オレゴン州
- 1994年 第7回 Washington, ワシントンDC
- 1995年 第8回 San Diego, カリフォルニア州
- 1996年 第9回 Pittsburgh, ペンシルバニア州
- 1997年 第10回 San Jose, カリフォルニア州

1997年の第10回の参加者は、主催者の発表によると5444名とのことであった。これは、例年並の人数ということであろう。会議の内訳は、基調講演、招待講演などが18件、研究論文の講演発表が61件(内、日本からは3件)、研究論文のポスター発表が69件(内、日本からは2件)、研究展示(Research Exhibits)が48件(内、日本からは4件)、企業展示(Industry Exhibits)が72件(内、日本からは4件)、その他チュートリアルや Education Program、Exhibitor Forum などがあった。会議録は、紙ではなく CD-ROM だけで供給されていた。

### 1. 開会式および基調講演

まず、会場となった San Jose 市の女性市長の挨拶があった。続いて、IEEE Computer Society の会長 Barry W. Johnson 氏が、新たに "Seymour Cray Award" を設けることを発表した。SGI/Cray はこのための20万ドルを拠出したとのことである。次に、クレイリサーチで SGI の副社長である Irene Qualters 氏から、この新たな賞の対象として、「Seymour Cray が示したような創造性のある高性能計算システムに対して、革新的な貢献をした個人に贈られる」というアナウンスがあった。この賞の最初の受賞者は、1998年の会議で発表されるとのことである。

基調講演は、未来学者 Paul Saffo 氏による "Is Digital Dead?" と題するものであり、過去の技術の歴史の傾向を鑑みて、21世紀のデジタル技術の予測を行おうとするものであった。講演者によれば、1980年代は、マイクロプロセッサの時代として特徴づけられ、その主要な機能は processing であった。1990年代は、レーザーなどの光技術の時代であり、その主要な機能は access であった。これは、光通信によってコストが下がり、WWW やニューメディアが可能になったからである。では、2000年代の主要な技術は何であろうか、それは sensor の時代であるというものである。安価で小型のセンサが日常生活に革新をもたらし、その主要な機能は interaction となるというものであった。このような、未来予測の話を、もう少し具体的なセンサ素子の応用例などをあげて述べてはいたが、あまり技術論としてこれからの研究開発の方向性に対して役に立つような話は特に得られなかった。

## 2. 招待講演

招待講演は全部で 17 件あったが、そのうち 2 件は教育に関するものであり、1 件は各種の賞の受賞者による講演であった。招待講演の中で、特に本会議の 2 日目と 3 日目の朝一番のセッションが、State-of-the-field talks と題して基本的な分野のサーベイに当てられた。2 日目は、John L. Hennessy (Stanford) と Ken Kennedy (Rice) による、アーキテクチャとコンパイラの現状に関する講演であり、3 日目は、ペンシルバニア大学の David J. Farber による、通信とネットワーキングに関する講演と、スタンフォード大学の Pat Hanranhan による visualization に関する講演であった。ここでは、これらの講演を中心に紹介する。

### 2.1 Hennessy 教授による講演

スタンフォード大学の Hennessy 教授は MIPS アーキテクチャの提唱者であり、分散共有メモリアーキテクチャの DASH プロジェクトの推進者として有名である。この講演の演題は、"Perspectives on the Architecture of Scalable Multiprocessors: Recent Development and Prospects for the Future" というものであり、過去の歴史を引き合いに出しながら、スケーラブルな並列計算機の将来について述べていたが、結論的には教授の提唱している CC-NUMA (cache coherent nonuniform memory access) のアーキテクチャが有利であるという主張のみであり、あまり新鮮味はなかった。詳細な講演内容は以下のようなものである。

性能価格比の面で、マイクロプロセッサは圧倒的であり、スーパーコンピュータも、この流れに抗することはできず、やはりマイクロプロセッサベースになるであろう。スケーラブルなアーキテクチャへの挑戦は、スケーラブルなメモリバンド幅を提供することによってなされるが、それには 2 つのアプローチがある。1 つは、分散した局所メモリをプロセッサに持たせて、プロセッサ間の通信を行うものと、物理的に分散したメモリを 1 つのアドレス空間とするものである。どちらのアプローチも、アドレス空間に対する局所性とス

ケーラブルな相互結合網が必要であるという点で共通している。様々な観点から見て、結局スケーラブルな共有メモリが重要であり、その論点を整理するとコヒーレントキャッシュに基づく分散共有メモリが重要であるという結論になる。キャッシュコヒーレントのアーキテクチャはいくつかあるが、スケーラブルな特性を持たせるためには、バスのような放送的な手段では駄目である。directory を用いれば、これをポイント間通信で置き換えられる。結局、コヒーレントキャッシュのための directory とメモリが共に分散しているアーキテクチャが有効であり、これは DASH のプロトタイプによって 1991 年に実証された。

#### [マルチプロセッサの収れん]

共有メモリ型計算機も、メッセージパッシング型計算機も、クラスター型計算機もネットワークのインタフェースとコントローラを除くと、違いがなくなってきた。ただし、ネットワークに関しては、プロトコルの機能の違いや遅延やバンド幅が異なるという点がキーポイントである。バンド幅とパフォーマンスとのトレードオフ、および遅延のトレードオフからキャッシュコヒーレントによる分散共有メモリタイプのアーキテクチャは、高いバンド幅と低い遅延を持ち、コストは高いが、スケーラビリティに優れている。

#### [将来の方向性]

- (1) プログラミング可能性 (容易性) は依然として、キーポイントである。小規模の並列計算機の普及によって、並列ソフトウェアが増加すれば、この点は多少は改善される。
- (2) アーキテクチャ的な方向性は収れんしてくる。分散共有メモリとクラスター計算機の融合した超並列計算機を設計できるかがポイントであろう。
- (3) 相互結合網のバンド幅と遅延の改善が進む。
- (4) クラスター化によって、より大規模化の方向にいくであろう。

そこで、このような方向性を仮定して、マルチプロセッサにおけるプログラミングの問題点を考えてみると次のようにまとめられる。分散共有メモリ (DSM) におけるリモートメモリアクセス時間は、時代とともにクロックサイクル数でみるとむしろ増加している (たとえば、1960 年代のバス結合のマシンでは 50 サイクル程度であったものが、1996 年の商用の DSM では 180 サイクルになっている)。したがって、プログラミングモデルとしては、DSM については、いかにして本質的な局所性を引き出すかがキーであり、またクラスタ計算機については、いかにして効率的なプログラムにするかといった点が重要である。

結論としては、スケーラブルコンピューティングのモデルとして、DSM とクラスタコンピュータがあるが、将来はクラスター化された DSM に向かうと考えられる。その場合に、パフォーマンスを向上させるためのカスタム化と、コスト削減のために商用部品のバランスをいかに保つかがポイントとなる。また、どのようなアーキテクチャが必要になるかについて、応用が鍵となる役割を果たす。

個々の技術的な動向に関しては、興味深い点もあったが、全般的にはあまり目新しい話ではなく、DSM にこだわりすぎているような印象を持った。これからの、半導体技術の進展や、それによるメモリ混載アーキテクチャやオンチップマルチプロセッサの将来への技術展望などを聞かせて欲しかった気がする。

## 2.2 Ken Kennedy (Rice University)教授による講演

演題は、"Programming Support Software for High Performance Computers" というものであった。

まず、ソフトウェアの成功の鍵として①メモリ階層の管理、②並列性の抽出、③サポート技術 の3つをあげた。さて、実際に成功したかどうかという評価であるが、スケーラブルな並列計算のためのコンパイラやツールの技術は多少の成功はあったものの貧弱である。成功例としては、MPI や PVM などの message passing ライブラリや性能評価や最適化ツールなどがあげられよう。そのように限定された成功の理由としては、次の2つがある。

- (1)アーキテクチャの変化のペースが速くて、汎用的な戦略を立てられない
- (2)HPCC による投資が、あまり生産的ではなかった

その理由としては、ソフトウェア技術への直接的な投資は不適切であるという点と grand challenge の応用が技術的な発展を犠牲にして性能面だけを重視したためである。また、産業への技術移転に対する時間差 (HPF コンパイラは今頃になってやっと洗練されてきた)、あるいはマシンに対する独立性が薄いことや決まりきっていない問題に対するソフトウェアのサポートが弱い点 (たとえば、定型的でない問題はランタイムサポートを必要とする) などもあげられよう。

つぎに、将来のアーキテクチャはどうなるかという点であるが、1つはペタフロップスアーキテクチャで1万から10万のプロセッサを持ち、10レベル以内のメモリ階層から構成され、この場合には高いレベルの並列処理が必要とされる。もう1つは、異機種分散システムで、これには、高性能システムが地理的に分散したものやネットワークで結合されたPCなどがある。

このような予想のもとに、なすべきことを列挙すると以下ようになる。

- (1)性能上のボトルネックに挑戦すべきである。具体的には、メモリ階層問題を改善したり、より並列性を抽出することへの挑戦が必要である。
- (2)ポータブルな HPC(High Performance Computing)をサポートする必要がある。
- (3)プログラミングの抽象度を高める必要がある。
- (4)コンパイラの設計を再考する必要がある。具体的には、最適化をロードやランタイム時まで延期したり、データやランタイム情報をコンパイラに統合したり、ツールのサポートを統合することなどである。

では、将来のプログラミングはどのようなであろうか？ 仮定として、プログラムは既存のコンポーネントから構成され、しかも高性能が要求されるとする。結論としては、将来のプログラミングはスペシャリストとエンドユーザーという2つのクラスに2分されるであろう。エンドユーザーに対する未来のプログラミング言語は、コンポーネントライブラリへのインタフェース的なものや図的なものとなるであろう。

ここで、Ken Kennedy は script-based programming system とされるものを提示した。このシステムは、全システムを、まず中間言語(Intermediate Language)に落として最適化し、それから個々の違ったマシンに落とす。中間言語は、並列性とメモリ階層をサポートしなければならない。高性能な並列プログラムへの変換は、標準的な中間コード+通信という形態になる。また、このような状況における新しいコンパイラのアーキテクチャとしては、計算というものをより柔軟に定義し、コンパイラは部分計算と位置づける。そして、いろいろなレベルでのコンパイラのステップがあると考え、信頼性やセキュリティを高める必要がある。具体的に、様々な会社からのコードを利用するために、trusted compiler という方式を提案した。これは、様々なソースコードを公開暗号キーで暗号化して、コンパイラに送り、その計算機上でコンパイルできる形にしてコンパイルし、ターゲット計算機上で実行するというものである。

Ken Kennedy によって提案された、このようなコンパイラの全体システムの様子を図1に示す。これは、従来から知られているコンパイラというよりは、コンパイラを含んだ実行システムというようなものではないだろうか。

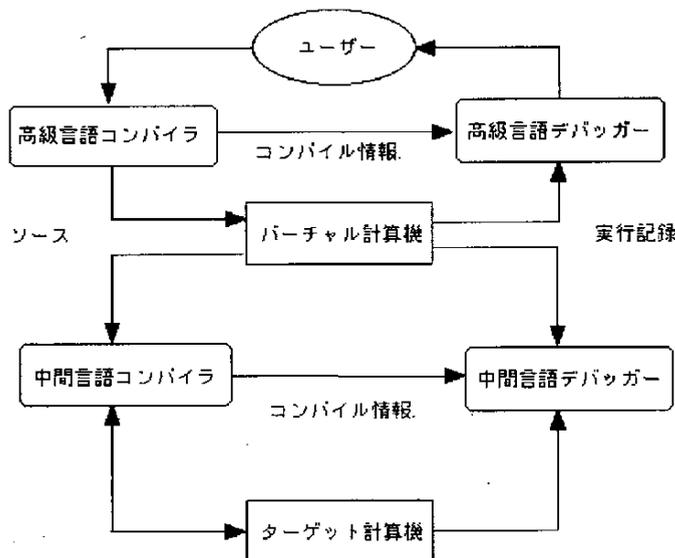


図1 Ken Kennedy 教授が提案したコンパイラシステム

### 2.3 David J. Farber (ペンシルバニア大学) 教授による講演

演題は"The Future of Telecommunication and Networking --- an Attempt to Predict the Unpredictable"というものであり、20年前は、computers with network であったが、その後 computing and communication の融合があり、将来は network with computers となると指摘。ネットワーク関係の技術の急激な進歩（ケーブル、モデム、無線、光ファイバ）にもかかわらず、現状の状況はジョウゴのようなものになっている。すなわち、バックボーンは回線容量が太くても末端はストロー並の太さ（容量）になってしまっていると指摘。将来は、人工現実感やビデオ配信などのために大きなバンド幅が必要であり、そのボトルネックを除くために、通信プロトコルを再設計する必要があると主張した。また、今後は、地理的に離れた場所にあるソフトウェアを有効に働かせるメタコンピュータが重要であると述べた。

### 2.4 Pat Hanrahan (スタンフォード大学) 教授による講演

演題は"The Visual Computer"というものであり、まず、人間の歴史から見て、物事のビジュアル化が非常に有効であったと指摘し、今後人間と計算機のインタラクションをパワーアップするためにはますます、ビジュアライゼーションが重要になってくるという主張を展開した。そして、Ray tracing などのビジュアル化手法ばかりでなく、イメージに基づいた表現などをサポートするために、超並列のグラフィクスが必要になってくるという主張が展開された。

## 3. パネルについて

この会議では、例年多くのパネルが行われている。この年も、全部で12個のパネルがあった。そのほかに、教育に関するパネルが5つもあり、懐の深い会議であると感じさせた。ここではそのうち3つだけを紹介する。

### 3.1 アーキテクチャに関するパネル

これは、Breakthrough and Challenges ahead in Computer Architecture と称するパネルで、モデレータはUCBのDavid Cullerであった。このパネルでの論点は、アーキテクチャにおける、進歩の可能性はどこまであり、ブレークスルーは何であるか、それに対してアーキテクトはどう挑戦し、またプログラミング上の挑戦はどうであるかというものである。パネリストは以下の4人であった。パネリストの意見をまとめると次のようになる。

#### (1) J. H. Gray (Microsoft)

これから少なくとも25年間は、1年にバンド幅が3倍になる。これを活用するための計算機が必要になる。予想としては、テラバイトの通信インタフェースを持つシステムオンチップが重要である。

## (2) Greg Papadopoulos (サンマイクロシステムズ)

彼の主張によれば、並列処理の難しさは、並列性の抽出とその動的な割り当てのような制御に関わる部分と、データの分散の2つの問題に集約される。データ分散については本質的に SMP (シンメトリカル・マルチプロセッサ) ができればあまり問題にはならない。とは言っても、アドレスのバンド幅、データのバンド幅、それにメモリに対する遅延によって限界はある。DSM は共有メモリに比べると良い解ではない。なぜならば、ローカルなメモリとリモートメモリとのアクセスの差が3倍以内でないといけないからである。

## (3) Burton Smith (Tera Computer)

彼は、マルチスレッドアーキテクチャに基づく Tera コンピュータの開発者であるが、今後のアーキテクチャとして、共有メモリは Uniform か Non-Uniform かに関わらず必要であり、同期は高価でありベクタパイプラインはまだ死なないなどと述べた。そして、Latency に関わる問題点として、ローカルでない大域的な通信、アムダールの法則、それにベクトル長が十分長くないことによるベクトル化の不完全性などを指摘した。それを克服するための、ブレイクスルーとして、彼が設計した Tera コンピュータのような、細粒度のマルチスレッドと PIM (Processor in Memory) をあげた。ただし、後者はローカルな部分でしか速くならないと指摘した。したがって、これからのアーキテクチャは、マルチスレッドを前提にして、キャッシュの問題や、分岐予測や、投機的実行や同期の問題を解決するアーキテクチャを研究すれば良いとの指摘をした。

## (4) J. Hennessy

彼の指摘は、予想される進歩として、マイクロプロセッサ技術の進展により 5-10 年で single chip multiprocessor ができるというものである。こうなれば、プロセッサ間の相互結合網はチップ内でなされるために、有利になるが、しかるにメモリについては SDRAM や Rambus などで、バンド幅は増大するが latency は大きくなる。そこで、その解決手段として processor in memory が登場する。メモリモデルについては、Papadopoulos とは逆に、SMP は全般的に遅いが、DSM はリモートは遅いがローカルは速いと DSM を擁護した。Programming challenge については、プロセッサ数の増大に伴って、効率化のために必要なことがどうしても増大するが、その割合をできるだけ小さくすることが必要という抽象的な指摘に終わった。

このあとの会場からの質問や議論では、smart card の将来や将来の並列のアプリケーションにはどのようなものが想定されるかなどが話題となった。アプリケーションに関しては、データベースや薬の開発 (B. Smith)、テレコミュニケーション (G. Papadopoulos)、コンピュータ同士の通信 (J. Gray) それに各種の認識 (J. Hennessy) に対して並列処理が重要であるなどといろいろな意見があった。また、共有メモリや分散共有メモリなどの違いに関する議論として、J. Hennessy と G. Papadopoulos で意見が分かれた uniform アーキテクチャと non-uniform なアーキテクチャの得失についてもホットな議論が展開された。

### 3.2 PACI に関するパネル

PACI (Partnership for Advanced Computational Infrastructure) とは米国の NFS (National Science Foundation) のスーパーコンピュータセンタが中心となって展開する大学連合のプロジェクトである。NSF では、現存する 4 つのスーパーコンピュータセンタ (Cornell, Pittsburgh, Illinois, San Diego) をリストラし、Illinois (NCSA) と San Diego (NPACI) の 2 つに絞った。この 2 センターを中心として大学連合を形成し、センターとの間を高速のネットワークで接続し、計算資源を共用しようとするものである。年間予算は全体で、65M ドルとのことである。このパネルは、その関係者が集まったものである。

モデレータは、NFS の Bob Borchers という人で、パネリストとして上記の 2 つのセンターとその他にカリフォルニア大学のサンフンシスコ校とサンタバーバラ校からそれぞれ 1 人ずつ参加していた。パネルでは、今まで孤島にみんなバラバラにいたけれども、このプログラムによって、みんなが共同できるようになるとか、最新のデータや最新のアルゴリズムを組みあわせることができ、これらの実行結果をビジュアライズすることもできるようになるなど、このプログラムの良さが強調されていたが、固定の予算ではだめだという指摘もあった。

討論では、デスクトップの進歩によって HPC マシンはいらなくなるのでは、というような質問があったが、速い計算を必要とする問題はいくらでもあると反論していた。また、アルゴリズムについて特に molecular dynamics についてうまく行くのかという質問があったが、5 年後には、いろいろなレベルでうまく結びついているはずだと答えていた。

### 3.3 ASCI Strategic Alliances のパネル

ASCI (Accelerated Strategic Computing Initiative) プログラムは、基本的には核兵器の仮想実験などを目的として、米国の 3 国立研究所の共同研究としてスタートした。この中で、大学に研究を委託するプログラムがあり、これは ASAP (Academic Strategic Alliances Program) と称されている。本パネルは、このプログラムに関するものであり、DoE (米国エネルギー省) の Alex Largelere がモデレータをつとめた。まず、このプログラムでは核兵器の研究をしているわけではなく、難しいシミュレーションを行っているという説明があった。さらに、ASAP プログラムで実際に書かれたプログラムを 5 人のパネリストが紹介した。最後に、このプログラムのレベル I は終了し、新たにレベル II の募集をしているという紹介があった。これは、3 年間で、1 年のトータルな予算が 500 万から 700 万ドルというものである。研究内容については、(1) コンピュータサイエンスと数値計算、(2) 数値科学の応用という 2 つの分野で募集し、レベル I よりも間口を広げるとのアナウンスがあった。募集要項では、アルゴリズムはもちろんデータ操作やビジュアル化、ソフトウェアツールなどがあり、後者では、エネルギー関係だけではなく物性や計算科学や計算機械なども含んでいる。これに関する情報は、<http://www.llnl.gov/asci/> にある。

#### 4. 展示

この国際会議の大きな特徴として、展示を大規模に扱っていることがあげられる。展示は、企業展示と研究展示に分けられる。企業展示は、もともとは最新のスーパーコンピュータの発表と売り込みの場であったが、スーパーコンピュータ需要の伸び悩みにより、全般的には縮小傾向にある。これに対して、研究展示の方は、高性能計算のアプリケーションを中心にむしろ増大傾向にある

##### 4.1 企業展示

企業展示は全部で、71 件あった。それを分類すると、スーパーコンピュータ関連のベンダが 11、大容量記憶装置が 11、ネットワーク関連が 12、ソフトウェアやツールなどが 17、出版が 6、その他学会、国際会議、地域、システムインテグレーションなどが 14 であった。全般的に、記憶装置、ネットワーク、ソフトウェアなどの展示が増えている。

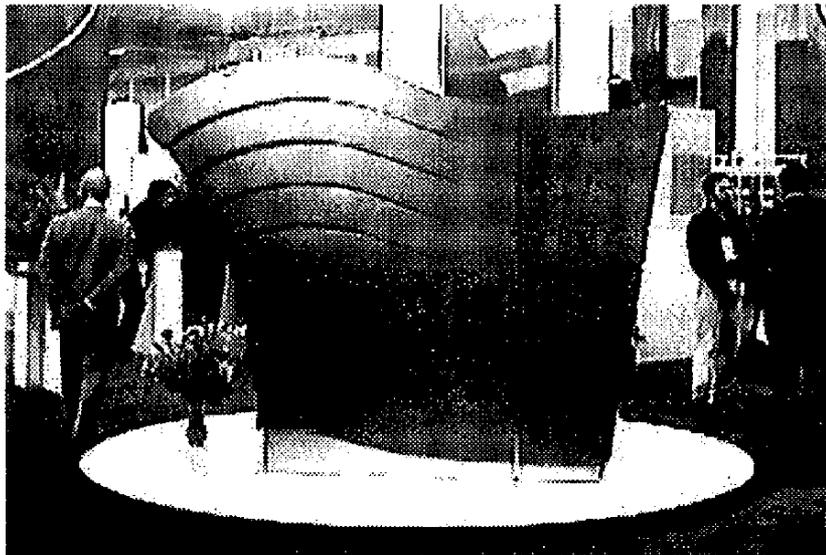


図2 Tera コンピュータの MTA システムの筐体

スーパーコンピュータ関連のベンダでの注目は、Burton Smith の Tera Computer Company が、マルチスレッドアーキテクチャ(MTA)に基づく、Tera の 1 号機をやっと出荷するという事で、その筐体が展示されていた (図2 参照)。ガリウムヒ素を使ったプロセッサを用い、1 プロセッサあたり 1Gflops で、最大 256 プロセッサ (ただし 1 筐体には最大 32) まで増設可能であるとのことである。Tera の特長は、そのスケーラビリティの良さにあると強調していた。1 号機は、カリフォルニア大学サンディエゴ校(UCSD)に納入されている。98 年中に、16 プロセッサまで拡張される予定だそうである。なお、CMOS 版のプロセッサも設計中であるとのことである。テラコンピュータの情報は <http://www.tera.com/> に、また UCSD での Tera の評価については、<http://www.ucsd.edu/> にある。

スーパーコンピュータの老舗である SGI/Cray は、大きなブースで展示と説明会を行っていた。SGI/Cray は、従来からの Cray 計算機の延長上にある 600MHz の Alpha を使った T3E-1200 の他に、CC-NUMA に基づく並列計算機の Origin 2000 の 64 台版を会場に持ち込んで宣伝をしていた。IBM は、8 ノードの Deep Blue でチェスの対戦を行って見せていた。また、今後の Power PC (ASCII Blue Pacific (LLNL)を含めて) の中心技術の一つである銅配線の実物なども展示されていた。

日本のスーパーコンピュータのベンダである、富士通、NEC、日立はそれなりのブースを構えて参加していたが、最近の米国におけるスーパーコンピュータの入札問題などもありあまり意気があがっている感じではなかった。日本からは、他にソニーが大容量ディスクのブースを構えていた。

SUN マイクロシステムズでは、いわゆる汎用サーバマシンである Enterprise に加えて、HPC 10000 (通称 Starfire と呼ばれている) という高性能なハイエンドマシンを売り込んでいた。この計算機は、共有バスに基づくいわゆる SMP (シンメトリカル・マルチプロセッサ) のアーキテクチャで、1つのユニットには 250MHz の UltraSPARKCII を最大 64 プロセッサまで搭載可能で、最大 35Gflops、ディスクも 20TB まで搭載可能とのことである。ターゲットとしては、ネットワークの高性能サーバとして売り込んでいるようであったが、複数のユニットをクラスタとして結合して、最大 16 ユニットで 1024 プロセッサのスーパーコンピュータを構成する計画もあるとのことである。サーバマシンとしては、ほかに HP の PA-RISC を用いた Exemplar や DEC Alpha プロセッサの 614MHz を用いたサーバの展示などがあった。DEC のブースでは、人の体や目や唇の動きをセンサやカメラで読みとって、そのデータをもとにアニメーションをリアルタイムで動かすというデモを行って、人目を引いていた (図 3)。



図3 SUN マイクロシステムズでのデモの様子

ソフトウェアやツールの関係では、OpenMP のブースが注目に値する。OpenMP は David Kuck が設立した Kuck & Associates 社と Cray Research 社が中心になって進めている共有メモリアイプの並列計算機上でのプログラムをポータブルにするための取り決めで、基本的に directive に基づいて並列化を記述するモデルである。ブースでは、OpenMP のコンセプトの紹介や実例によるデモンストレーションを行うと共に、参加企業や参加者の勧誘が行われていた。OpenMP の Web サイトは、<http://www.openmp.org/>にある。

その他、目新しいところでは、世界最大のパソコンシェアを持つ Compaq がパソコンのクラスタを出展していたことがあげられる。スーパーコンピュータのダウンサイジングが本格化する前触れということかも知れない。

#### 4.2 Research Exhibits

Research exhibits は、昨年は 44 件であったが、今年は 47 件と増加しており、またブースの広さや取り組み方をみても、昨年よりは充実していた。特に、米国の国立研究所や NASA の研究所はそれぞれに広いブースを構えて成果を宣伝するとともに、ASCI プロジェクトなどでは共同のブースを構えてプロジェクトについて広報活動を行っていた印象があった。具体的にその内訳をみると、米国からの出展が 39 と圧倒的で、その他の国からは日本が 4 件、ヨーロッパが 3 件、それにブラジルから 1 件となっていた。米国の内訳をみると、大学単独の出展は 13 件で、また国立研究所やスーパーコンピュータセンタ単独の出展は 13 件であったが、その他に ASCI プロジェクトなどのように米国の国立研究所や大学のプロジェクト単位の展示が 13 件もあった。米国の研究所は、このような場をプロジェクトの成果の公開や宣伝に利用しているということが理解できた。特に、米国におけるスーパーコンピュータ・アプリケーション分野における取り組みの豊かさが実感できた。日本からは、電総研(EM-X と Ninf)、RWCP(PC クラスタ)それと航技研と埼玉大学の 4 件が出展していた。昨年の 6 件に比べると件数は減っているが、昨年が続いてブースを構えた電総研、RWCP、航技研などは、ブースの大きさや展示の内容などがより充実していた印象がある。来年以降、日本のスーパーコンピュータ関係のプロジェクト特にアプリケーションに関係する組織やプロジェクトの参加を希望したい。

#### 5. Technical Paper

Technical paper のセッションにほとんど参加しなかったのが、具体的な報告はできないが、今年は、投稿論文が 334 編もあり、例年になく technical paper session が盛り上がったとのことである。採択された論文は 57 編(採択率 17%)と通常の国際会議よりもかなり厳しいものとなっていた。その中で、日本からは、お茶の水大、電総研などの Ninf 関係の論文、東大の喜連川研のデータマイニングに関する論文、それに東大の米沢研の共有メモリアイプ上での GC に関する論文の 3 編が採択された(投稿は 17 件)。採択された論文の全体的な分野の傾向としては、アプリケーション関係が 1/4 程度を占め、あとはア

ルゴリズム、最適化、ビジュアライゼーションなどが続き、アーキテクチャやシステムそれに言語とコンパイラなどは全体の1割程度という感じであった。

## 6. Gordon Bell Prize と Sidney Fernback Award

実際的に使用されている大規模なプログラムで最高の実行速度を出したチームに与えられる Gordon Bell Award は、通常は、Performance 部門と Price/Performance 部門の2つのチームに与えられるが、今年を受賞した1つのチームが両方の部門にエントリーしており、Performance 部門を受賞するとともに、Price/Performance 部門でも良い結果を出していたので、こちらは両方を兼ねて受賞するという、変則的な受賞となったとのことである。受賞チームは以下の2チームであった。

### (1) Price/Performance 部門 シドニー大学とロスアラモス研究所のチーム

28 個の DEC Alpha ステーションを用いて、液体中の粒子の混濁した状態について解析するプログラムを PVM で記述したもので、10.8 Gflops/s/\$M という値を得ている。

### (2) 両部門

ロスアラモス研究所と CALTECH 他のチーム

Performance 部門は、ASCI Red の 3400 ノード(6800 プロセッサ)を用いて重力場の多体問題を解いたもので、解き方の違いにより 430 Gflops および 636 Gflops を達成。また、Price/Performance 部門では、16 台の Pentium Pro のクラスタで同様の問題を解いたものである。

また、大規模な計算問題に関してパイオニア的な仕事をした人に与えられる、Sidney Fernback Award は、コロラド大学の Charbel Farhat が受賞した。

## 7. 全般的な感想

SC と略称されるこの会議は、以前はスーパーコンピュータのためのものであったが、近年のダウンサイジング化の波などを反映して、ハイパフォーマンスコンピューティングとネットワークの会議となった。テクニカルセッションもあり、招待講演やパネルもあり、展示もあり、種々雑多であるとの印象も受けるが、これが、逆に米国の底力を感じさせるものでもある。種々の側面から、研究や開発したマシンやプログラムの優位性を広く宣伝し、また使ってもらおうという姿勢が今日の、情報分野におけるオープン化の中での戦略として根付いているとの印象を受けた。この会議は、海外にも門戸を開いてはいるが、本質的には米国のためのものであり、米国におけるハイパフォーマンス計算の分野での動向を知る上で、今後も大いに参考になると考えられる。

## 謝辞

本稿をまとめるにあたって、東京大学の小柳義夫教授からいろいろと御教示していただいた。ここに感謝する次第である。

## 付属資料 3 HECC 研究プロジェクト概要

ここでは、1998年度版 Blue Book にトピックスとして掲載されている研究プロジェクト・テーマを、1997年度版 Implementation Plan を参照し、各省庁毎、予算項目毎に分類した結果を紹介する。

### 1. NSF (全米科学財団)

#### (1) Supercomputer Centers

この予算項目に入るのは、PACI (Partnerships for Advanced Computational Infrastructure) と呼ばれるプログラムで、NSF のスーパーコンピュータセンタを中心に研究が行われている。米国の学術領域では、2~3 年以内に使用可能となるシステム性能がテラフロップスに達するため、大部分の大学が持つ計算能力の 1 桁~2 桁上のハイエンドシステムを有する 2 ヶ所のサイトとの共同研究が行われている。

ハイエンドな研究開発に直接関係しない研究者は、教育、訓練等に貢献し、インフラ全体を強化、助長するソフトウェアの開発を行う。

競争により次の 2 サイトが選出された。

① NPACI (National Partnership for Advanced Computational Infrastructure)  
California 大学 San Diego 校 (UCSD) を中心に、California 大学 Berkeley 校、Caltech、Michigan 大学、Texas 大学の共同研究サイトに高性能サーバが置かれており、リモートセンシングデータ、数値シミュレーション出力、専門的データベースなどの大規模データ解析を可能とする計算科学・工学に見受けられるデータ主導のコンピューティングに焦点を当てている。

② NCSA (National Computational Science Alliance)  
Illinois 大学 Urbana-Champaign (UIUC) の他に Boston 大学と DOE の Argonne 国立研究所が計算資源を提供している。共同研究では、可視化用スーパーコンピューティング技術を Silicon Graphics, Inc. と進めている。

#### (2) Computing systems

この予算項目に該当する Blue Book のトピックスは、見当たらなかったが、ここでは将来を見通したコンピューティングシステムのハードウェアについての研究がテーマとなっており、ペタフロップス関連の研究テーマが、入っている可能性が高いと考えられる。

### (3) Applications

この予算項目は、高度コンピューティングの必要性をアピールする重要な役割を担っていると考えられ、次の二つのテーマが掲載されている。

#### (a) 体積レンダリング技術

体積レンダリング技術は、3次元スカラー場を可視化するいくつかの技術を含んでいる。例としては、3次元スカラー場から定数面を抜き出す isocontouring がある。また、前進もしくは後進投影を用いた直接的3次元データ表現がある。前進投影法は、一般的に用いられてきたグラフィック技術であるが、3次元場のサンプルをスクリーンに投影する。後進投影法は、通常、体積レイキャスティング法として参照されるが、ピクセルの色とその混成したものを反映した3次元場の部分集合を見つけることによって、各ピクセルの色を決定する。NSF が支援している体積レンダリング研究は、Purdue 大学が先導している。

#### (b) 非線形電子輸送研究

NSF が支援する研究者は、ナノデバイス中での平衡からかけ離れた電子の動力学特性に現れる強い非線形性を研究する新しい量子動力学シミュレーション方法を開発した。この方法は、平均場近似における電子-フォノン相互作用とランジュバン方程式を通じた散逸を統合している。この方法は、アモルファスシリコンにおける電子移動度を含んだ多くの分野での非線形電子輸送の研究に用いられている。

ダブル量子ドット電子波動関数のシミュレーションでは、電子-フォノン相互作用により、確率密度の局所化とドット間の振動を表現することができた。

#### (c) Macromolecular Structure

UCSD Computational Center の Macromolecular Structure (CCMS) の研究者は、分子表面でのシミュレーションの結果を出すために AVS ソフトのフィルタとモジュールを開発した。CCMS は、ドッキングや静電性や親水性のような薬品設計のキーの役割を果たす分子の構造的特徴を解析するソフトウェアを開発している。CCMS は、NSF によって支援された UCSD、Scripps Research Institute、および San Diego スーパーコンピュータセンタの共同プロジェクトである。

## 2. DOE (エネルギー省)

### (1) ACTS (Advanced Computational Testing and Simulation)

DOE のバックボーンとなる研究プログラム DOE2000 の主要な部分は、グランドチャレンジと ASCI (Accelerated Strategic Computing Initiative) であるが、その中で CIC がらみの主要なものは ACTS Toolkit である。これは、DOE の責務に関連したプログラムであり、先進的なコンピューティング技術使用を加速するソフトウェアツール、アルゴリズム、環境の統合システムを整備しようとしている。

例えば、Scientific Template Library プロジェクトでは、多数の DOE 研究所と大学の開発者が参加しており、簡単に再利用でき、柔軟性のある data-structure-neutral manner で書かれたソフトウェアツールキットを作成している。DOE として、国立エネルギー研究計算科学センターと Argonne、Lawrence Berkeley、Los Alamos、Oak Ridge の 4 国立研究所で新しい高度コンピューティングアーキテクチャを使用できるように支援している。

一方、米国の核弾頭の安全性、信頼性、性能を維持するための最先端のモデリングとシミュレーション能力を確立するための ASCI プログラムが存在し、その関連の ASCI Academic Strategic Alliance Centers は、Science Based Stockpile Stewardship (SBSS) の対象とする計算科学的、および計算機科学的領域の研究を促進する高等研究センターとされている。その目標とするところは、大規模な、モデリング、シミュレーション、計算の科学的方法論を確立することである。このセンターは ASCI 研究所と連携し、ASCI 研究者との共同研究・開発を促進する。

### (2) National Energy Research Scientific Computing Center

ここでは、関連が深いと思われる Grand Challenge に含まれるトピックスも紹介する。

#### (a) 軽原子核研究

軽原子核 (40 個以下の中性子と陽子) の性質は、実際の 2 つか 3 つの原子核相互作用によって計算される。核子のスピンと荷電状態 (isospin) に強く依存した複雑な力のため、核子の性質を計算するのに多体法が用いられる。原子もしくは物性計算に用いられるクーロン力とは異なり、この力を定義する有効な基礎理論は存在しない。核子-核子散乱データをフィッティングすることにより、部分的に二体力で計算できるが、多体計算は三体の相互作用力を持つような他の性質をテストする必要がある。DOE が支援する研究者は、この力に関する知識と核子の振る舞いを予言する知識を高めている。

## (b) 数値トカマク乱流プロジェクト (NTPP)

DOE が支援する NTPP の主要なテーマは、プリンストンのプラズマ物理研究所トカマク融合試験炉、ジョージアの DIII-D、日本の JT-60 といったトカマクで最近発見されたプラズマ乱流抑制法研究で出てきた gyrofluid の研究である。これらの方法をスケールアップすれば、より魅力的なかつ経済的な核融合炉設計が出来るようになる。いくつかの NTPP 研究は乱れた渦を伸ばしたりちぎったりする後背流におけるずれの重要性に焦点を当てており、乱流の抑制、付随した輸送の削減につなげる。速度のずれの安定効果は多くのトカマク実験で得られている。トカマク実験の磁場形成はシミュレーションで見られるずれ流れの安定性に関係し、最近閉じ込め実験を改善した物理に導く。この学際的研究のゴールは、核融合炉の性能を最適化するのに必要なトカマクプラズマ乱流の実際的なシミュレーションを行うことである。

## (c) 最適設計技術

感度解析は、いくつかの異なる問題を含んだ航空機の翼設計のような最適設計で用いられる。最適設計は、設計パラメータに関するプログラムコードの感度を計算し、勾配ベースの最適化技法を用いることにより、自動的に出力される。ANSI-C (ADIC) のための自動化微分ツールは、Argonne 国立研究所の数学およびコンピュータ科学部の研究者によって開発され、南バージニアの NASA Langley 研究センターのいろいろな専門的設計最適化 (CSCMDO) コードのための設計・感度解析計算機に適用された。CSCMDO は計算流体力学で用いられる構造体積格子の自動的改変のような設計最適化環境に適合している。ADIC が改善した CSCMDO は、要求された体積格子感度を自動的に作成する。CSCMDO は 高速かつ高度に自動化された 3 次元体積格子作製能力を有しており、基本システムの摂動を反映した表面および体積での格子作製ができる。代数的技術は設計最適化から生じる幾何学的変化を反映させたブロックの面と体積の格子を作成、改変するのに用いられる。CSCMDO の自動微分版から得られた数値は異なる設計パラメータに対する体積格子感度導関数である。

## (d) MDH の表面反応性

DOE が支援する研究者は、りんご酸脱水素酵素 (enzyme malate dehydrogenase (MDH)) の計算機シミュレーションと反応機構解析を行っている。良好な初期的な結果により、研究者は、最小エネルギー面と MDH により、りんご酸とオキサロ酢酸とで相互変換する反応パスを計算した。エネルギープロファイルの解析から蛋白質マトリクスによる溶媒効果は劇的に MDH 反応に含まれる機能グループの固有反応を変化させる。この酵素は効果的に気相の発熱反応を MDH の蛋白質溶媒環境における等熱反応に変化させる。エネルギー分解解析は基

板近傍の特定の MDH 残留物が著しく陽子輸送の安定と水素化物輸送の非安定化へのエネルギー分に寄与するこのデータはアミノ酸が MDH の表面反応性に重要な役割を果たしており、部位特異的突然変異誘発実験と無矛盾であることを示唆している。

#### (e) 分子動力学

全体最適化(GO)技術はマクロ分子モデリングとシミュレーション問題解法の中心である。というのは、これらの分野の多くの基礎的な問題は、GO 問題として定式化されるからである。この研究開発の狙いのひとつは、巨大分子系におけるモデリングとシミュレーションで現れる GO 問題を解く大規模 GO アルゴリズムとソフトウェアを支援している Argonne 国立研究所の IBM SP 上の高性能環境を開発することである。研究者は、イオン系の安定配置をみつける最適化法を用いている。100 イオン数より小さいシステムの安定配置は、IBM SP 上の全体連続アルゴリズムを用いて得られる。この研究のゴールは、非常に大きなシステム、いかなれば 20 万イオンシステムの安定配置を見つけることである。そこからイオン系の相転移が観測される。研究者は、全体連続アルゴリズムをレナード-ジョーンズ-ポテンシャルを基としたマイクロクラスタ形成問題へ応用し、75 原子の小クラスタの最適構造を得た。この研究は、半経験的ポテンシャルでの金属クラスタのシミュレーションのような、分子クラスタシミュレーションの一般的領域へ拡張されている。

### (3) HPC Resource Providers

この項目に分類されるテーマは、Blue Bookには掲載されていないが、Grand Challenge をテーマとする研究者に、Intel Paragon、IBM SP2、Cray T3Eなどの最先端計算機リソースや、異機種からなるコンピューティング環境を提供する役割を担っており、そのための研究、整備がなされている。

## 3. NASA (航空宇宙局)

### (1) Grand Challenge Support

#### (a) 計算航空科学プロジェクト(CAS)

NASA の計算航空科学プロジェクトは、航空宇宙領域で必要とされる価値、柔軟性、短サイクル時間を反映したコストで航空宇宙工学問題を解ける高度コンピューティング環境を用いることに注力している。航空宇宙設計問題には、高速度問題、先端亜音速技術、回転翼航空機がある。研究成果は、できるだけ早くシステムベンダに使えるようにする。CAS の研究者は、燃料や設計時間を減らすために航空機や省エネルギージェットエンジン

の計算流体力学シミュレーションにかかわっている。NASA は、ターボファンエンジンの高圧コンプレッサの設計・開発時間の全体を、並列計算シミュレーションによって 18 ヶ月から 14 ヶ月に削減したことを示した。これによって、数百万ドルの節約が出来た。NASA のガスタービン低圧サブシステム(LPS)プロジェクトの流れ解析の狙いは、航空力学の流れをガスタービンエンジン内の完全な低圧サブシステム内でシミュレートすることである。LPS の詳細モデルは数値推進システムシミュレータ(NPSS)に統合され、数値試験セルとして使われるであろう。LPS・NPSS プロジェクトのゴールは、先進的なガスタービンエンジンの設計に関連したリスク、不確実性、コストを削減するツールを提供することにある。

#### (b) 地球と宇宙の科学(ESS)プロジェクト

地球と宇宙の科学研究は、2つの領域をカバーしている。

第一には、先進的な専門分野モデルと全体の変化をよりよく理解するためのスケーラブルな全体シミュレーションの関連付けを行う。

第二には、統合モデルと科学的ミッションから予想される巨大データの処理・解析・理解を行う解析アルゴリズムを開発する。

ESS では、地球の天候のシミュレーションを行っている。また、太陽系の動力学、太陽風、宇宙の生成などのモデル形成をする大型並列計算機を使っている。

#### (2) Testbeds、System Software

NASA の HECC 研究開発センタでは、並列システムソフトウェアツールの開発と、本務としてのアプリケーション開発を行っている。NASA の高度コンピューティング研究施設は高度コンピューティング技術の新世代への脱皮を加速するために設立された。ここでは、NASA 研究所と教育ネットワーク(NREN)や、ストレージシステムの初期システムまたはプロトタイプ及び最先端可視化アプリケーションの使用ができるようになっている。NASA は 1998 会計年度に 100-250 Gflops の性能を持つ、スケーラブルな Teraflops テストベッドを導入する。

#### 4. DARPA (国防総省高等研究計画局)

##### (1) System Environments

High Performance C++

DARPA(Defense Advanced Research Projects Agency) が支援している High

Performance C++ project は、C++ プログラミング言語を用い、並列プログラミングの標準モデルを定義しようとしている。このモデルは、データ並列と分散オブジェクトタスク並列の両方をサポートし、ポータブルな並列アプリケーションの作成を可能とし、コンパイラの最適化をサポートする企業のターゲットを提供する。この最終製品は複雑なメモリ階層を有する超並列システムばかりでなく対称・共有メモリマルチプロセッサのような複数のプラットフォームで使用されると考えられている。DARPA のねらいは、科学的、工学的、商業的アプリケーションのシームレスな統合であり、そしてそれは、従来のハードウェアプラットフォームで走るソフトウェアを含んだ新世代のスーパーコンピュータの資源を必要としている。

## (2) Scalable Systems and Software

### (a) スケーラブルシステム及びソフトウェア

DARPA のスケーラブルシステム及びソフトウェアプログラムは、安全、スケーラブルで分散したコンピュータの技術を開発する。プログラムは、アーキテクチャとソフトウェアコンポーネントを創造することに焦点を当てる。それらは、DOD が必要としている新世代のスケーラブルな高度コンピューティングシステムの商用開発のリスク低減と開発加速を必要としている。ゴールは、ハードウェア、OS、アプリケーションなどの絶え間ない再開発を要求されない優れたアプリケーションが走る高性能システムの調達を可能とすることである。テラフロップクラスシステムのための先進的技術は、この領域では、DARPA が投資する開発から生まれる。

### (b) M-Machine

M-Machine は、細粒度並行動作コンピュータであり、DARPA の "Mechanisms for Teraflops" プロジェクトのもとで、MIT(Massachusetts Institute of Technology)の Concurrent Very Large Scale Integration (VLSI) Architecture グループが設計・作製した。このプロジェクトのねらいは、multi-ALU (Arithmetic Logic Unit) processing nodes の構成により、より効率的に回路密度を増加することである。演算器、レジスタファイル、メモリを設計し、一つの集積回路に落としていくのは容易だが、1 チップ上の多くの演算器を組織化し、制御するには新しい技術が必要である。M-Machine は、H (horizontal)-Threads と呼ばれる並列命令の組によって全ての並列性のグラニュラリティで効率的にプログラムを実行するように設計されている。同様に、V (vertical)-threads を通して、スレッドレベル並列性を利用し、可変パイプライン、メモリ、通信の遅れを隠蔽する。M-Machine は、ノードあたり 1,200 MIPS もしくは 800 Mflops を実現するシリ

コン製造技術を使用する。

(c) National Compiler Infrastructure project

このプロジェクトのめざすところは、コンパイラ分野での開かれた学術的共同研究を支援し、産業への技術移転を促進して、コンパイラ開発のための共通プラットフォームを開発することである。このプロジェクトには、Stanford 大学の Intermediate Format (SUIF) プロジェクトと Virginia 大学の Zephyr プロジェクトが含まれる。SUIF プロジェクトでは、DARPA と NSF が共同設立し、Harvard 大学、Rice 大学、California 大学 Santa Barbara 校の研究者と共同研究が行われている。

ここでは、はじめから複数の産業からのパートナーの参加を予定しており、その目的は、モデュラで、拡張・維持・支援容易であり、現在の研究を支援し、ソフトウェアの再利用が支援できるシステムの開発である。

Zephyr プロジェクトは、DARPA と NSF により支援された Princeton 大学と Virginia 大学の共同研究である。Zephyr プロジェクトの組織構造は、内容により中間言語の形を分類している。Zephyr 中間表現は抽象的な構文設計により記述するツリー様の構造をしている。Zephyr 中間表現の言語的もしくは機械特有な内容は他のコンパクトな設計により記述される。Zephyr インフラを使用していれば、形式が内容から分離されることによって、必要な中間言語の選出や、現存の中間言語の拡張、新しい高レベルな中間言語の開発ができる。

(d) 商用の工学ソフトウェアの並列化

DARPA が支援する商用の工学ソフトウェアの並列化プロジェクトで、連邦政府と産業の共同プログラムである。その目的は、小規模な並列開発プラットフォームしか持たない先端的独立ソフトウェアベンダ (ISV) を直接指名してベンダの投資リスクを最小とし、米国産業市場へのスケーラブル並列ソフトウェア導入に拍車をかけることである。

このプログラムの直接的ゴールは、産業標準となっているシステムソフトウェアとイネイブラ (プログラマが効率的にアプリケーションを実現するのに用いるソフトウェアツール) を用いて、現存の商用工学ソフトウェアの第一世代の並列化を行うことである。

(e) 並列システムソフトウェアツール

このプログラムに参加する各 ISV は、スケーラブル並列コンピューティングハードウェアや最先端のシステムソフトウェアとイネイブラ、さらに標準インターフェースである MPI との互換技術にアクセスできる。各アプリケーション開発プロジェクトに特有な資産

は保護されるが、できあがった並列アプリケーションがポータブルで産業標準と合致するという条件をみたすことで、IBMの共通プラットフォームのサポートが自然に保証される。

### (3) Embeddable Systems

この項目は、1998年度から新設されたものであり、トピックスの分類が正確ではないが、前年度までとほぼ同額の予算を獲得し、1998年度に消滅した Defence Technology Integration and Infrastructure 中の項目名 Middle Ware に含まれると考えられることからこの項目に分類した。

Quorum は情報科学プロジェクトの一部であるが、グローバルな異種分散コンピューティング環境で DOD が絶対に必要とする末端でのサービスの質をエンドユーザが予想でき、制御できることを可能とすることに焦点を当てている。これらの技術とアーキテクチャは、エンドユーザにリソース源が単一プラットフォームのように見えるように、グローバル OS に統合される。それらは、船上でのコンピューティングや指揮制御のような防衛問題のキーとして具現化される。そこでは、人とコンピュータの関わりと設計法、および、人とコンピュータ間の対話をより自然にする技術に焦点が当てられている。

## 5. NSA (国家安全保障局)

### Supercomputing Researchプログラム

NSA の スーパーコンピューティング研究プログラムは、知的処理の計算機支援を一桁改善する、研究とアプリケーションに焦点を当てている。NSA の研究活動は、新しいコンセプトの創造とプロトタイピングを始めとし、並列化に焦点を当てた先端商用プロジェクトを用いた改善にまで及んでいる。現行のプロジェクトは次の3つのプロジェクトを支援している。

- ・空軍機の飛行テストを行う MARQUISE (埋め込み型高性能コンピュータ)
- ・DARPA との協力で、ナノ秒システムクロックの3次元コンピュータアーキテクチャを披露できる試験車を構築する3次元ダイヤモンドMCM立方計算機プロジェクト
- ・量子コンピューティングの可能性を探索するいくつかの基礎的研究への投資

## 6. NIH (国立衛生研究所)

### (1) バイオメディカル研究

National Center for Research Resources (NCRR)のミッションは、バイオメディカル研究発展を可能とするリソースを開発することによって、支援している NIH のために発見の触媒として働くことである。

NCRR は、科学的専門分野間の協力を推進し、新しく出現する研究ニーズにすばやく、柔軟に対応する。その最大の挑戦は、基礎、臨床の療法の研究のトレンドのすばやい展開を広範囲に維持することである。

### (2) Biomolecular computing

分子生物学領域での高度コンピューティング研究は、蛋白質構造、および、他の分子または薬品との相互作用によるその構造的、機能的変化を予測することである。NSF との協力のもとに NIH は、構造ベースの薬品設計のための蛋白質受容体の合成モデルのようなバイオ技術産業に役立つ新しい蛋白質の機能の有益性を証明した。

### (3) 分子生物学計算と AChE シミュレータ

高度コンピューティングを含んだ分子生物学計算は広範囲でかつ、しばしば複雑な計算となる。この研究は、アセチルコリンエステラーゼ(AchE)という人間から昆虫までの種に存在する神経伝達物質、アセチルコリンを抑制する酵素のシミュレーションである。自然界に遍在し、生物学のシステムの鍵となっていることから、AchE は、多くの一般的な医薬品や毒素のターゲットとなっている。NIH と NSF が支援する臨床研究は、タクリン(tetrahydro-9-aminoacridine)のようなアセチルコリンエステラーゼ抑制物質がアルツハイマー病患者の記憶力強化に有効であることを示唆した。

## 7. EPA (環境保護庁)

### 科学における可視化

EPA 科学可視化センタと Wisconsin 宇宙科学工学センタ(SSEC)の協力により EPA の科学者にデスクトップワークステーションを用いて 3 次元可視化能力が与えられた。この能力は、SSEC の環境静止軌道衛星(GOES)からの混成画像を含んだ大気と海洋の出力を可視化する Vis5D システムと一体となる。この画像は、国立大気研究センタ(NCAR)メソスケール気象モデルによって予報された 3 次元雲データの底の白雲として見える GOES から

遠隔で探索した雲のデータと比較される。雲中水は青く見え、降雨水は白く見える。雲の温度は、冷たい青から、暖かい赤まで表現される。Vis5D は、ワークステーションメモリに合うようにデータを圧縮することにより対話型の探索を可能とする。

## 付属資料4 SIA ロードマップ

SIA(Semiconductor Industry Association)ロードマップは、米国の半導体工業会(SIA)が行っている、CMOSを中心としたデバイス技術の将来予想であるが、これまで、1992年版、1994年版、1997年版が公表されている。このロードマップは、単に米国の目標を表すだけでなく、グローバルな技術開発指標になっており、最近では、その世界版を作成しようという動きもある。本資料では、これまで公表された3つのSIAロードマップについて、主な諸元を図表の形で併記し、1997年ロードマップと従来ロードマップの比較を行う。

### 1. DRAM

#### 1.1 DRAM 最小寸法

チッププロセスを表現する最小寸法は、97年版からDRAM HALF-PITCHとMPU GATEに分離されてマッピングされるようになった。これまでの、ロードマップの最小寸法とDRAM HALF-PITCHを重ねてプロットすると図1のようになり、ほぼ一致する。97年版ロードマップでは、1年程度の前倒しとなっているが、現実には、1999年初め、三星が0.18 $\mu\text{m}$ プロセスによる256Mbit DRAMの出荷を予定している。

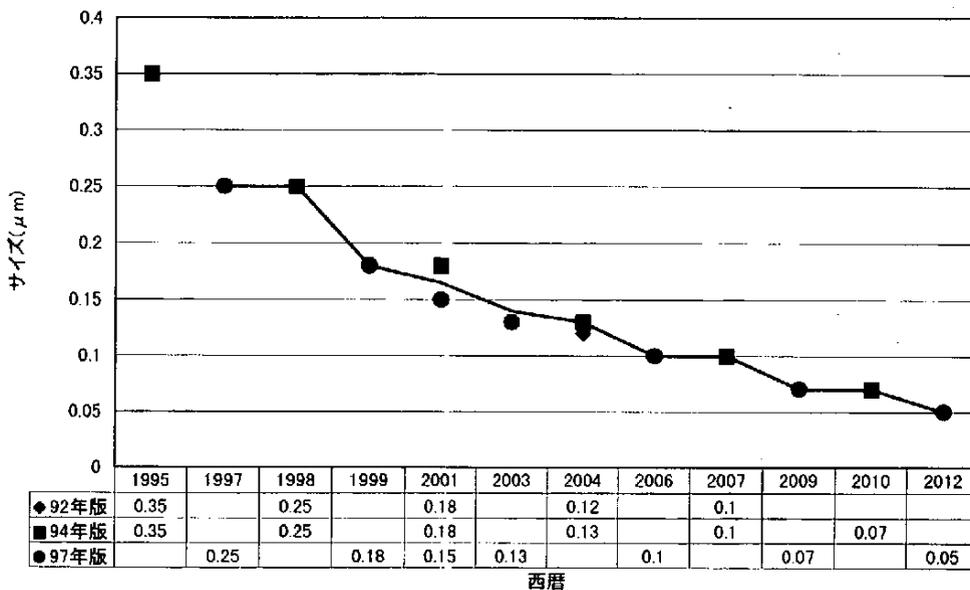


図1 DRAM 最小寸法 ( $\mu\text{m}$ )

### 1.2 DRAM 集積度

図 2 にプロットした結果を示すが、DRAM 集積度も 1 年の前倒しとなっている。実際の 256Mbit DRAM の出荷は、1999 年初めの三星が最初となる可能性が大きい。

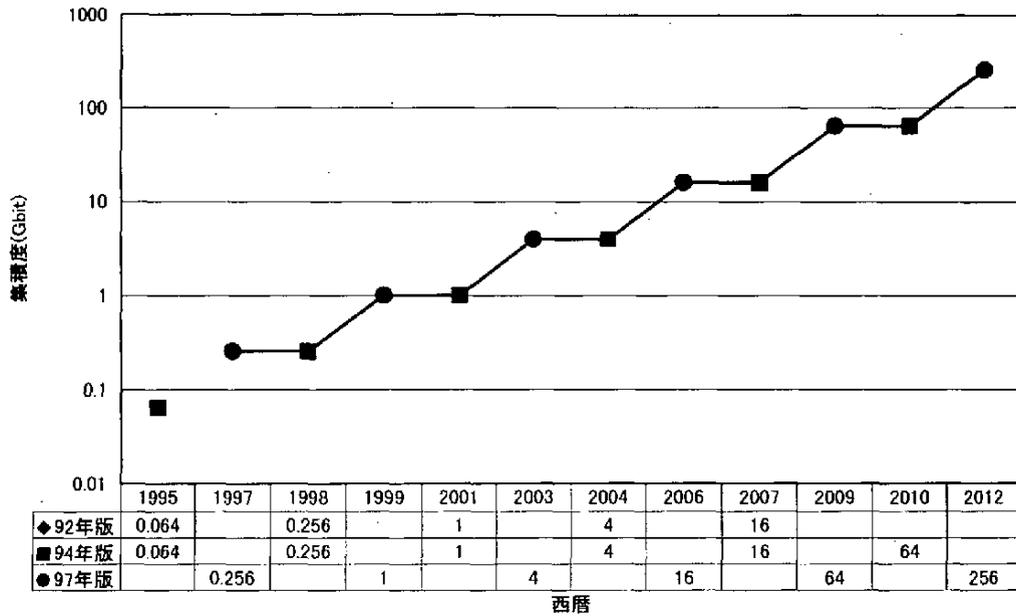


図 2 DRAM 集積度

### 1.3 DRAM チップサイズ

図 3 に DRAM チップサイズのプロットを示すが、近い将来は、これまでの予想よりやや大きめで、将来は従来の予想の大きさになるとしている。

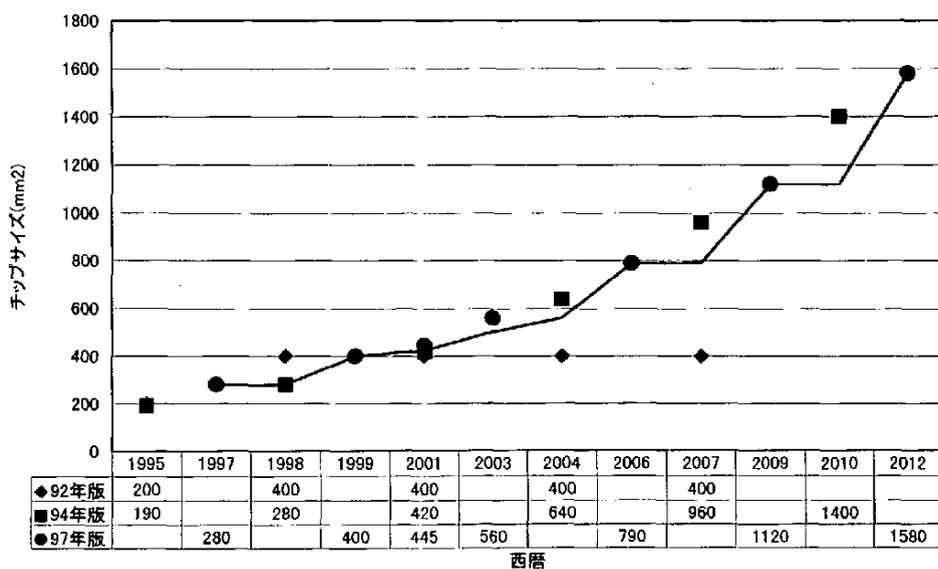


図 3 DRAM チップサイズ

## 2. MPU

### 2.1 MPU 最小寸法 (ゲート長)

これは、1997年版のみ収録されているのでその値を表1に示す。

表1 MPU ゲートサイズ (nm)

暦年	1997	1999	2001	2003	2006	2009	2012
サイズ	200	140	120	100	70	50	35

これらの値を Intel 社の予定しているロードマップと比較する。Intel ロードマップを図4に示す。

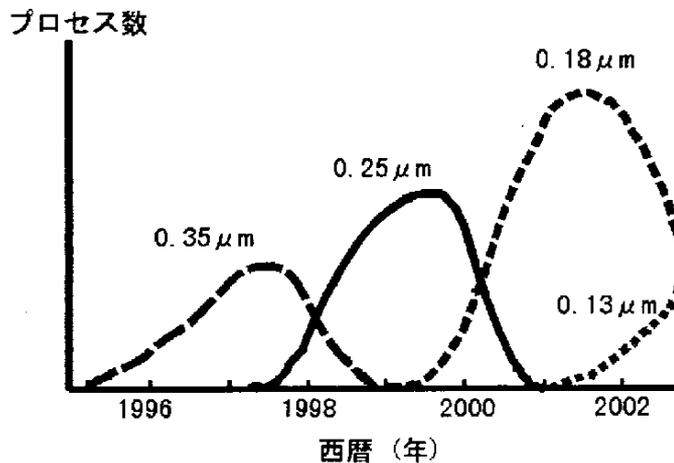


図4 Intel MPU ロードマップ

130nm プロセス投入は、2001年第一四半期といわれており、Intelの微細化の加速のアナウンスはあるが、このままでは2001年120nmの達成は微妙である。

しかしながら、SPIE(The International for Optical Engineering)の第23回国際シンポジウムでTexas大とDupontが発表した光リソグラフィの技術が本物であれば画期的に世代が進むので、表1のロードマップも夢ではない。発表の内容を以下に示す。

- ① 0.08 μmの線幅を波長193nm (ArF)の紫外光を用いて切った。
- ② Texas大学Austin校のgraduate student teamがレジストを開発。  
DuPont Photomasks Inc.がフォトマスクを開発した。
- ③ Sematechのfundを受けた研究である。

## 2.2 MPU トランジスタ数

トランジスタ数は、ほぼ予想通りの推移をしていると考えられる。

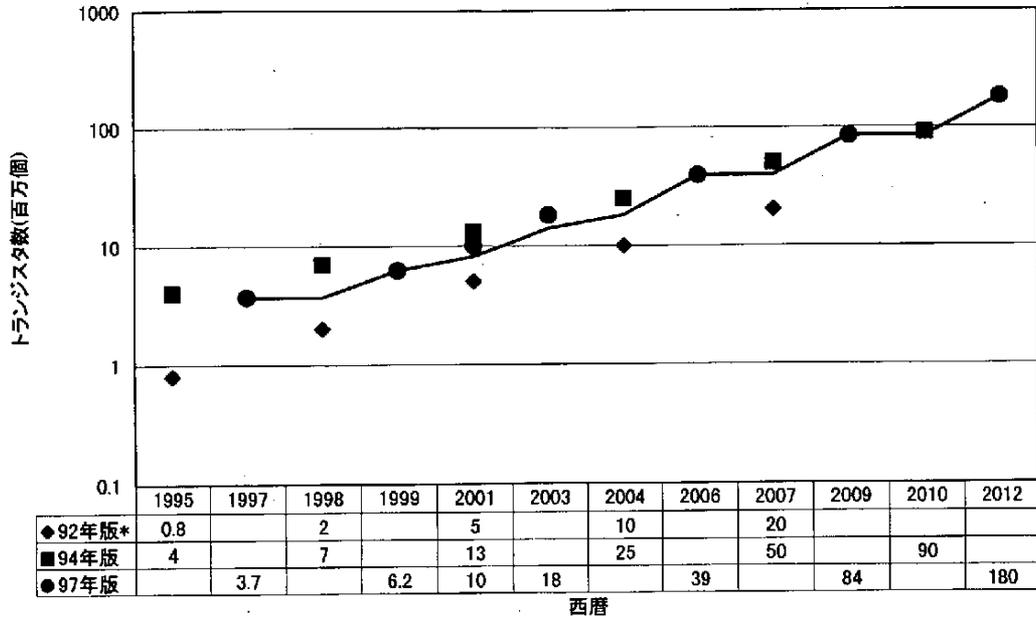


図5 MPU トランジスタ数

## 2.3 MPU チップサイズ

チップサイズの推移は、これまでの予想より大き目になるとされている。

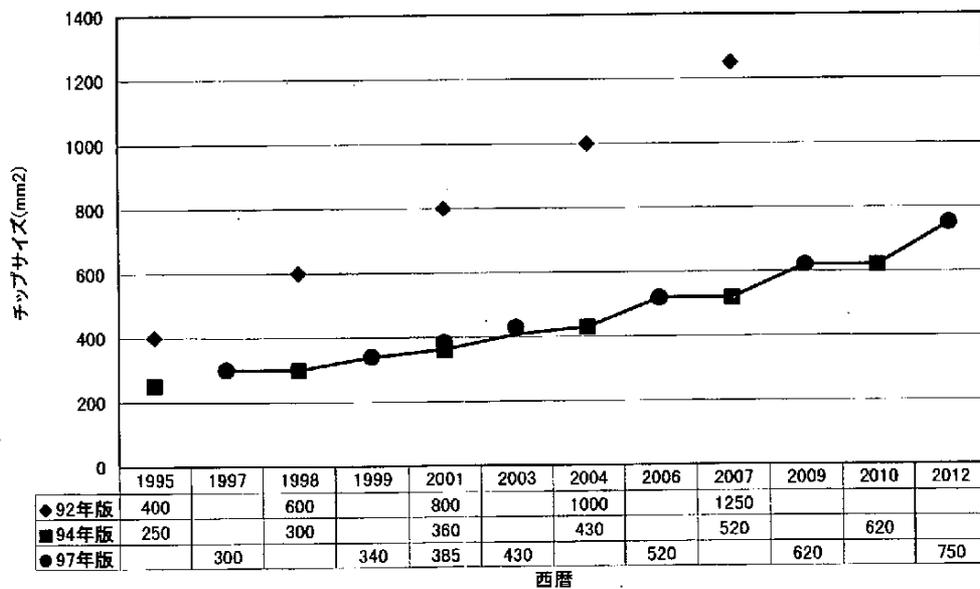


図6 MPU チップサイズ

### 2.4 MPU 層数

配線用酸化膜層数は、1997年版ではかなり前倒して増加している。これは、MPUの世代交代を早める要因になると考えられる。

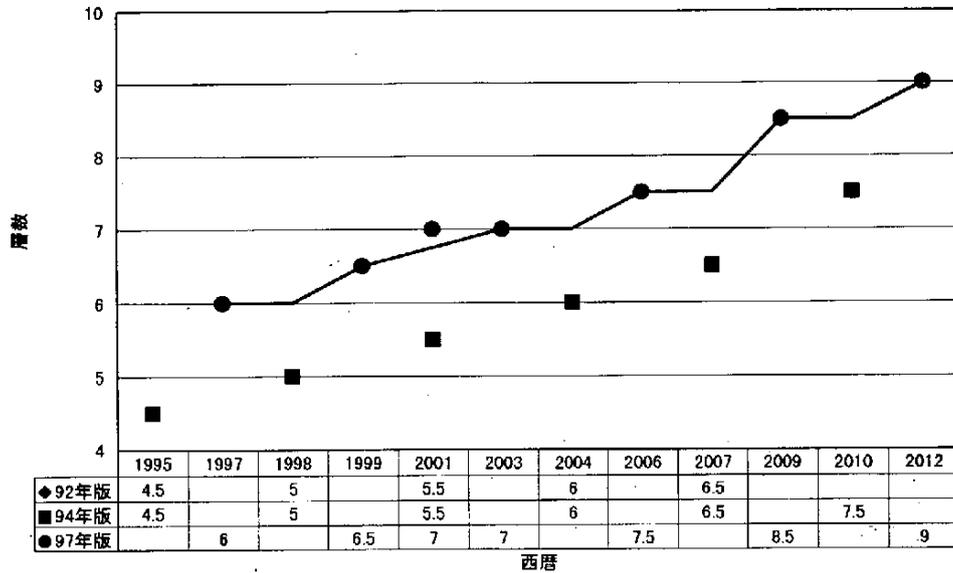


図7 MPU 層数

### 2.5 CPU クロック

MPU クロックの推移を図8に示す。97年版は、ハイエンドと商用の二種類に分離されたが、ハイエンドでは2倍以上、商用においてさえ、かなり加速され、2~3割以上の高速化が予想されている。

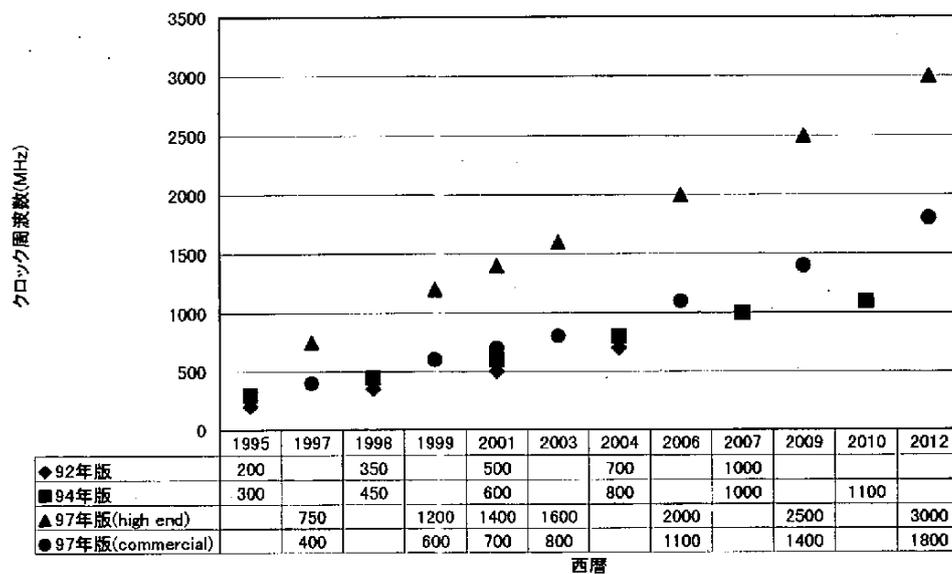


図8 MPU クロック周波数

## 2.6 MPU 動作電圧

動作電圧は、1年程度の前倒しで推移すると予想されている。

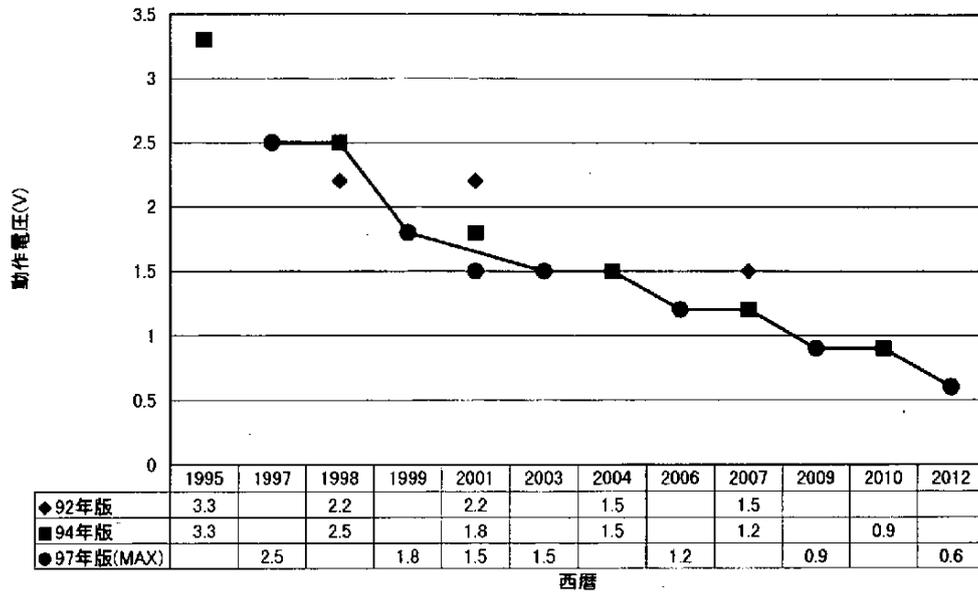


図9 MPU 動作電圧

## 2.7 MPU 消費電力

消費電力は、以前からの予想に比べ、5-10%の減少が予想されている。

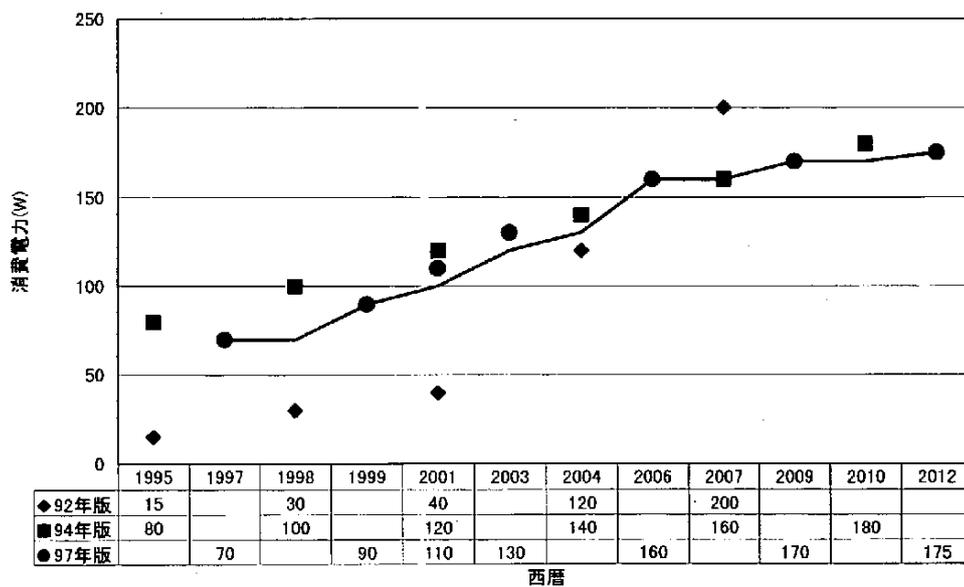


図10 MPU 消費電力

### 2.8 CHIP I/O 数

1997年版の高性能チップで比較すると、2007年位までは従来の予測より2割程度の増加が見込まれているが、その後急速に増加する。

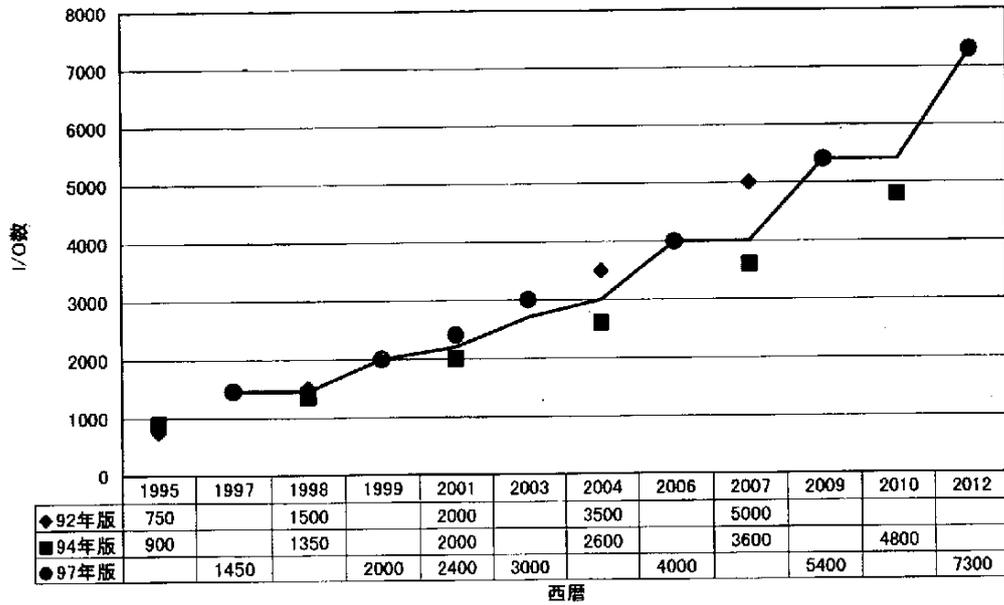


図 11 Chip I/O 数

### 3. ウエハサイズ

ウエハサイズは、従来予想とほぼ同じかやや大き目なると予想されている。これは、DRAM、MPUのチップサイズが大きくなることを反映している。

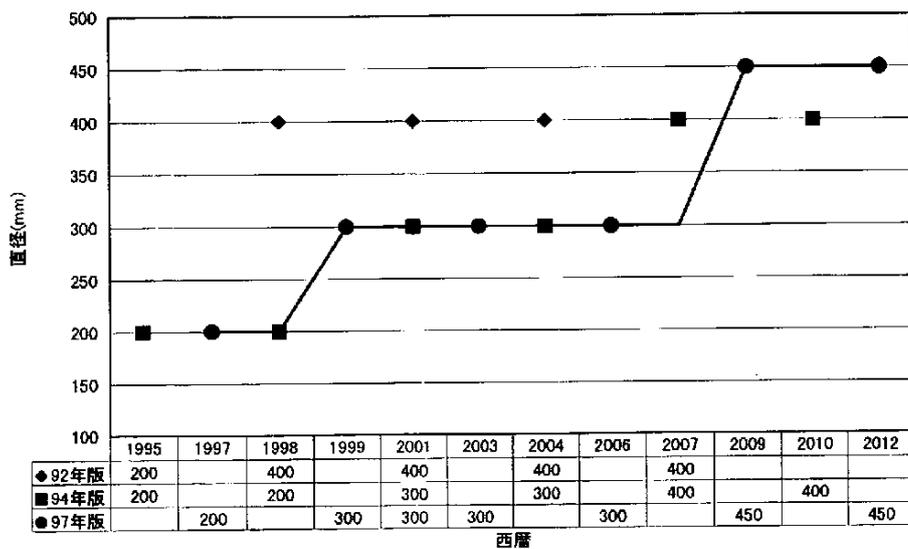
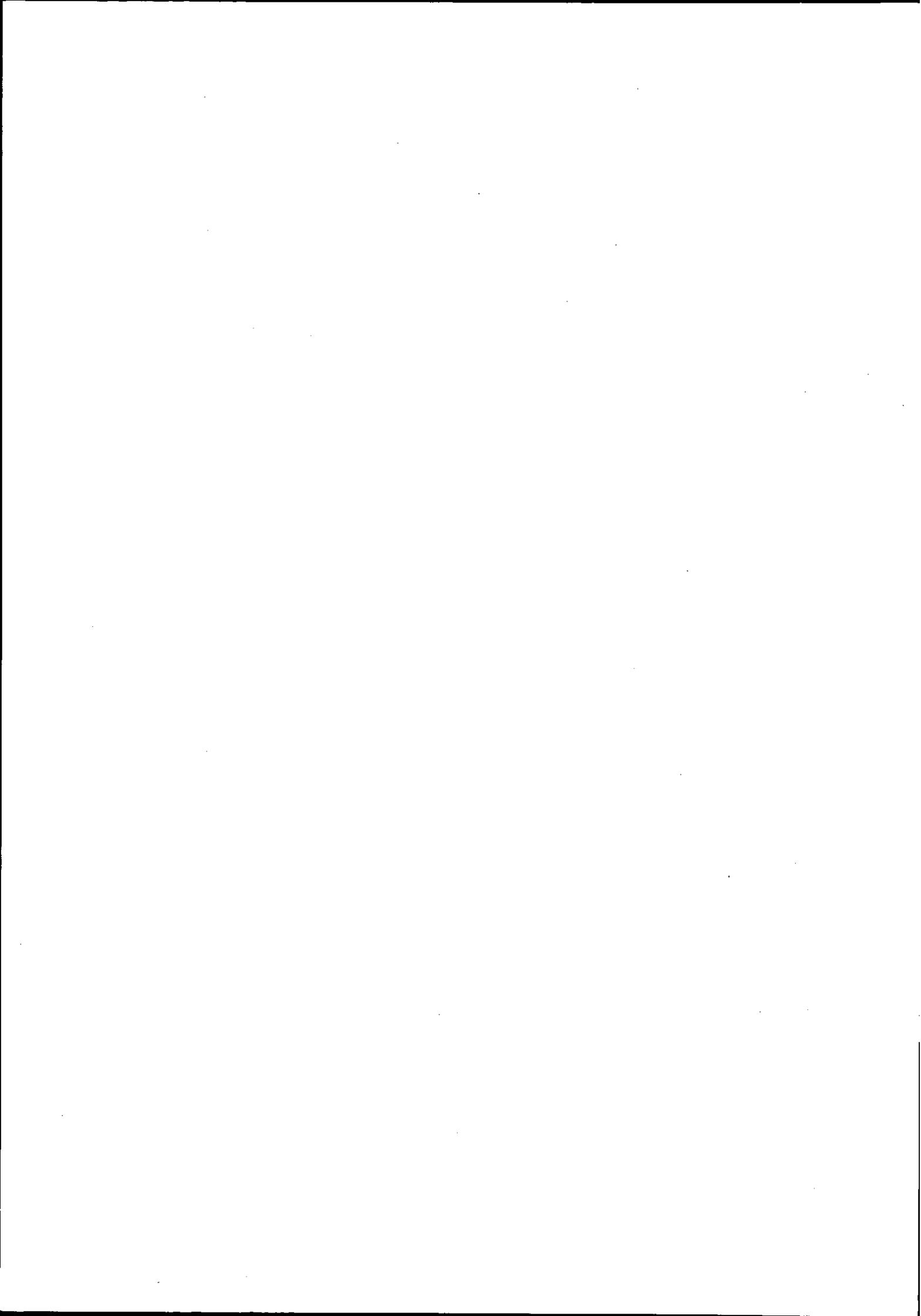


図 12 ウエハサイズ

## 付属資料 5 ワーキンググループ資料リスト

資料番号	資料タイトル	配付日	備考
PFM-WG-1001	ペタフロップスマシン技術調査 WG 資料	97.10.21	山口主査運営方針 他
PFM-WG-1002	「今あるものの次にくるべきものは何か、 または、何がきてほしいか」	97.10.21	天野委員提出
PFM-WG-1003	「今あるものの次にくるべきものは何か、 または、何がきてほしいか」	97.10.21	石川委員提出
PFM-WG-1004	「今あるものの次にくるべきものは何か、 何がきてほしいか」	97.10.21	笠原委員提出
PFM-WG-1005	「今あるものの次にくるべきものは何か、 または、何がきてほしいか」	97.10.21	関口委員提出
PFM-WG-1006	「次にくるべきもの/来て欲しいもの」	97.10.16	中島 浩委員提出
PFM-WG-1007	「今あるものの次にくるべきもの…」	97.10.16	中島克人委員提出
PFM-WG-1008	「今あるものの次にくるべきものは何か、 または、何がきてほしいか」	97.10.16	濱中委員提出
PFM-WG-1009	「今あるものの次にくるべきものは何か、 または、何がきてほしいか」	97.10.16	福井委員提出
PFM-WG-1010	「今あるものの次にくるべきものは何か、 または、何がきてほしいか」	97.10.16	横川委員提出
PFM-WG-1011	Hot Chips and Hot Interconnects for Hot Computer Systems	97.11.28	SC97 tutorial M5 資料
PFM-WG-1012	米国の研究開発の環境、制度	97.11.28	AITEC 提出資料
PFM-WG-1013	「今あるものの次に来るべきものは何か ～ペタフロップスマシンの将来～」	97.11.28	妹尾委員提出
PFM-WG-1014	「気象シミュレータ」	98.1.28	木本助教授 OHP コピー
PFM-WG-1015	「新プロセッサのロードマップ」	98.1.28	天野委員 OHP コピー
PFM-WG-1016	平成9年度報告書作成要項	98.1.28	AITEC 提出資料
PFM-WG-1017	並列コンパイラ研究課題	98.1.28	笠原委員 OHP コピー
PFM-WG-1018	Tera Flops マシンとアプリケーション	98.1.28	久門委員 OHP コピー
PFM-WG-1019	ソフトウェアの課題 (並列システム用言語処理技術)	98.1.28	妹尾委員 OHP コピー
PFM-WG-1020	SIA Roadmap -1997 Edition Appendix B	98.2.18	AITEC 提出資料
PFM-WG-1021	アプリケーションからのロードマップ	98.2.18	福井委員 OHP コピー
PFM-WG-1022	地球シミュレータ開発の概要	98.2.18	横川委員 OHP コピー





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Ⅱ

©平成10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4階

TEL(03)3456-2511

