

超並列計算機
CP-PACS
1992 — 2005

筑波大学 計算科学研究センター
2010年 5月



表紙

制作：金谷 和至

解説：背景は CP-PACS の CPU である HARP-1E の基板を
画像処理したもの.基板の実写真は写真 5 を参照.

超並列計算機

CP-PACS

1992 ~ 2005

筑波大学 計算科学研究センター

2010年5月



写真1 超並列計算機 CP-PACS (2048PU) 1996年9月



写真2 超並列計算機 CP-PACS (2048PU)

筐体カバーを取り外したもの

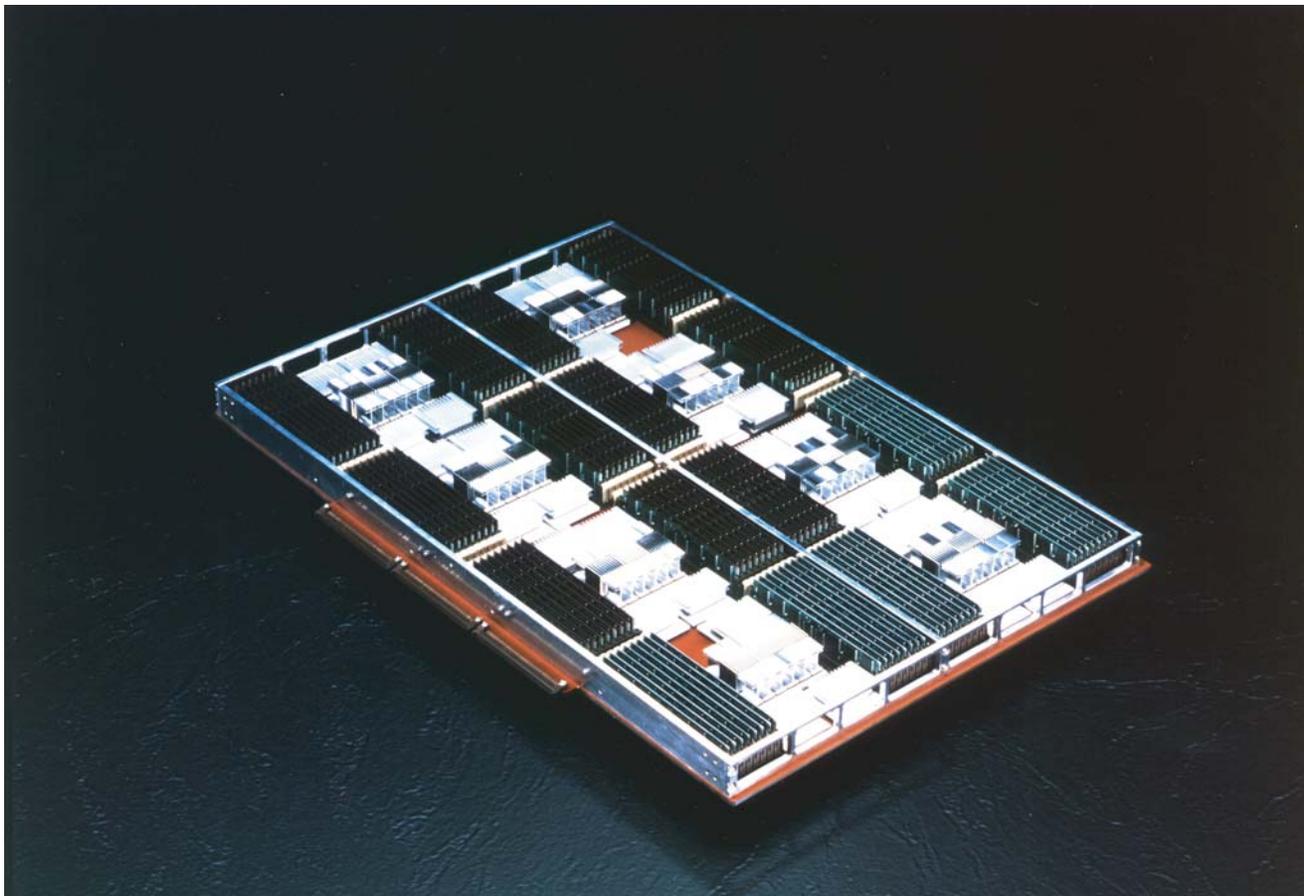


写真3 CP-PACSのボード 8PU搭載

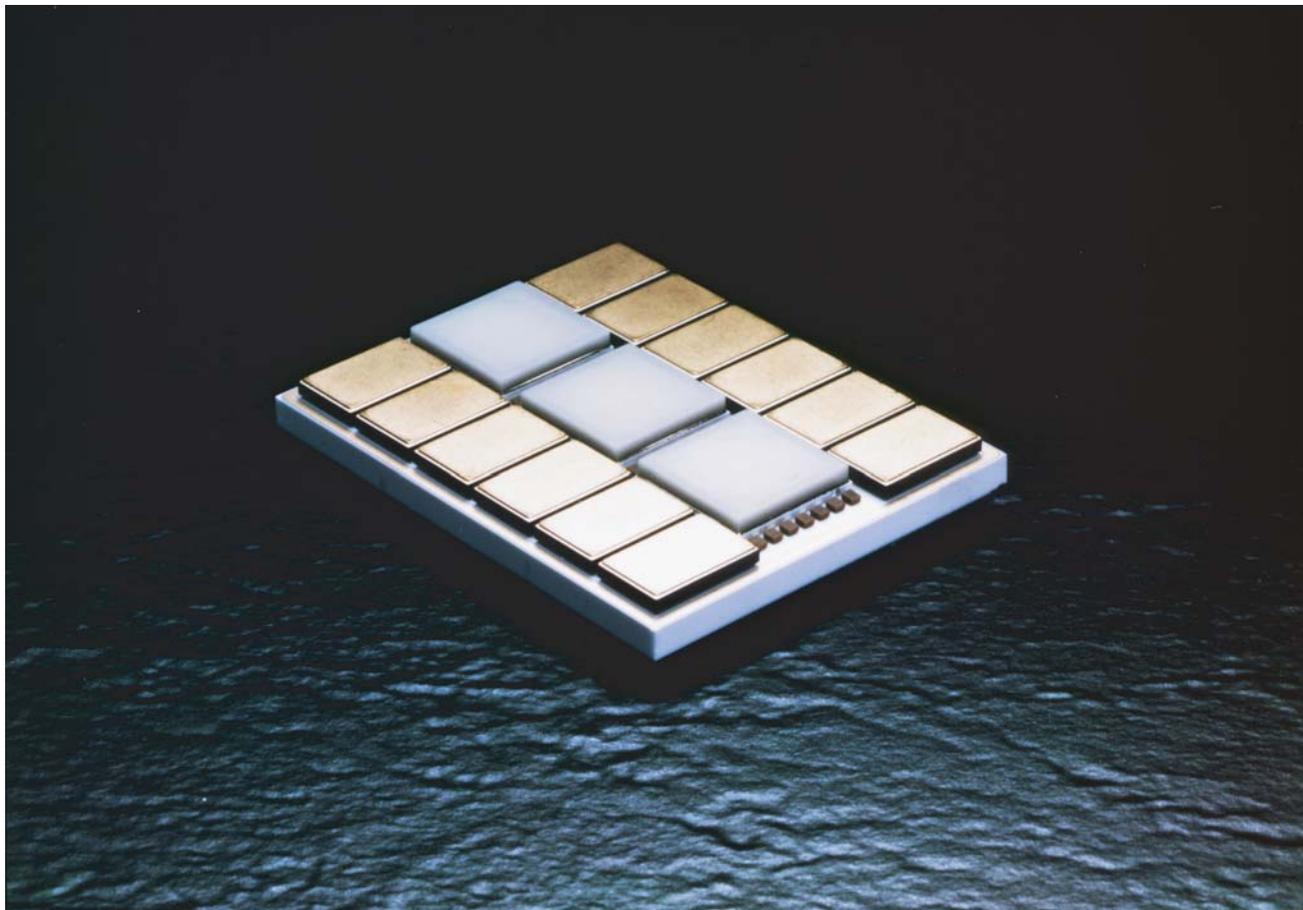


写真 4 CP-PACS の CPU HAPR-1E

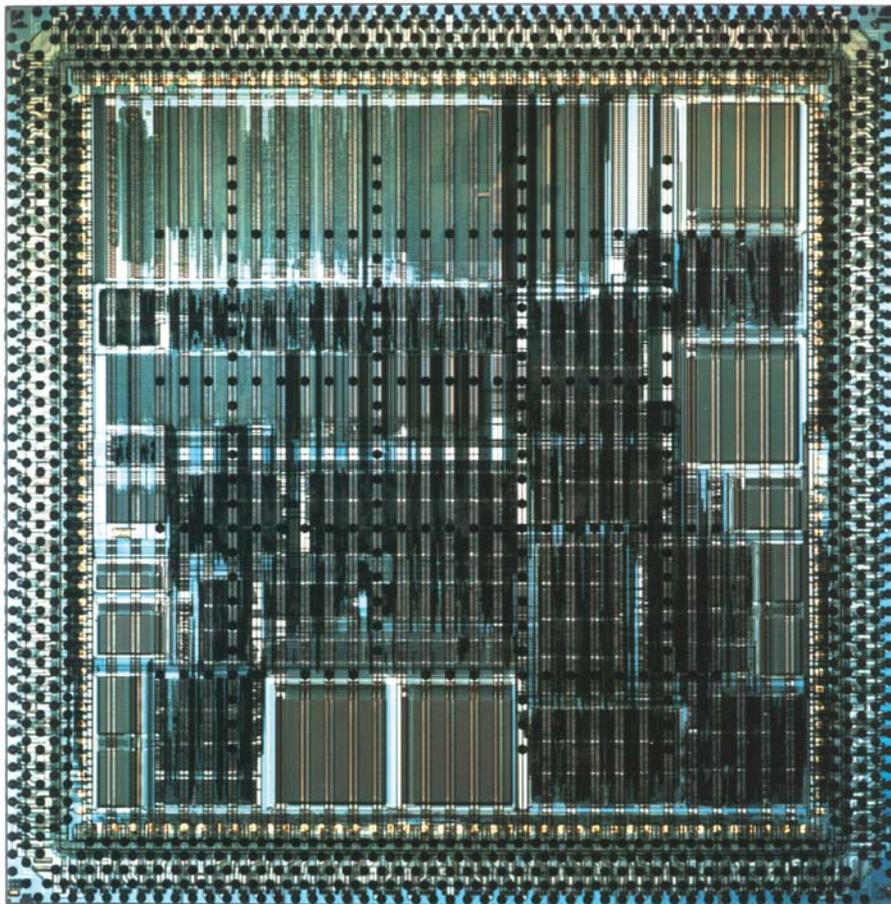


写真5 HARP-1E 基板



写真 6 CP-PACS の配線

List for November 1996

R_{\max} and R_{peak} values are in GFlops. For more details about other fields, please click on the button "Explanation of the Fields"

 EXPLANATION OF THE FIELDS

Rank	Manufacturer Computer/ Procs	R_{\max} R_{peak}	Installation Site Country/Year	Inst. type Installation Area	N_{\max} N_{half}	Computer Family Computer Type
1	Hitachi CP-PACS/2048/ 2048	368.20 368.20	Center for Computational Physics, Univ of Tsukuba Japan/1996	Academic	103680 30720	Hitachi SR2xxx CP-PACS
2	Fujitsu Numerical Wind Tunnel/ 167	229.70 281.00	NAL Japan/1996	Research Aerospace	66132 18018	Fujitsu VPP VPP500
3	Hitachi SR2201/1024/ 1024	220.40 307.00	University of Tokyo Japan/1996	Academic	138240 34560	Hitachi SR2xxx SR2201
4	Intel XP/S140/ 3680	143.40 184.00	Sandia National Labs USA/1993	Research	55700 20500	intel Paragon Paragon
5	Intel XP/S-MP 150/ 3072	127.10 154.00	Oak Ridge National Laboratory USA/1995	Research	86000 17800	intel Paragon Paragon SMP
6	Intel XP/S-MP 125/ 2502	103.50 125.10	Japan Atomic Energy Research Japan/1996	Research		intel Paragon Paragon SMP
7	Cray/SGI T3D MC1024-8/ 1024	100.50 152.00	Government USA/1994	Classified	81920 10224	T3E/T3D T3D
8	Fujitsu VPP500/80/ 80	98.90 128.00	National Lab. for High Energy Physics Japan/1994	Research	32640 10050	Fujitsu VPP VPP500
9	Fujitsu VPP700/56/ 56	94.30 123.20	Kyushu University Japan/1996	Academic	100280 8280	Fujitsu VPP VPP700/VPP300
10	Fujitsu VPP700/46/ 46	94.30 101.20	ECMWF UK/1996	Research Weather	100280 8280	Fujitsu VPP VPP700/VPP300

写真 7

1996年11月のTOP500リスト

CP-PACS が世界最高速と認定された

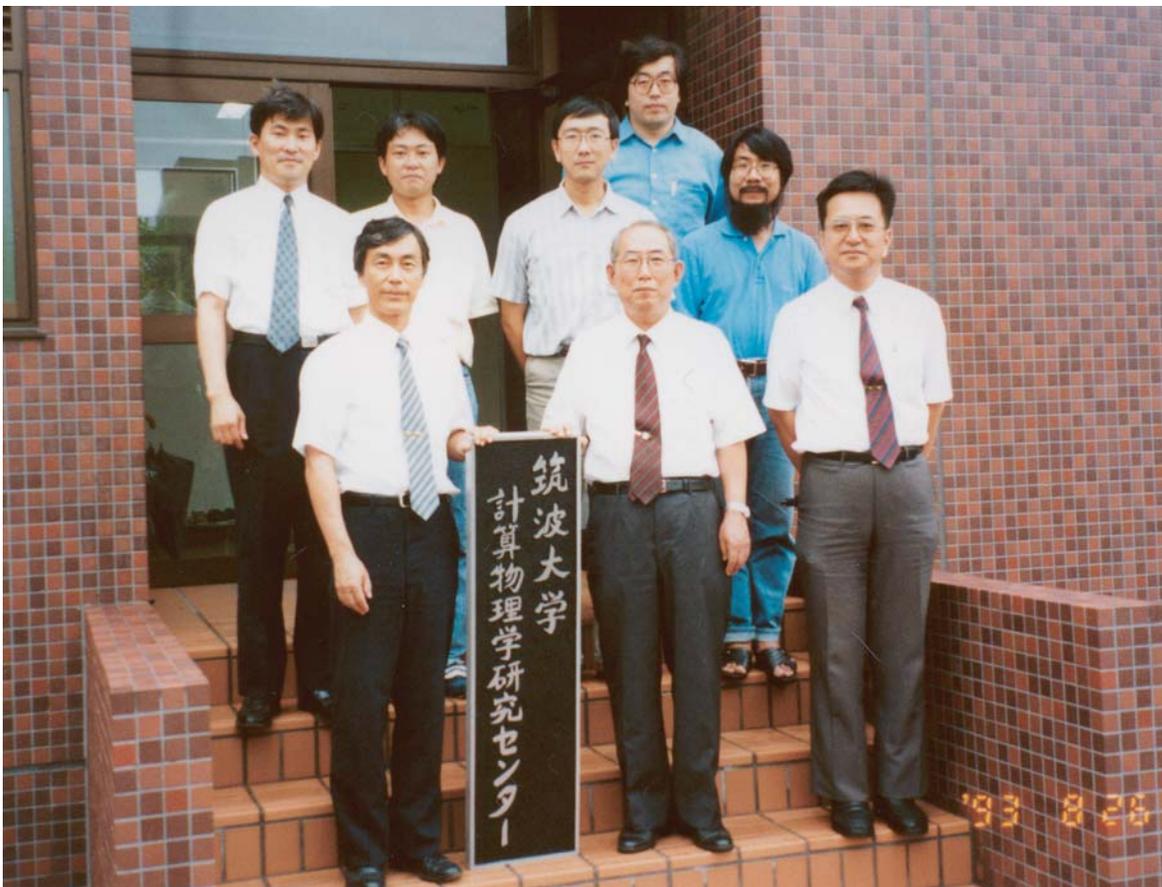


写真 8 計算物理学研究センター 1993年8月

前列左から：岩崎洋一，中澤喜三郎，小柳義夫

後列左から：中村宏，山下義行，金谷和至，朴泰祐，吉江友照



写真9 計算科学研究センター 1996年3月

前列左から：宇川彰，渡瀬芳行，星野力，中澤喜三郎，岩崎洋一，中田育男

中列左から：金谷和至，大川正典，小柳義夫，青木慎也

後列左から：安永守利，朴泰祐，中村宏，吉江友照，山下義行



写真 1 0 CP-PACS の組み立て 1 9 9 6 年 2 月



写真 1 1 CP-PACS (1024PU) の完成 1996年3月

前列左から：宇川彰，星野力，岩崎洋一，中澤喜三郎，中田育男，吉江友照，青木慎也
後列左から：坂井修一，朴泰祐，中村宏，渡瀬芳行，小柳義夫，金谷和至，大川正典，山下義行



写真 1 2 CP-PACS 稼働終了式 2005年9月

前列左から：渡瀬芳行，川合敏夫，星野力，宇川彰，岩崎洋一，中澤喜三郎，小林二三幸，佐久間嘉一郎，小平光彦，能沢健
中前列左から：富田雅，澤本英雄，河辺峻，小柳義夫，吉江友照，和田耕一，安永守利，中村宏，朴泰祐，金谷和至
中後列左から：浅野朋広，瀧田忠郎，高橋大介，荻山得哉，安崎篤朗，谷口裕介，三好一義，石川健一，白石賢二，橋本耕平，宮脇孝
後列左から：工藤紀之，谷嶋則幸，藤田不二男，佐藤三久，平下博之，原川竹氏，田中博，篠原敏一，伊藤洋志，板倉憲一

GP-PACS Jobs Used Time

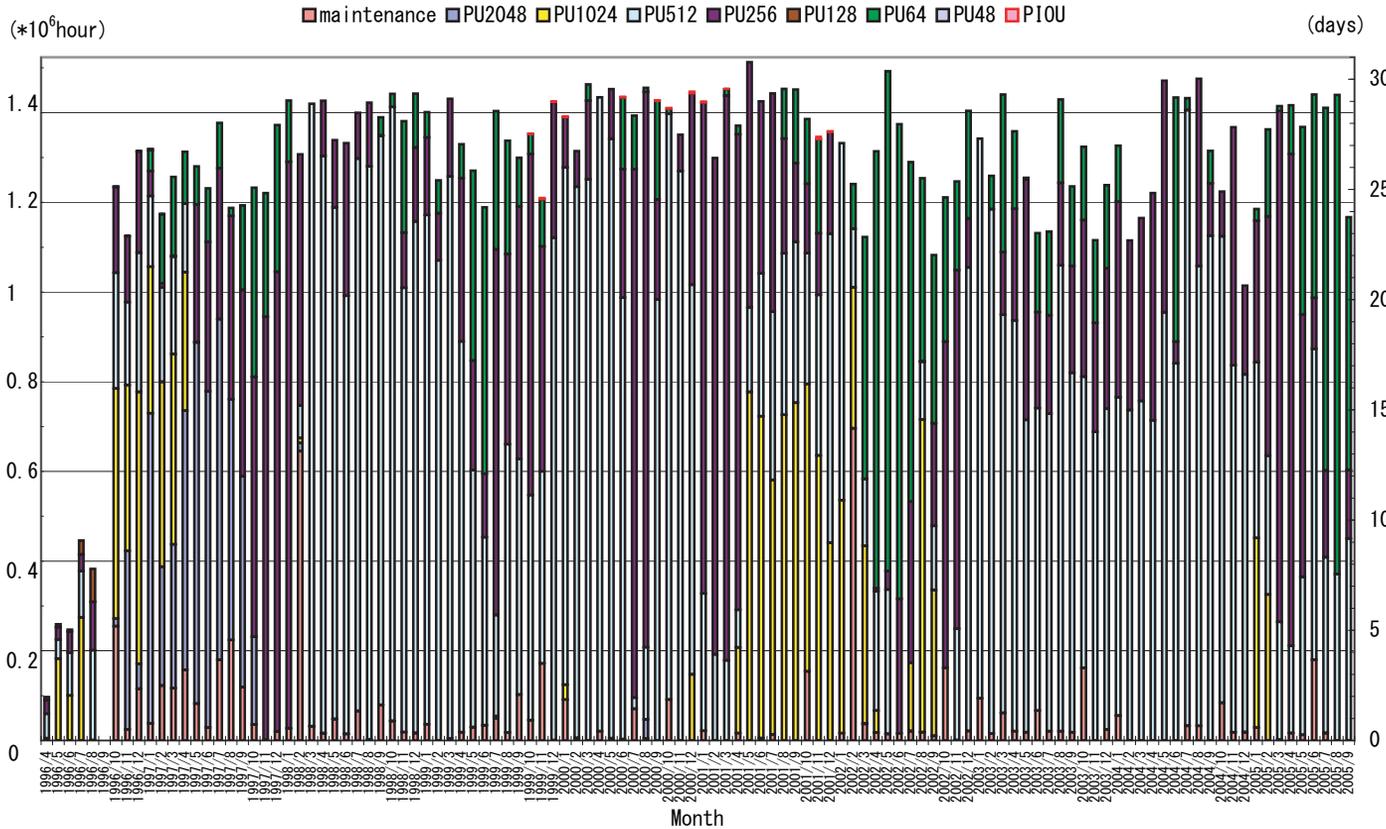


写真 1 3 CP-PACS 稼働履歴

毎月の実稼働時間を日数であらわしている。定期メンテナンスや大規模な調整・修理でシャットダウンしている時以外は、常に 85-90% 程度の高い稼働率を維持していた



写真 1 4 CP-PACS 展示模型 2006 年 2 月完成
計算科学研究センターエントランスホールで、QCDPAX と並んで展示中

序

「CP-PACSプロジェクト」は、1990年(平成2年)にその構想を立ち上げ、1991年に当時の文部省の「学術の新しい展開のためのプログラム」(略称新プロ)として採択され、1992年度に5ヶ年計画として正式に発足した。また、同年4月に、新プロの推進母体として、計算物理学研究センターが筑波大学に全国共同研究計算施設として設置された。

プロジェクトの目的は、物理学研究者と計算機工学者による密接な共同研究により、メーカの協力を得て、超高性能な並列計算機を設計開発・製作し、その計算機を用いて、物理学の研究分野で、「紙と鉛筆」では解くことが困難な、または実験が困難な問題を数値シミュレーションによって解決することにあった。「CP-PACS」は「並列計算機による計算物理学」を意味する、「Computational Physics by Parallel Array Computer System」の頭文字をとったもので、我々の開発・製作したコンピュータの愛称である。

CP-PACSは1996年9月に完成し、同年11月に発表された「世界スーパーコンピュータ500」の第1位にランクされた。このコンピュータが半分完成した時点から、素粒子物理・宇宙物理・物性物理などの研究は開始され、世界最先端の研究成果を数多く発表した。

その後、センターは大学の法人化と時を同じくし、研究分野を拡張し、陣容も強化し「計算科学研究センター」と名称も変更した。超並列計算機CP-PACSは2005年9月まで9年間稼働し続けて、その任務を終えた。その際に、CP-PACSを記念して、記念冊子を発刊する構想が持ち上がり、当初2006年6月発刊を目指していたが、諸般の事情により遅れていた。今回、計算科学研究センターが「先端学際計算科学共同研究拠点」と認定され、そのキックオフ・シンポジウムが開催されることになり、それに併せて記念冊子を発刊する運びとなった。この冊子の発案、編纂は、2005年当時のセンター長であった宇川彰氏によるものであることを記しその労苦に深く感謝したい。

このプロジェクトの詳細にわたる報告は、「専用並列計算機による「場の物理」の研究」科学研究費補助金(創成的基礎研究費)研究成果報告書

(計算物理学研究センター平成9年8月)にすでに載せているので、この「CP-PACS記念冊子」は、CP-PACSプロジェクトに関わった人々の個人的な回想、技術論、エピソードなどを記録し伝えたいとの趣旨で編纂された。関係者以外の方々にも、プロジェクトのことをより深く知っていただき、何らかの参考になれば幸いである。

最後になるが、このプロジェクトを支えて下さった文部省(後の文部科学省)、大学の事務職員並びに関係者の方々に改めて感謝の意を表したい。また、メーカの日立製作所の方々の誠意ある対応と確かな技術力なしには、このプロジェクトを語ることはできない。心から感謝したい。

岩崎 洋一
プロジェクトリーダー(当時)
平成 22 年 4 月 15 日

目次

序.....	1
写真解説.....	5
CP-PACS プロジェクト回顧録.....	岩崎 洋一 7
付記:QCDPAXプロジェクトでの体験.....	岩崎 洋一 19
CP-PACS の PVP-SW 方式誕生のいきさつ.....	中澤 喜三郎 24
CP-PACS とコンパイラの思い出.....	中田 育男 30
CP-PACS に教わったこと.....	中村 宏 34
CP-PACS プロジェクトに携わって.....	山下 義行 37
人生の転機になった CP-PACS プロジェクト.....	朴 泰祐 39
私に取っての CP-PACS プロジェクト.....	宇川 彰 45
CP-PACS の思い出.....	金谷 和至 52
CP-PACS プロジェクト終了にあたって思うこと.....	吉江 友照 56
CP-PACS プロジェクトの思い出.....	青木 慎也 61
CP-PACS プロジェクトと宇宙物理学.....	梅村 雅之 65
超並列計算への挑戦.....	中本 泰史 68
QCDPAX から CP-PACS へ.....	小柳 義夫 76
CP-PACS プロジェクトに携わって.....	河辺 峻 83
CP-PACS システムの開発を省みて.....	和田 英夫 88
CP-PACS との出会いと感動.....	荻山 得哉 93
CP-PACS 稼働終了.....	浅野 朋広 99
超並列計算機 CP-PACS 年譜.....	105
CP-PACS 関係の主要な成果発表.....	106
稼働終了後について.....	107
あとがき.....	109

写真解説

写真1 超並列計算機 CP-PACS (2048PU) 1996年9月

写真2 超並列計算機 CP-PACS (2048PU) 1996年9月
筐体カバーを取り外したもの

写真3 CP-PACS のボード 8PU 搭載

写真4 CP-PACS の CPU HARP-1E

写真5 HARP-1E 基板

写真6 CP-PACS の配線

写真7 1996年11月の TOP500 リスト
CP-PACS が世界最高速と認定された

写真8 計算物理学研究センター 1993年8月
前列左から:岩崎洋一, 中澤喜三郎, 小柳義夫
後列左から:中村宏, 山下義行, 金谷和至, 朴泰祐, 吉江友照

写真9 計算科学研究センター 1996年3月
前列左から:宇川彰, 渡瀬芳行, 星野力, 中澤喜三郎, 岩崎洋一, 中田育男
中列左から:金谷和至, 大川正典, 小柳義夫, 青木慎也
後列左から:安永守利, 朴泰祐, 中村宏, 吉江友照, 山下義行

写真10 CP-PACS の組み立て 1996年2月

写真11 CP-PACS (1024PU)の完成 1996年3月

前列左から:宇川彰, 星野力, 岩崎洋一, 中澤喜三郎, 中田育男, 吉江友照, 青木慎也
後列左から:坂井修一, 朴泰祐, 中村宏, 渡瀬芳行, 小柳義夫, 金谷和至, 大川正典,
山下義行

写真12 CP-PACS 稼働終了式 2005年9月

前列左から:渡瀬芳行, 川合敏夫, 星野力, 宇川彰, 岩崎洋一, 中澤喜三郎, 小林二三幸,
佐久間嘉一郎, 小平光彦, 能沢健
中前列左から:富田雅, 澤本英雄, 河辺峻, 小柳義夫, 吉江友照, 和田耕一, 安永守利, 中
村宏, 朴泰祐, 金谷和至
中後列左から:浅野朋広, 瀧田忠郎, 高橋大介, 荻山得哉, 安崎篤朗, 谷口裕介, 三好一義,
石川健一, 白石賢二, 橋本耕平, 宮脇孝
後列左から:工藤紀之, 谷嶋則幸, 藤田不二男, 佐藤三久, 平下博之, 原川竹氏, 田中博,
篠原敏一, 伊藤洋志, 板倉憲一

写真13 CP-PACS 稼働履歴

毎月の実稼働時間を日数であらわしている。定期メンテナンスや大規模な調整・修理でシャ
ットダウンしている時以外は、常に 85-90%程度の高い稼働率を維持していた。

写真14 CP-PACS 展示模型 2006年2月完成

計算科学研究センターエントランスホールで、QCDPAX と並んで展示中。

CP-PACS プロジェクト回顧録

岩崎 洋一

筑波大学物理学系／計算物理学研究センター（当時）

高エネルギー加速器研究機構（現在）

この小稿ではプロジェクトのエピソード的なことを主として記述するつもりである。プロジェクトの詳細は「正史」(研究成果報告書)を参照してほしい。

◎「天の時」, 「地の利」, 「人の和」

いま, CP-PACSプロジェクトを回顧すると, このプロジェクトほど, 「天の時」, 「地の利」, 「人の和」全てが揃ったプロジェクトは珍しいのではないかと思う。

・ 天の時:

スーパーコンピュータは大きな転換期を迎えていた。当時, スーパーコンピュータとはベクトル型のコンピュータを意味し, 計算科学の分野で著しい成果を挙げていたが, 半導体技術の制約などから, ベクトル型による一つのプロセッサのコンピュータの高速化は限界に近付いていた。一方で, プロセッサの小型化と安価化によって, 千台規模の並列計算機を現実的なものとして考えられる時代に入っていた。しかし, 並列計算機といっても方式は千差万別で, どのアーキテクチャを採用するかは自明なことではなく, まだ, 日米の主要メーカは製品化に踏み切れずにいた。

そのような状況下, 筑波大学では, 星野力と川合敏雄によって始められた並列計算機の開発の歴史があった。特に, その5代目にあたる並列計算機「QCDPAX」は素粒子物理学の研究のために, 1987年度から1989年度の科学研究費補助金(特別推進研究)(研究代表者:岩崎)を受け, アンリツ株式会社の協力を得て, 開発・製作され, その当時のいわゆるスパコン(スーパーコンピュータ)の3倍の最高計算速度を達成した。しかし, ソフトウェアの不備など, 一般ユーザには使いこなすのが困難であり, 改

善すべき課題がかなり残されていた。(QCDPAXプロジェクトでの経験に関しては付記に記した.)

さらに、ちょうど1990年度から、文部省による「新プロ」が開始された。このプログラムは、科研費(科学研究費補助金の略)(特別推進研究)などで成果を挙げ、さらなる飛躍が見込まれるプロジェクトに対して、文部省のあらゆる政策方式、すなわち、科研費、設備費、組織、特別研究員、外国人特別研究員など、を総動員して研究をサポートするという新しいシステムである。

まさに、「CP-PACSプロジェクトの時来たれり」、であった。

- 地の利:

筑波の地に人材が集結していた。筑波大学は、新しい構想のもと設置され、特に講座制を撤廃し、分野間の連携が取りやすいシステムを採用していた。また、新しい大学ということで、様々な有能な人材が揃い、先進的な分野で研究を推進する意気に燃えていた。

具体的には: 並列計算機のパイオニア 星野力、日本のスパコンの父 中澤喜三郎、日本のコンパイラのパイオニア 中田育男、KEK(当時高エネルギー物理学研究所)のスパコンを駆使し、素粒子分野で国際的な研究成果を挙げている宇川彰、そして「QCDPAX」のプロジェクトリーダーの岩崎。さらに、「QCDPAX」プロジェクトのメンバーである金谷和至、吉江友照、小柳義夫、白川友紀。中堅・若手として、素粒子物理の青木慎也、計算機工学の中村宏、朴泰祐、山下義行。途中から、宇宙物理の梅村雅之、中本泰史。これらの分野の異なるメンバーが一つのプロジェクトに何の制約もなく参加することができた。筑波の地では、大学のほか、KEKから素粒子物理の大川正典からもメンバーに加わった。

- 人の和:

素晴らしい人々の集団で、それぞれがお互いにリスペクトし合い、個々の能力を最大限発揮した。私は、単に、「旗振り」、「メッセンジャーボーイ」、「尻たたき」の役を果たしたに過ぎない。

一つだけ付け加えることにする。同じころ、欧米で、やはり素粒子物理学研究用の並列計算機が開発されていた。1年に一度、我々の分野の国際

会議が開催されている。そのような国際会議のとき、米国のコロンビア大学のプロジェクトリーダーであるノーマン・クリストとプロジェクトの進め方について話したことがある。私が、「なぜ、計算機工学の研究者と組んでやらないのか？」と聞いたときに、「最初は一緒に始めたのだけれど、両方の目指すゴールが異なるので、結局うまくいかなかった。」との回答を得た。

物理学者と計算機工学者の発想の違いなどは、「QCDPAX」プロジェクトでも痛感していたし、それなりに理学と工学の連携の「こつ」のようなものはもっていたつもりだが、この会話の後、特に気を付けたのは、理学のゴールと工学のゴールを両方持ち、それぞれの実現の時期はずれるが、二つのゴールを両方とも実現することをプロジェクトの目的にすることだった。

また、メーカ(日立製作所)の方々が、延べ数百時間に及ぶ打ち合わせに忍耐強く付き合ってくれ、我々とコミュニケーションを密にとり、我々の要求を実現して下さったことも大きい。

◎ CP-PACSプロジェクト構想の「立ち上げ」から「採択」まで

並列計算機QCDPAXは、様々な障害を乗り越え、プロジェクトの最終年度である1989年度に完成し、最終日1990年3月31日には、素粒子物理の計算を本格的に開始していた。関係者に対する感謝の意を表すためにも、記念式典開催を考えている間に、一部の新聞にスクープされてしまい、急遽4月6日に記念式典と新聞発表をすることにした。有馬朗人氏(当時東京大学総長)、菅原寛孝氏(当時高エネルギー物理学研究所所長)などお忙しい中、駆けつけて下さった。記念式典を午後控え、午前中、プロジェクトのメンバー一同、研究室に集まり、やっとここまでたどり着いた達成感と安堵感にたっぷりと浸っていた時に、「新プロというものが今年度から発足した。次のプロジェクトはそれで考えたらどうか？」との一本の電話がかかった。3年間の苦勞、特に最後の1年の筆舌に尽くしがたい辛苦を乗り越え、やっと、本来の物理の研究にしばらく、じっくりと取り組もうと思っていたが、この電話で状況は一変した。すぐさま、団欒の場を対策会議に変更せざるを得なかった。結論は当然「次のプロジェクトは新プログラムとして筑波大学を中心にやる」となった。QCDPAXコンピュータ完成式典

その日に、次のプロジェクトが立ち上がったことに、何か運命的なものを感じる。

まずは、「新プロ」とはどのようなものなのかを知ることから始めなくてはならなかった。「新プロ」の趣旨は前述のとおりであることはすぐに理解できたが、通常の科研費などがボトムアップ方式であるのに対して、トップダウン方式であることが経験したことのない点であった。科研費（特別推進研究）などで成果を挙げ、さらなる発展が期待できるものを、学術審議会の委員会のメンバーが推薦し、審査するという方式であった。

我々のプロジェクトは要件を満たしていると自らは判断していたが、申請することはシステム上でできなかった。しかし、幸い積極的に推薦してくれる方がいて、最終的に推薦件数7件の中の1件となった。そこまで辿り着くのに、これだけ大型のプロジェクトになると、紆余曲折があったが、我々としては、関係者からの助言を得ながら、プロジェクトの内容をつめることに専念した。また、QCDPAXでも物理の結果が順調に出てきており、QCDPAXのことが科学誌やビジネス誌などでもとりあげられるようになっていった。

推薦されたプロジェクトに対して書類審査があり、7件が2件に絞られ、2件に対してヒアリングが実施され、そのうち1件が採択されることになったとの通知があった。我々のプロジェクトと、もう1件であるが、それは強敵であるとのことであった。ヒアリングには万全の準備をして臨んだ。厳しい質問にも適切に回答したつもりであった。1週間後位に、我々のプロジェクトが採択されたとの通知を受け、関係者一同大いに喜んだことは言うまでもない。（後日談があり、3ヶ月後位にもう1件も追加採択になったとの情報を得た。）

◎ センター新設

前述のように、新プロは、文部省のあらゆる政策方式、すなわち、科研費、設備費、組織、特別研究員、外国人特別研究員など、を総動員して研究をサポートするという新しいシステムである。

この「組織」という用語がぎりぎりまで私を悩ました。新プロのことが話題に上がるようになってからは、大学の執行部、すなわち研究担当副学長、

研究協力部長などに情報を上げ、相談するようにしていた。新プロの施策として「組織」があり、「組織」にはセンターを含むことも分かっていたので、是非「組織」を要求したいと持ちかけても、筑波大学には特有の「特別研究プロジェクト組織」(特プロ)があり、研究員のことは特プロで対応するから、新プロとして「組織」は必要ないという執行部の回答であった。いよいよ、新プロのヒアリング対象として選定されたあと、新プロを所管する文部省の学術課に我々の案を説明にいくと、「組織」はどうするのですかと必ず聞かれた。大学の執行部の考え、「特プロで対応」、を伝えると、なにか腑に落ちない顔をされる。なにかミスマッチがあると直感的に思い、事務局長にまで面会に行ったが、同じ回答であった。

思案に暮れているとき、ある人のことを急に思いついた。筑波大学は前にも述べたように新しい大学であり、新設間もないころは、レストランも飲み屋もない中で、事務職員と教員は同じ公務員宿舎に家族と共に暮らし、他に娯楽もないので、仲間でテニスを共に興じた。その中の一人が当時、主計課課長補佐を担当していた。彼のところに行き、状況を話すと、「先生、それは重要な話です」と顔色を変え、そのまま文部省に飛んで行ってくれた。そして、その日のうちに、新プロを所管する学術課とともに、特プロを所管する大学課(別の局)の両方と面談してくれた。

その結果、学術課は「新プロでセンター新設の可能性がある」、大学課は「新プロと特プロの2本を並列に出すことは極めて困難である」と考えていることが判明した。その情報を基に、私としては、すぐに、新ブロー本に絞って、センターの設置を要求することを決断した。ヒアリングが設定されていたのが2月28日で、センター新設を要求する判断をしたのが、ぎりぎり1週間前の21日であった。

急遽、センター新設の件でアポイントをとり、文部省に赴いたのが25日。会議室に入ると、それまでは、1、2名で対応してくれていたのだが、その日は、学術課、研究機関課、研究協力室などの総勢10名くらいで、私を囲むような配列であったことにまず圧倒された。それから、生まれて初めての尋問のような厳しい質問が次から次へと出された。特に、その中でも、当時の研究機関課課長補佐は厳しいことで、省内でも有名であったとのこと。

その厳しさに少々とまどいながらも、1年近く練りに練った計画であったし、QCDPAXプロジェクトの実績もあったので、きちんと適切に対応できたのではないかと思っていた。

とくに、「もし、センターが新設されなかったら研究の方はどうしますか？」という質問に対しては、科学者らしく答えた。「3つの場合が考えられます。」1番目は、センター設置が認められた場合。2番目はセンター設置が認められなく、大学の「特プロ」が認められた場合。3番目として、どちらも認められなかった場合。それぞれの場合の、研究に対する影響を述べ、3番目のケースは、世界と戦っていくにはとても厳しいが、最善を尽くし、研究は続けていく。と答えた。これに対し、「先生のような回答は初めてだ。ふつつ、センターができないと研究が続けられなくなる、とこたえる。気に入った。」と言ってくれた。厳しい質問は続いたが、後になって、厳しい質問は、局内、省内、大蔵省(当時)で承認を得るためのものだとやっと理解できた。さらに、だいぶ後になってから、その時の研究機関課補佐から、あの時の答えを聞いて、先生と心中できると思ったという言葉までもらった。

これで、新プロの申請の中に、センター新設の要求を書き込むことがやっとなできるようになった。間一髪という感じであった。センター新設の可能性は高まったが、これで要求が認められた訳ではない。予算要求が認められて、初めてセンターが設置できる。まだ、道半ばである。

◎ 予算要求

新プロの科研費は新設の科研費(創成的基礎研究費)から措置される。我々としては、当然、開発・製作するコンピュータは世界最高速のものを目指すつもりであった。通常、メーカーが新規に大型コンピュータを開発・製作するには数100億円が必要であるが、文部省の予算規模は1桁少ないということは分かっていた。そこで、我々は、並列計算機に関する我々のノウハウと人的貢献を最大限に大きく見積もってもらい、メーカーはその並列計算機をベースに商品化して将来赤字分を取り戻してもらおう、という虫のよい算段をするしかなかった。しかし、最低限の半導体などの材料費などをカバーする額は必要であろうと考えた。世界最高速を狙うには、600GFLO

PS程度のコンピュータを開発する必要があり、そのためには、どんなに低く見積もっても、25億円は必要であるとの結論を得て、その計画書を最初は提出した。しかし、文部省の予算の枠はどんなに頑張っても、10億円規模であるとのこと。そこで、文部省とだいたい交渉したが、どうしても総枠からして無理であるとのこと。結論として、格別の額として、15億円を5年間で措置するという事になった。文部省としてもぎりぎりの線だったと思う。そこで、当初目標として、性能が半分の300GFLOPSのコンピュータを開発・製作することとした。もちろん、世界最高速を諦めたわけではなかった。

センターの新設の可能性が高くなり、それに伴い、定員、建物、設備、運営経費などを要求できることになった。定員はその当時ですでに厳しく、4名の純増で、6名は学内振替となった。

新しい建物は、文教施設部から当時の情勢からいって到底無理であるから、学内で既存の場所の中から探しておくようにとの話であった。これはこたえた。適した部屋などあるはずもなく、共同研究棟のうす暗く、柱などの位置がコンピュータの設計を制限するだろうし、電源もない部屋を、中澤氏と一緒に見に行き、暗澹たる気持ちになったことを今でも覚えている。ところが、ふつうはあり得ない話らしいが、研究機関課が文教施設部に強く働きかけてくれ、計算機棟だけは、1992年度の補正予算で認められた。青空が広がったような気持ちであった。後に、研究棟も認められた。

設備も、空調、変電施設、落雷などによる停電時のための無停電装置など、裏方に相当する設備が極めて重要であること、また金額もばかにならないことなど、私としては初めて知ること多かったが、中澤氏のおかげで、建物の設計も含め、順調に見積もることができた。

運営経費としては電気代が大きな部分を占める。コンピュータ本体と空調分である。しかし、初めのうちコンピュータは存在しないので、電気代はきわめて少なくて済む。その後、年度ごとに少しずつ増えていき、コンピュータが完成した時にピークの額が必要になる。これを年次計画として、綿密に積み上げて積算した。具体的な計算は、かなりの時間をかけて、主計課が行った。

運営費も、設備費も、毎年正直に積み上げて予算要求をした。先輩にこ

の話をしたところ、ばかだな、年次計画など丸めて、予算要求は2割増しで要求し、2割減で査定されるものだよ、と言われた。しかし、我々のこの真摯な態度が分かってもらえたのだろうか、5年間に毎年かなりの額を要求したが、一度も減額されたことはなかった。全面的にサポートしてもらった気がしている。

予算要求を概算要求と呼ぶこともこの時学んだが、それと同時に、その作業がどれだけ大変かということ、大学の事務がどんなに頑張っているかを初めて知った。2月末にセンターの話が出て、6月末には大学から概算要求書として正式に文部省に提出する必要がある。その間、2週間ごとに、新しい資料を作成し持参し、文部省に行き、説明した。土日なく、宇川氏といかに分かり易い書類を作成するか腐心した。また、事務に聞かないとわからない点が出て本部棟にいくと、土日関係なく、主計課の人が必ずいて仕事をしているのを目のあたりにし、認識を新たにした。3月から6月にかけて、中澤氏、宇川氏らと一緒に文部省に出向いたが、研究機関課課長補佐を中心に、相変わらず厳しい質問を浴びせられた。これが、我々の要求を何とか通そうとする「愛の鞭」であることが分かったのは、前述のとおり、だいぶ後になってからだった。

文部省に概算要求書を提出した後、7月になってからも、次々とポンチ絵と説明文などの要求がきた。土日もほとんど休んでいなかったのでも、夏休みを8月に入ってから月曜日1日だけとり、週末と合わせて3日の休みをとった。旅先まで、電話がかかってくるだけでなく、月曜日やつくば市に戻って、大学に電話をすると、相談したいことがあるから、大学まできてほしい、とのこと。お盆前が文部省内部の山場であることをこの時知り、その後4年間は夏休みも取らなかった。

◎ 調達手続き

新プロが採択された頃は、いわゆる「日米スパコン摩擦」が政治問題化していた。そのような状況を考えて、我々は、協力してくれるメーカーを選定する調達手続きを透明化、公正化することに努め、当時のスパコン購入の手続きに準ずることにした。ただし、プロジェクトの開始時期を考え、期間を

少し短縮することにした。

この手続きに関して、関係官庁の意見を聞いておいた方がよいと判断して、通産省機械情報産業局電子機器課課長とアポイントをとり、中澤氏と一緒に面談した。30分の予約であったが、15分位話したところで、「分かりました。開発・製作するということですね。その手続きでまったく問題ありません。」と会談を終了した。問題がないということの確認がとれたので、ほっとして、道路を挟んだ文部省に向かった。やはりアポイントメントをとっておいた。行ってみると、学術課、学術機関課に会計課の人を加えて10人くらいの人が待っていてくれた。2時間くらい会談をしたと思うが、その日に我々の話したことを文書化して提出するという宿題をもらって帰ってきた。その後、何回か説明資料を提出し、2か月掛って問題ないとの回答を得た。この「日米スパコン摩擦」は通商問題であり、文部省の通常の所管ではないので、慎重な対応であったのであろう。

1991年4月2日に14社(内米国系3社)に対して開発協力可能性を打診し、その後、全体計画書提出、説明会、官報公示などを経て、一般競争入札で日立製作所と契約を締結したのが、1992年6月2日であった。

メーカーとのやり取りを含む関係資料はすべてファイルに綴じておき、それが地球シミュレータ・プロジェクト発足時に役に立ったことを付記しておきたい。

◎ プロセッサ数の倍増

当初、600GFLOPS程度の世界最高速のコンピュータを目指していたが、予算の関係で、300GFLOPSの目標に変更せざるを得なかったことは、前述のとおりである。しかし、完全に当初の目標を諦めたわけではなかった。文部省の新プロを統括する学術課にも、そのことは暗黙のうちに伝わっていた。プロジェクトが発足してだいぶ経ったころ、総額25億円は無理だが、ぎりぎり22億円までなら可能性が全くないわけではない、ということになった。有難い話である。並列計算機であるから、プロセッサ数を1000個(正式には1024)から、倍の2000個(2048)に増やせば、性能は倍増する。まだ製造段階に入っていなかったもので、この点は、全く問題なか

った。

しかし、メーカーにとっては、25億円でもぎりぎりの線なのに、簡単にのめる話ではない。また、コンピュータのプロセッサ数が倍増するということは、電源も、空調も、倍増に近くなる。センター設備費は研究機関課の所管である。

メーカー、学術課、研究機関課の3体問題になった。文部省の2つの課の対応は、メーカーが製作してくれることが当然の大前提である。また、2つの課は、それぞれ、「先生の話はよく分かりましたが、もう一方の課が認めてくれるなら。」という対応で、一向に前進しない状況が長く続いた。2つの課は壁一枚で隔たれているだけだったが、その壁は思いの外高かった。一方、メーカーの反応もはかばかしくない状況が続いた。メーカーの幹部にまで会談にいったが、「検討しましょう」という回答であった。これでは、文部省に強く働きかけることもできず、三方を同時に解決しなければならない状況が長く続いた。その年の概算要求の実質的な最終日、予算の取りまとめをする研究機関課は忙しくアポイントも取れなかったが、研究機関課課長補佐(異動で1991年とは異なる人)が、会議と会議の間、廊下を歩くところを捕まえて、最後の「交渉?」を行い、やっと認められた。また、メーカーの幹部との会談の3回目もほぼ同時に開かれ、メーカーとして極めて厳しい判断だったと思えるが、最終的に2048プロセッサのコンピュータを製作することに合意してくれた。この間約10ヶ月ほどかかった。その時の補佐と今でも会うと、「あの時は先生の熱意に負けました」と、その時の情景が話題になる。

◎ プロジェクト遂行とプロジェクトからの教訓

このようにして、CP-PACSプロジェクトは発足し、遂行された。そして、所期の目的を達成することができた。

CP-PACSプロジェクトからの教訓は、QCDPAXプロジェクトからの教訓を再確認し強化するものであるが、コンピュータを開発するときに極めて重要な点は、そのコンピュータを用いて何を解き明かしたいのかの大目標(グランドチャレンジ)をまず最初に設定し、それを目指して、概念設計、基

本設計，詳細設計，製作することである。コンピュータ本体のみならず，データの入出力，周辺機器を含め，アプリケーションを念頭に置くことにより，具体的な設計目標が定まる。それを目標に，計算工学者とメーカは新機軸を考案し，技術的な困難さを克服する強い動機を持つことができ，結果として，ユーザにとって使い勝手がよく，高性能なコンピュータが完成する。この際，物理学科学者と計算機工学研究者とメーカの共同研究，協力が必要不可欠である。そのためには，異分野の研究者が日常的に直に顔を合わせながら，気軽に議論できる「場」が必要である。我々の場合，計算物理学研究センターがその「場」であった。

「専用並列計算機」を開発・製作するということは，結果として，「汎用並列計算機」を開発することに繋がり，最初から「汎用」を目指す構想は纏まりにくい。これは，逆説的に聞こえ，考えにくいかもしれないが，ベクトル型のコンピュータから並列計算機への転換のように，パラダイムシフトが起こる時には，この点が重要な点である。目標を明確にするという思想は，星野・川合が原子炉内燃焼をシミュレーションしようとしてPAXシリーズを始めた時まで遡ることができる。さらに，QCDPAXは素粒子物理学でのシミュレーションを目指して開発された。そしてどれもが，コンピュータとしては汎用性をもつものができた。自然現象が基本的に近接作用であることによる。

CP-PACSプロジェクトも，当初は素粒子物理学の格子量子色力学の計算を大目標の一つに掲げた。素粒子物理学研究者の宇川，金谷，吉江，青木が，その計算の核に当たるルーチン，我々が「MULT」と呼んだ部分を，抜き出した。「MULT」をいかに速く実効できるかが，コンピュータの実効速度を実質的に定める。キャッシュによるプロセッサでは10%程度しか性能が出ないことが判明した日が「暑気払い」に当たっており，ビールを飲めば飲むほど苦く，暑くなる経験を初めてした。しかし，夏休みが終わると，計算機工学のハード担当の中村，中澤が見事な解決策を見出していた。ソフト担当の中田，山下がさらに工夫を重ね，最終的に疑似ベクトルスライドウィンドウという新機軸に結実した。この点に関しては，それぞれ担当者から具体的な話があることと思う。それに基づき，素粒子物理学者が「MU

LT」を最適化したアセンブラコードで書き下し、60%以上の効率を実現することができた。これが、掲げた目標に向けての物理学者と計算機工学者の共同研究の典型的な例である。ターゲットを明確にしていたからこそ、この新機軸も生まれたのだと思っている。プロセッサとメモリ間のスループット、ノード間のネットワークのスループットに対する要求性能もこの「MULT」から導かれた。計算結果のデータの入出力は宇宙物理からの要求を基に、並列IOと大容量の外部記憶装置を装備することになった。もちろん、メーカはこれをベースに商品化を考えていたから、他のアプリケーションでのクロスチェックは当然行っていたと思うし、実際に後に商品化された。これが、私の言う、「専用」が「汎用」に繋がるの意である。

並列計算機の黎明期であったからこそ、並列計算機のアーキテクチャは多様であり、選択の余地が余りに多く、最初から「汎用」を考えると、具体的にアーキテクチャを絞り込むのが困難であった。「汎用性」を目指したプロジェクトで、「実用的」な並列計算機を開発したプロジェクトは当時見当たらなかった。

大目標を最初に掲げるという開発方針の結果として、計算機工学者のゴールと物理学研究者のゴール、2つのゴールをともに実現することができた。1つは「世界一高速で、幅広い分野のユーザに使い勝手のよいコンピュータを開発・製作する」というゴールである。もう一つは、「そのコンピュータを用いて、物理の分野で世界最先端の成果をあげる」というゴールである。CP-PACSが稼働し続けた2005年まで、素粒子物理・宇宙物理・物性物理で、世界最先端の研究成果を創出し続けた。とくに、素粒子物理学の分野では、クォークの理論である格子量子色力学によるハドロンと総称される質量の計算で、1983年以來の懸案であったクエンチ近似での結果に対して最終的な決着をつけ、さらに近似のない計算でも世界をリードする結果を導いた。

プロジェクトを遂行していく間の様々な光景が思い出され、また創出された物理の結果に関して強い思いもあるが、他のメンバーの回顧録に委ねたい。

付記:QCDPAXプロジェクトでの体験

この付記では、なぜ、理論素粒子物理学者にすぎなかった筆者が、並列計算機を開発するプロジェクトに関与することになったのか、そのいきさつから、その顛末を記すつもりである。

筆者が並列計算機に関係することになったきっかけは、1985年に、小柳氏から星野氏を紹介され、科研費(特別推進研究)に申請するので、研究分担者にならないかという話があった時だ。その判断をする前に、並列計算機に一度触れておく必要があると思い、PAXシリーズの4代目にあたるPAX-32Jを見学に行った。

PAXシリーズは、原子力工学を専門にしていた星野氏に、原子炉内燃焼のシミュレータとして、川合氏が、マイクロ・プロセッサのアレイを提案したことがきっかけで開始され、4代目までのPAXは星野氏が中心となって開発された。すべてのPAXのアーキテクチャは2次元トラスであった。だいぶ後に、川合氏の「自然現象は近接相互作用で並列に進行している。これを計算機システムに直接写像するのが、自然の摂理にかなっている」という、伝道師のような説得力ある話を聞いたときに、妙に納得してしまった記憶がある。後述のように、原子炉燃焼シミュレーション用に開発されたアーキテクチャが、素粒子物理学の4次元格子量子色力学の計算にそのまま通用したように、幅広い分野に応用可能なアーキテクチャである。実用機として完成したことを含め、星野・川合の先駆性は外国でも高い評価を受けていた。

その当時、私には、並列計算のプログラムはいったいどうやって書くのか？ どのくらい大変なのか？ と見当もつかなかった。PAX-32Jの見学後、早速、教わりながら、ポアソン方程式のプログラムを書いた。思ったよりずっと簡単であった。それで、分担者になることを決断した。

特別推進研究のヒアリングには、研究代表者の星野氏、研究分担者の小柳義夫氏、私の3人で赴いた。ヒアリングの席上、星野氏と審査員との間で感情的とも思える激論になってしまい、それだけでヒアリングは終了した。

結果を待たずに、採択されないことは3人ともよく分かった。筑波への帰路、星野氏が、「物理の岩崎さんが代表を引き受けられないなら、これ以上はあきらめる。」と言い出した。私は、「少々待ってください。一週間下さい。」と戸惑いながら答えた。

悩んだ。自分としては物理をやりたい。でも、せっかく、星野・川合がここまで育ててきた財産を無にはしたくなかった。さらに、欧米で素粒子物理専用の並列計算機を開発しようという機運が高まっていた。迷いに迷い、引き受けた。45歳の時だった。

当時、国際会議などで欧米の研究者と議論していて感じていたことは、欧米の科学者にとって研究するのに必要なことはどんなに泥臭いことでも自分でやる、という精神風土である。ガリレイが望遠鏡を自作し、近代科学の扉を開き、その後も、その伝統が生きているということなのか。世界で最先端の新発見をするには、その背後に、地道に実験装置を開発する努力が重ねられている。たとえば、ノーベル物理学賞の約3分の1は、新しい装置を開発し、新発見をしたことに対して与えられている。一方、日本では、最初輸入文化であったことに起因しているのか、できればスマートに研究成果を出したい、泥臭いことは避けたいという風土があるような気がする。これは科学一般にみられる傾向であるが、計算科学者の場合も、出来ればプログラムを最初から書き込むのは避けたい、アセンブラ・コードなどで書くのはまっぴらだ。ましてや、コンピュータを自作するなど考えもしない、というのが一般的な風潮ではないかと思う。素粒子物理学の場合、実験では自前の加速器を設計・製作しないと研究自体が始まらない。素粒子物理学の理論研究者もそのような環境に囲まれているので、自然と影響も受けていたのだろう。物理の格子量子色力学の研究には、既存のスパコンでは計算能力が充分とはいえない。では、苦労してでも、コンピュータを自作する価値はあるのではないか、という考えが自分の背中を押してくれた気もする。コンピュータは計算科学のための加速器に相当すると考えた。

そのような思考過程を経て、「特別推進研究」を研究代表者として

申請することにした。「特別推進研究」の趣旨は比較的少人数で研究目的を明確にして計画・実行することだとわかり、研究題目は自分が解明したいと思っているテーマを出来るだけ具体的に「格子ゲージ理論によるハドロン の質量の計算及びクォーク・グルーオンの相構造の解明」とした。1986年の秋の1カ月を費やし申請書類を精魂傾けて書いた。今回は、ヒアリングも無事切り抜け、1987年度からの3カ年計画として無事採択された。

星野氏らと協力メーカ探しが始まった。富士通、日立、NECからは体よく断られた。しかし、アンリツ株式会社が協力してくれることになった。アンリツは最後まで責任を持って対応してくれた。その他、プロセッサ関係で、LSIロジック社なども協力してくれた。

このプロジェクトは、工学と理学の共同研究である。週一回のミーティングを中心に進められた。まだパソコンもインターネットもない時代であった。理学系棟と工学系棟とが、同じキャンパス上で300メートルしか離れていないことは有り難かったが、それでも不便であった。そんな時、池田克夫氏が自分の研究のパイロット実験として、2つの建物の間を結ぶLANを構築してくれた。ワークステーションを研究室に導入し、これには随分と助けられた。メールのやり取りで、仕事のはかどった。共同研究では、研究体制、「場」の構築が大切である。

ミーティングでは、まず、工学と理学との間で言葉、発想、手法が異なることに戸惑い、コミュニケーションをとることから始めなくてはならなかった。C MOS, ECLなどの言葉が飛び交い、その分野の事典をあわてて買いに走ったりした。

計算機のアーキテクチャは、PAXシリーズと同じ2次元トラスとした。4次元を2次元に投影することは何も問題ではなかった。計算機の愛称は、PAXシリーズの5代目であることと、格子量子色力学を意味するQCDを組み合わせて、QCDPAXとした。

ハードウェアは、星野氏の指揮の下、主に白川友紀氏がアンリツの技術者と打ち合わせしながら進めた。個々のノードの計算速度を最大限速めることにした。CPUは市販のマイクロプロセッサを、演算用FPUにはLSIロジック社のDSPを用い、FPUのコントローラFPUCをゲートアレイで製作し

た。ゲートアレイの設計は、小柳氏と白川氏が協力し、ベクトル計算の高速化を実現した。

プロジェクトを開始するにあたり、星野氏からは、物理学者は並列プログラムを書いて、計算機が完成するまで待ってればよい、とのことだったので、ミーティングでも、初めのうちは、報告を聞いていることが多かった。しかし、しばらくすると、状況はそう単純でないことが分かってきた。コンパイラーは修士一年生が受け持つことになったが、経験はゼロであった。進捗状況を聞いていると、とても2年間で完成するとは思えなかった。(アンリツも途中でそのことに気づき独自に開発を始めたが、我々のプロジェクトには間に合わなかった。)物理は、私と吉江友照さんと大学院生でやるつもりであったが、とんでもない事態になりそうな予感がして、人事担当副学長に状況を話したところ、即決で一人助手を採用できることになった。大型プロジェクトに大学として教員を補強する最初の例となった。筑波大学の柔軟な人事システムのおかげだった。早速、ヨーロッパで頑張っていた金谷和至さんに来てもらった。

2年ほど経過して、QCDPAXのハードウェアの半分は完成した。しかし、それからが地獄であった。最初は、簡単なプログラムでもエラーがでた。コンパイラーのバグであった。だいたいバグをとったが、とても通常のプログラムを通すことは無理であった。そこで、苦肉の策として、行列乗算のような基本演算は全てアセンブラコードで書き、関数呼び出しをすることにした。この作業は、岩崎、吉江、金谷でやった。それからがさらに厳しい山が控えていた。やっと本格的なプログラムを流せるかに思えたのだが、必ずどこかでこけた。これが、コンパイラーのバグなのか、ハードのバグなのかを見分けることから始めなければならなかった。コンパイラーの方は、関数呼び出しという方法を採用したこともあり、だんだんと、ハードのバグが姿を現した。

そのころ、新聞で九州の大学の教授が特別推進研究で、経費を設備購入にほとんど使ってしまい、研究を推進できなくなり、自殺したという記事が載っていた。他人事とは思えなかった。

それから、金谷、吉江が、睡眠時間を削り、約10カ月かけてハードのバ

グの原因を抜き出す作業をした。バグを必ず起こす1000行程度あるプログラムを縮めて、バグを起こす10行程度のプログラムまでもっていく。この作業に、約1ヵ月かかった。それで初めて、オシログラフなどを用いて、ハードのバグをえぐり出すのである。基本的には、3つの条件が重なると起こる現象が多かった。大体が、タイミングのずれとクロストークの組み合わせだった。コンピュータのデジタル計算がアナログ信号によっていること、しかもタイミングがかなり際どいことも、身にしみて実感した。この作業で10パターン位のバグを見つけ出した。これに約10ヵ月かかった。クロストークにはパッチをあて、タイミングのずれの解決のため、クロック周波数を少し下げた。その間アンリツは献身的な対応をしてくれた。

この後は安定して稼働し続けた。最終的に完成したQCDPAXは、当時のスパコンの最高速度の約三倍を達成し、経費は約10分の一であった。このQCDPAXを用いて、「宇宙の初期の相転移」に相当する「ゲルオンの相転移」の計算で、従来の結果をはるかに凌駕する世界最先端の見事な結果を導くことができた。コンピュータがまともに稼働するようになるか不安いっぱいの際のことを考えると夢のようだった。

一方で、将来の発展性を考えると、2次元トーラスの改良すべき点も明らかになった。自然を計算機に直接写像する点は何も問題がないが、自然界の格子点の数と計算機のノードの数の対応関係が厳しすぎる点である。たとえば、一方向のノードの数が8とすると、自然界の格子点も、8の倍数でないとプログラミングが極めて複雑になる点である。また、計算機を分割運用するのも困難なことも挙げられる。

物理の目標を掲げてコンピュータを作ることの重要性、理学と工学の連携の重要性、さらなる柔軟性をもったノード間ネットワークの必要性、ハードとソフト両方のデバッグ機能の整備の重要性などの教訓は、CP-PACSプロジェクトに受け継がれ、活かされた。

CP-PACS の PVP-SW 方式誕生のいきさつ

中澤 喜三郎

筑波大学電子・情報工学系(当時)

CP-PACS の思い出話について何か書くようにという御依頼ですので、ここには、筆者が関係した極くざつばらんなエピソードを披露させて頂く。

分散メモリ型超並列機である CP-PACS システムの大きな特徴として、

- node CPU に PVP-SW (pseudo vector processor based on slide windowed registers) 方式と称するアーキテクチャを導入し、擬似的に microprocessor であっても vector processor 並みの高性能を実現したこと
- node processor の相互結合ネットワークに cost performance の良い hyper crossbar network を使用したこと

を上げることができる。

ここでは、CP-PACS の独自の方式であり、本開発を通じて新たに考案された PVP-SW 方式が誕生した経緯を書き留めておくこととしたい。

CP-PACS プロジェクトのスタート時に、計算機工学分野のメンバーも参画することになったが、その際に、筆者らが意識していたのは、ある応用のための専用機ではなく、一般の科学技術計算にも充分能力を発揮するものを狙いたい、そしてその時点で充分確立した技術と云えるものではなかった超並列計算機の分野の技術の前進を図りたい、cost performance 的にも充分一般に導入可能なものを狙いたいと言うことであつた。

しかし限られた予算面からも、大学がもつ開発力からも、世の中一般製品化されているものと基本的なところで上位互換性を保てるものである必要があり、おのずから一般の computer 技術の発展の方向に合致したものでなければならぬことも強く意識していた。

一方、CP-PACS に先行した QCD-PAX での経験から、QCD 計算のグループでは CP-PACS では 相当高性能なマシンの開発が期待されて

いた。

そこで、確か 1991 年の夏、予算が付きそうな感触が得られた時点で、初歩的なシステムの構成と、QCD 計算の核になる部分の性能予測を募ったところ、最初に某会社から提案のあったものではその性能は期待されていたものとは大きな隔たりのある低いものであった。

幸か不幸か、この不満足な性能がわかった日の夜は、プロジェクトのメンバーによる夏休み直前の”暑気払いのコンパ”をやる日であったが、会が進むほどに、この性能不足のせいで皆の気分は沈み込み、“暗黒の暑気払い”として記憶に残る日となってしまった。

そこで、これではいけないと、筆者も夏休み中、性能が出ない理由を自宅で熱心に考えることになった。

提案されていたシステムでは node CPU は;

- 基本の命令 set architecture は HP 社の HP RISC architecture であるものの、
- node CPU は 2 命令同時 issue の super scalar RISC microprocessor で、clock 周波数もその時点での先端な 150 MHz、であった。
- pipeline 化された演算器も浮動小数点とは別々に固定小数点用も独立に用意されていて、
- Multiply and Add 命令も備わっていて、
- cache memory も通常のワークステーション向けのものに比し大幅に強化を予定されていた。

ということで、一応、先端的な汎用の microprocessor であった。

しかし、QCD 計算の核の部分での性能予測の結果は 理論的に期待される peak の 300MFLOPS はおろか、その 20~30 % 位の性能しか出ていないものであった。

そして、性能が出ない理由を検討したところ、最大の原因は、QCD 計算が必要とするような大規模なデータに対する計算の場合には、cache miss が頻発して、主記憶より cache memory への転送に伴う cache miss penalty が主要原因であることに思い至った。

つまり、命令の発行、演算器の pipeline 動作は 充分休み無く peak 性

能が出るように用意してあったとしても、演算器に供給してやるべきデータが間に合わなければ十分な性能が出る筈がないわけである。

実は、筆者はこれと似たような困難を嘗て経験したことがあった。日立、NEC、UNIVACなどで、汎用の main frame computer の付加機構として、内臓型の vector 処理機構(integrated array processor IAP と云う) , micro program を工夫して vector 処理を恰も vector processor のように処理しようという機構のことで、演算器 pipeline を整備して遊ばせないようにして cost の高い vector register を備えることなしに、経済的に vector 処理性能を稼ごうという機構を導入したのである。しかし、この試みは結局あまり成功しなかった。

一方、筆者らも手がけたことがある vector supercomputer の場合を考えてみると、vector supercomputer では cache memory は存在せず、それに代わるものとしてかなり大容量の vector register(VR) が用意されていて、主記憶と VR の間には load/store のための pipeline が複数用意され、演算器が必要とするデータは予め VR に用意しておいて、データは主記憶とは比較にならないほど高速の VR から遅滞無く途切れることなく演算器へ供給されるようにすることで、演算器の pipeline 能力をフルに利用できるようにしている。

つまり、通常の supercomputer では、cache memory より高速な register を十分な個数用意し、しかもこの register と主記憶の間に強力な pipeline を用意して、比較的低速な 主記憶 access latency に因る penalty を避けるために、register を指定しての先行 load(pre-fetch), と、置いてきぼり store(post-store) の命令機能が用意され、主記憶の遅さが表面に現れないようにしているのである。言い換えれば演算器へのデータ供給能力に大きな差があるところに scalar processor と vector processor では決定的な差があることに思い至った。

問題が発生する原因がほぼ明らかになったので、これを解決するには scalar processor といえども、主記憶との間に何らかの pre-fetch , post-store 手段を設けるべきで、そのためには主記憶との間でデータのやり取りをするための pre-fetch data buffer register をある程度の本数用

意しなければならないという考えに至った。

しかし、floating point register など program から直接指定できる通常の register の個数を単純に増やすことは、ベースとなる processor の 命令 format , ひいては 命令 set architecture に重大な変更を要することになり、直ぐには採用できないものであることも明らかであった。結局、pre-fetch data buffer を導入する必要性は明らかになったものの、導入方法に関しては成案なしに夏休みを終わってしまった。

夏休み明けに大学に戻って、筆者の研究室で中村宏講師、大学院修士 1 年の位守弘充君の 3 人で、休み中に筆者が考えた上記のような所までの説明をし、まだ解決策に達していない旨の話をし、議論をした。

そして、議論を終わって一応解散となり、筆者が自室に引き上げて帰った直後に、中村講師が筆者の自室に来て、

「先生、位守君が”register の本数を増やせば良いというのなら、SUN MICRO の SPARC のような register window なんて云う方法も考えられるのでは“と云っていますが、これでは駄目ですよね。」と告げたのです。

それを聞いて、筆者も「そうだな。」と最初は生返事をしていたが、中村講師が部屋を出た直後に、”待てよ。これは上手くいかも知れない！”と急に思いついた。

そこで、直ぐに floating point register の数を拡張して、register window 的な構造を導入し、

- 別途 pre-load, post-store 命令を設け、
- 主記憶を擬似的な pipeline memory としておく

ことで、base processor とは architecture 的に上位互換性を保って拡張したことになり、一応の解決策となる PVP-RW (pseudo vector processor based on register window)方式の仕様書を書き上げた。

ここでは、register window の構造(global area, overlap area, local area などの register の個数)は殆ど固定的なものであった。それでも、application program 中の Do-loop 部分を loop unrolling で modulo scheduling 手法で擬似 vector 化して扱えば、vector processor 並みの性能が発揮できる方式であった。

この時点ではまだ、この機能を活かして性能を稼ぐには、最悪の場合 application program はユーザの hand coding に頼るのも止むを得まいと思っていた。

しかし、ユーザが機械語のレベルでプログラムを hand coding しなければならないのでは、如何にもお粗末であるので、早速、compiler の権威である中田育男教授に PVP-RW の案を纏めたものを見せて、その妥当性と compiler support の可能性について相談をした。

中田教授は、この方式には大方において賛意を表してくれたが、PVP-RW では register window の window size も、offset の採り方もハードウェアで固定的であって、あまり自由度の無いものであったのだが、これらを program で任意に指定できるようにして一般性を持たせるようにすれば compiler support も可能になると思われるので、そうしたらどうかという更なる改良案の示唆をいただいた。

そうして纏め上げたのが、任意の場所へ window をスライドして持つことができることが出来る PVP-SW (pseudo vector processor based on slide windowed register) 方式である。

研究室の何人かの学生さんには、この PVP-SW 方式で modulo scheduling 方式がうまく行くかどうかを検討してもらい、一方、中田研究室、山下研究室の方々には compiler support についてよく検討していただいた。

結局、このようにして誕生してきた PVP-SW 方式は、丁度新たに一般向けの MPU を設計しようという日立製作所の開発フェーズと時期が合致し、その MPU の中に実現され、software のサポートも日立によって行われ、CP-PACS の高性能実現の強力な武器になったのである。

PVP-SW 方式は、後になって考えてみると、結局のところ、その後現れた高度 pipeline 制御を実現した MPU での hardware で有力な技術となった register renaming 技術を複雑な hardware の制御機構を設ける必要なしに software で assist した非常にスマートな方式であったわけで、その後現れた IBM の Power architecture や、Itanium architecture にも、類似の機構が導入されていることをみると、“暗黒の暑氣払い”を契機として

誕生したとはいえ, 一般的な computer architecture の発展の方向に沿った方式を編み出し得たものと考えている.

CP-PACS とコンパイラの思い出

中田 育男

筑波大学電子・情報工学系(当時)

CP-PACS プロジェクトが岩崎さんたちを中心として計画され、そのハードウェアを中澤さんが担当されるという話が進んでいた頃、私は中澤さんに「私にもソフトウェア担当として参加させてください」とお願いして、参加することになりました。私としては、日立から大学に移って以来ずっとしていなかったもの作りのプロジェクトに久しぶりに参加してみたいと思ったのです。

世界一の性能のものを作りたい、という計算物理系の利用者、情報系の開発者らの熱気に包まれた検討会は、コンパイラとそれに関連したソフトウェアのことしか知らない私にはよく分からないことも多かったです。充実感のある楽しいものでした。私の役割はソフトウェアシステムのとりまとめでしたが、私自身がそこで技術的に貢献したという記憶はあまりありません。ただ、CP-PACS のノードマシンの一つの特徴である疑似ベクトル機構の活用については多少とも貢献できたかなと思います。

CP-PACS は数千個のノードマシンからなる超並列計算機とし、ノードマシンとしては日立が製造している PA-RISC マシンをベースとすることで検討が進められていました。PA-RISC は他の一般のワークステーション用チップと同様に、キャッシュによってメモリとレジスタ間の転送時間の遅れを解消しようとするものでしたが、計算物理学の典型的な問題ではデータ量が多くてキャッシュがほとんど有効に働かず、目標とする性能が得られないことが問題でした。その解決策としてハードウェアグループから提案されたのが疑似ベクトル機構(スライド・ウィンドウ・レジスタ)でした。

その機構は次のようなものでした。

- (1) 浮動小数点レジスタを 32 個から 128 個に増やす
- (2) 通常の命令でアクセスできるのはその中の 32 個である(その 32 個だけが窓を通して見えると考える)。新設のプレ・ロード命令とポスト・ストア命令を使えば、すべての浮動小数点レジスタにアクセスできる。

(3) 窓をずらす(スライドさせる)命令がある。

この機構があれば、今見えている窓を通して計算している間に、後で使うデータを見えていないレジスタにプレ・ロードする命令を発行しておき、ロードが完了した頃に窓をそこにずらせば、ロードに要する時間を隠蔽することができます。

しかし、最初にこの機構が提案されたときは、レジスタ群を数個のブロックに分け、そのブロック単位で窓をずらすことができる、というものでした。それを聞いたとき私はコンパイラ屋として直感的に、ずらす幅は自由にすべきであると思い、それを提案して採用してもらいました。

この機構を生かすコンパイラのアルゴリズムは簡単ではないと思われました。中澤さんに、「コンパイラは大変でしょうね。」といわれたとき、私はただ「考えましょう。」とだけ言いましたが、実際にそれを考えてくれたのは、私と一緒に仕事をしていた山下さんでした。そのアルゴリズムの基本的な部分はソフトウェア・パイプラインングと呼ばれるもので、その当時主としてマイクロプロセッサの分野で研究されていましたが、我々はそれを知らずに独自に考え出してしまいました。山下さんは疑似ベクトル機構をもった PA-RISC 用のプログラムを完成させ、いろいろな例で確かめながら、アルゴリズムの改良も行いました。その結果、ループ 1 回ごとにスライドさせる窓の幅はその 1 回の計算に必要なデータの数(ロードの数)とするものになりました。そのプログラムはその後日立が開発した CP-PACS 用コンパイラに組み込まれました。

私は昔から、最適化コンパイラが実際にどの程度最適化をしているかを簡単に調べる一つの方法として、行列の掛算のプログラムの最内側ループの目的コードを調べています。そのプログラムは

```
do i = ...  
  do j = ...  
    do k = ...  
      sum = sum + A(i,k)*B(k,j)
```

という形であり、最内側ループは k のループです。通常の RISC マシンでは、最適化を相当頑張れば最内側ループの目的コードの命令数は 8 になりま

す。その内訳は、sum をレジスタに割付けて代入の必要がないようにし、データ(配列要素)のロード命令が2個、加算と乗算の命令が2個、次のデータがロードできるようにインデックスレジスタの値を増やす命令が2個(最適化を頑張らなければこの命令数が増える)、ループ終了判定とブランチ命令が2個、という構成です。

CP-PACS の Fortran コンパイラでは、この命令数が 3.5、それを実行したときのサイクル数が 2.5 になっています。これは私が知っている範囲では最も効率が良いものです。その内訳は次のようになっています。まず k の 4 回分のループ展開をして

```
do k = ...
    sum = sum + A(i,k)*B(k,j)
    sum = sum + A(i,k+1)*B(k+1,j)
    sum = sum + A(i,k+2)*B(k+2,j)
    sum = sum + A(i,k+3)*B(k+3,j)
```

とすることによって、PA-RISC の FMPYADD (floating multiply and add) 命令が使えるようにしています(これは乗算と加算を同時に実行する命令であるが、その 2 つの演算は別々のデータに対して行うものであるからもとのループの形のままで使えない)。このループに対するロード命令は8個、演算命令は FMPYADD が4個、窓を8だけずらす命令が1個、ループ終了判定とブランチ命令が1個、で合計 14 命令です(PA-RISC では次のデータがロードできるようにインデックスレジスタの値を増やす命令はロード命令の中に組み込むことができるので、それを利用すれば命令数としては必要ない)。これを展開数で割った $14/4=3.5$ がもとのループの 1 回あたりの命令数です。ここでのロード命令は、ループの繰返し後のほうで必要とするデータをあらかじめロードしておくプレ・ロード命令です。演算命令は以前のプレ・ロード命令によって既にレジスタにロードされているデータに対する演算命令で、したがって、プレ・ロード命令と演算命令は並列に実行され、演算命令の時間はプレ・ロード命令の時間にかくれてしまいます。したがって実行のサイクルはプレ・ロード命令の8、窓をずらす命令の1、ループ終了判定とブランチ命令の1の合計の 10 で、もとのループ 1 回あたり

のサイクル数は $10/4=2.5$ となります.

このプロジェクトで学んだり考えたりしたことや、勉強不足を感じて勉強したことなどは、拙著「コンパイラの構成と最適化」に盛り込むことができましたし、その後私が研究代表者として進めた「COINS コンパイラ・インフラストラクチャ」プロジェクトにも役に立ちました. CP-PACSプロジェクトに参加させてもらったことで、久しぶりの充実感も得られましたし、そこで得たものは、その後の私の研究にも大きな支えになったと感謝しています.

CP-PACS に教わったこと

中村 宏

筑波大学電子・情報工学系(当時)

東京大学大学院情報理工学系研究科(現在)

CP-PACS は計算機の名前でもありプロジェクトの名前でもあったが、私にとっては、いろいろなことを教えてくれた存在でもあった。私はノードプロセッサのアーキテクチャを担当し、擬似ベクトルプロセッサというアーキテクチャを提案し、皆さんに認めてもらって VLSI として実装することができた。幸い、完成した CP-PACS が世界最高性能を達成したことで、プロジェクトの皆さんの期待にこたえることができたわけだが、私は2つの大きなことを教わったと思う。

第一は、アーキテクチャの考え方である。実は擬似ベクトルプロセッサは私が生まれて初めて真剣に考えたアーキテクチャである。と言うと驚く方もおられるだろうが、これは本当の話である。それでも結果的に良いアーキテクチャを提案することができた理由の1つは、制約条件が明確であったからである。制約条件のひとつは吉江先生のグループにより開発された Fortran コード、サブルーチン MULT をとにかく高速に実行することであり、もうひとつの制約条件は従来のアーキテクチャと互換性を維持することである。私は CP-PACS を通して、アーキテクトの仕事は、制約条件下でできるだけ良い解、具体的にはより低コストで高性能を実現できる落としどころを見つけることである、というのを体で感じることができた。擬似ベクトルプロセッサでは、例えばリストベクトル処理用の高速化機構も考えたが、なくても制約条件を満たすし、あるとコストが増える、という理由で見送ったし、インターリーブメモリのバンク数をどこまで減らせるか、ということもかなり検討した。多くの場合、アーキテクトは新しい機構を入れその新機軸の有効性を主張したが、それもアーキテクトとしての重要な役割であろう。しかしもっと大事なものは、いかに不要なものを削り落とすか、その決断と裏づけとなる根拠を整理することである、ということ学んだ。駆け出しの研究者として、

最初に考えたアーキテクチャでそこまで教えてもらえるのは非常にラッキーなことであり、その意味でもこの計算機、そしてプロジェクトに感謝している。

第二は、参加メンバーが熱意を持って目的を共有することが、プロジェクトの成功には重要だ、ということである。擬似ベクトルプロセッサというアーキテクチャも、参加メンバーの熱意が実を結んだものであった。もう19年ほど前の話しになるが、私の記憶を整理したい。CP-PACS の初期検討では従来と同じキャッシュメモリを利用するアーキテクチャが考えられていたが、容量や連想度を現実的には無理と思われる程増やしても MULT ルーチンの性能は理論性能の 20%弱しか出なく、ずっと簡単な Livermore Fortran Kernel でも同じようなものであった。当時修士1年の学生だった位守君に詳細な評価をしてもらっていたが、彼曰く「問題はキャッシュです。大規模データではキャッシュヒット率が低く、キャッシュの構成をどう変更しても駄目です」。大学院の学生としてはこの回答で与えられた課題に対しては十分なのだが、その後続けてこう言った。「メモリから直接レジスタに先行ロードするしかありません。しかし、主記憶のレーテンシが大きいのでレジスタ数を増やす必要があります。それはできないのですか？」彼の立場でのプロジェクト成功への情熱があつてこそその意見である。互換性という制約条件は外せないからそれは無理だ、私は思っていた。しかし、「やっぱり駄目だね」というのは簡単だったが、私もすぐにはあきらめられず、問題点の整理と解決の糸口を見つけるべく話し合ったのだが、彼の分析と提案が良く整理されていたので、彼の主張と、SPARC アーキテクチャにおいて当時全く別の目的のために採用されていたレジスタウィンドウ構成が、互換性を維持しつつ多くのレジスタを利用しているという点で私の中でリンクされ、レジスタウィンドウ構成をレーテンシ隠蔽という別の目的のために使えるかもしれない、と感じた。早速その足で計算機側のリーダーである中澤先生にアイデアを持っていった。中澤先生は私とは違って多くのアーキテクチャを基本設計から最終実装まで手がけてこられた方なので、その基本アイデアの筋の良し悪しはすぐに見当がつくと思っていた。駄目出しされるかなと思いつつ、しかしそのアイデアに酔いながら、興奮しながら

ら話した記憶がある。中澤先生は慎重だったが、良いかもしれない、という表情もされていた。私は良い気持ちでその日は帰宅したように思う。そうしたら、次の日に先生に呼び出され、細かい手書きのメモを見せられた。驚いたことに、それは、提案するレジスタウィンドウ構成において Livermore Fortran Kernel を実行するときのレジスタ割り当てと命令スケジューリングであった。先生と私の間の、このプロジェクトに賭ける熱意の違いを感じた一瞬であった。そのメモは良好な結果を示しており、先生は物理学の研究グループに早速そのメモを回された。数日後のプロジェクトの全体会議で私は再び驚いた。物理の金谷先生が、表計算ソフトを駆使して綺麗な図を書いていた。見たらレジスタウィンドウ構成における MULT ルーチンのレジスタ割り当てと命令スケジューリングである。聞けば、メモを見ながら自分で適宜解釈・想像しながらやってみた、とのこと。それがまた完璧な最適化がなされていたのである。1サイクルでも実行時間を短くしたいという気持ちのこもった最適化であった。参加者の皆が、これはいける、と思ってその日の全体会議を終えることができたように思う。こういった熱意を持った人たちのおかげで、このアイデアは単なる思いつきから立派な提案になったのである。後にコンパイラの中田先生により改良され、スライドウィンドウ方式による擬似ベクトルプロセッサとなり、さらに製作・実装側の日立製作所の皆様の熱意も加わって、VLSI として実現されるに至ったわけである。ちなみに、先の位守君は、修士課程修了後に日立製作所へ就職し、熱意をそのままに擬似ベクトルプロセッサの実装までやり遂げた。

このように、極めて幸いなことに私は若くして貴重な体験を得ることができた。計算機としての CP-PACS はすでに現役を引退したわけだが、私の中では、CP-PACS は思い出としての記憶ではなく、教えてもらったことが脈々と今でも生き続けるものとして存在している。プロジェクトで一緒にさせていただいた皆様、そして CP-PACS に受けた恩を返すべく、これからの計算科学の発展に少しでも寄与しなければ、と実は今、多少焦っている。

CP-PACS プロジェクトに携わって

山下 義行

筑波大学電子・情報工学系(当時)

佐賀大学理工学部知能情報システム学科(現在)

プロジェクトから離れてもうだいぶ経ちますが、未だに様々な事柄を鮮明に思い出します。先日、当時指導していた大学院生と久しぶりに食事をする機会があり、当時のいろんな思い出話に盛り上がりました。話していて分かったことは、CP-PACSの思い出というと二人共、同じ話に収束してしまい、それというのは、スライド・レジスタを導入するとループ最適化法がこんなにも劇的に変化するものなのかと驚いたことでした。さらに私について言えば、未だにそれを引きずっています。以下、とても限定された話題ですが、記録に残していただければ幸いです。

まず、理論的に言えば、命令スケジューリングもレジスタ割り付けも一般にNP問題であって複雑なアルゴリズムを用いてなお準最適解しか求めることができない厄介な問題です。しかし、スライド・レジスタを用いるならば、いずれもP問題になるため、ほぼストレートに最適解に到達できます。特にレジスタ割り付けについては、当初、NPと思い込んでいたためにPと知って驚愕したことを憶えています。この結果はさっそく1本の論文にまとめたのですが、論文の書き方が悪かったせいか(当時、諸事情で落ち着いて論文を書く余裕がありませんでした。尤も今も落ち着いて論文を暇はありませんが)、査読者から「そんなことはあり得ない」と一喝されて不採録になってしまいました。しかしそれすら「査読者がそう断定しても不思議ではない」と妙に納得したものでした。スライド・レジスタの理屈はとても簡単で、スライド・レジスタを用いればループ内でのレジスタの自己干渉を自然に解消でき、組み合わせ問題の最適化に付きまとう様々な煩わしさが跡形も無く消えてしまいます。結果、プログラムのループ部を基本ブロック部であるかのように取り扱うことができるのですが、その辺りに不思議な感覚が残ります。

実用上も、スライド・レジスタを用いることで命令スケジューリングの最適

化アルゴリズムが劇的に単純化されます。これには助かりました。当時のプロジェクト内での私の役割は、最適化アルゴリズムをコンパイラに試験的に実装してみせることでした。本来 NP であるような問題に取り組むとなると、検討段階では単純なアルゴリズムでおおむね問題ないと思っても、実装段階では予想外の経験値を導入せざるを得なくなったりして次第に複雑怪奇なものに変容していくものです。しかしスライド・レジスタを使ってプログラミングしてみると「これでいいのだろうか」と思うくらいに単純な実装でも十分であることが分かり、調子抜けしたことを印象深く憶えています。短期間で最適化プログラムを試作できたことはスライド・レジスタあったればこそだったでしょう。

さて、未だにコード最適化の研究をしています。ちょうどプロジェクトを離れる頃に、Intel の Itanium プロセッサが rotating register という機構を持つことを知りました。これは CP-PACS のスライド・レジスタとほとんど同じものです。今ではこのプロセッサの上でいろんなプログラムを組んではぼつぼつと論文を書いています。前出の大学院生とは不採録の論文を書き直そうと約束していますが、果たしていつのことになるのか。残念なことに Itanium プロセッサは商業的には失敗し、市場から消えそうです。消えてしまう前にはすっきりとスライド・レジスタ関連の研究に全て決着をつけたいと考えています。

人生の転機になった CP-PACS プロジェクト

朴 泰祐

筑波大学電子・情報工学系／計算物理学研究センター（当時）
筑波大学システム情報工学研究科／計算科学研究センター（現在）

私が CP-PACS プロジェクトに参加させて頂いたのは、まさに 1992 年 4 月のプロジェクトの正式発足時点からであった。その直前の同年 2 月 16 日、前所属だった慶應義塾大学物理学科助手を退職し、筑波大学電子・情報工学系講師として筑波大に赴任してきた直後だった。

慶大では「計算機工学出身者（実際には慶大には当時 CS 学科がなかったため卒業は電気工学科だったが）でありながら、物理学科の助手になる」という、非常に特殊な存在だった。CP-PACS の一つ前のプロジェクトである QCDPAX プロジェクトに関係していた川合敏雄先生（当時慶應義塾大学物理学科教授）が、QCDPAX の小型機を同大物理学科に導入し、その管理と計算物理学研究への応用を行うにあたり助手を探されており、同大で並列計算機工学の研究をしていた当時博士課程 2 年だった私が選ばれたという経緯だった。慶大には「博士課程在籍中に学生の身分のまま助手になる」という例はいくつかあったが、私のように学科（専攻）を跨いで適用された事例はほとんどなかったと思う。つまり、博士号を取得するまでは、物理学科では助手だが電気工学科では学生という不思議な立場だった。当時、慣れない物理数学の演習助手等を苦勞しながらやっていたが、今にして思えば、この「計算機出身で物理を手伝う」という立場に早くからいたことが今日までの自分のキャリアを作ってきたと思えてならない。

ともかく、4 年の任期の最終年度になり移転先を探していた私のところに、筑波大で講師を探しているという話が、当時筑波大構造工学系教授だった星野力先生から舞い込み、結果として筑波大にお世話になることになった。私としても、それまで研究室で実験的な並列処理システムをいくつか開発していたが、本格的な、しかも「超」がつく並列計算機プロジェクトへの参加は初めての経験だった。所属は電子・情報工学系で、CP-PACS プロ

プロジェクトの発足と同時に立ち上がった計算物理学研究センターで勤務することになり、中澤喜三郎先生(当時電子・情報工学系教授)の下で、中村宏さん(当時講師、現東京大学准教授)と3人で研究室を運営しつつ、CP-PACS プロジェクトに参加することになった。

慶大の学生時代は並列計算機工学、特に並列処理ネットワークと並列プログラミングの研究を行っており、中村さんがプロセッサアーキテクチャの専門家だったことから、「演算と通信」という高性能並列システムハードウェアの両輪を二人でそれぞれ担当し、システムの総まとめを中澤先生が指揮されるという形であった。当時、CP-PACS の並列通信ネットワークである3次元 Hyper-Crossbar (3D-HXB)網が既に日立から提案されており、私はこのネットワークの面白さに嵌ってしまった。CP-PACS で採用された3D-HXB網は、クロスバ網の柔軟性とメッシュ網の低コスト性を持ち、さらに Hyper-Cube より degree(ノード間接続の次元数に相当)が小さく、非常にバランスの取れたネットワークであった。

最大の問題点は実装の難しさで、2次元までは比較的コンパクトだが3次元になるとケーブルによる「空中配線」を行わなければならない。今でもCP-PACS の保存筐体を見るとその実装の大変さがわかる。2048 ノードというシステムサイズは当時としては大変なチャレンジであったが、何とか性能を落とさず実装できるぎりぎりのサイズだったと思う。結果的に、日立の技術力のおかげで、リンク当たり 300MB/s という、当時として極めて高いバンド幅が達成された。これは wave-pipelining 技術により、電線中の波を定在波にせず、次々に送り込む事でネットワークの動作周波数を高く維持できたことが成功の鍵であった。今日、講義や講演等で高性能並列処理における CPU 性能:メモリ性能:ネットワーク性能の比率について必ず語ることになっているが、300MFLOPS : 1.2GB/s : 300MB/s=1:4:1 という比率は今日ではまず考えられない理想的なバランスである。演算性能とメモリ性能のこのバランスは、まさに疑似ベクトル処理機構のたまものであり、“4B/FLOPS”という「ベクトル処理の矜持」を守ったシステムとなった。これがあるからこそ、ユーザは演算性能を最大限に活かしたプログラムを開発することが可能なのである。

3D-HXBは本当に研究甲斐のあるネットワークで、手始めに3D-HXBのクロックレベルシミュレータを開発し、いろいろな並列通信アルゴリズムの実装や、他の並列ネットワークの組み込み等の研究を続けた。当時始まったばかりのNAS Parallel Benchmarksのシミュレーション評価も大変実りのある研究だったし、3D-HXBが実はFat-Tree網の一部と等価である事、ADI法のために日本で研究されていたADENA網がほぼ完全に組み込める事、3D-HXB向けの効率的な集合通信アルゴリズム等、どれをとっても面白いテーマが一杯で、CP-PACSが完成する前から、これらのテーマの下で学生と共に研究を楽しんだ。最終的に、それまで固定ルーティング方式にしか対応していなかった3D-HXB上で、適応ルーティングを行う理論とアルゴリズムの研究まで進んだが、残念ながらこれはまだ実用にはなっていない。

同時に、CP-PACSの開発では、プロジェクト関係者全員参加で、毎月1回丸一日を費やして行われる筑波大・日立の合同ワークショップにおいて、特にネットワークの開発と性能評価に力を注いだ。プロジェクト3年目頃によく出て来たネットワークの詳細設計を見た時に、本当にマシンがもうすぐできるのだということを実感した。ハードウェアの基本設計、通信バッファの取り方、パケットヘッダの詳細、ハードウェアによる同期機構やCOMBUF通信とそのチェーニング機構(パケット列を順次自動的に転送する仕組み)等、高性能化を目指した非常に緻密な設計仕様を隅々までチェックし、バグを見つけ、評価予測を立てるという作業に非常な充実感を覚えたことを記憶している。

プロジェクトに関係した全ての人々の努力により、CP-PACSは1996年9月に2048ノードのフルシステムが完成し、同年11月のSC'96におけるTOP500リストで世界第一位にランクされることになった。この発表の場(TOP500 BoF)での話だが、実はCP-PACSプロジェクトの参加者はほとんどSC会場におらず、当時は上位ランクの表彰(certification 授与)やスピーチもなかったが、プロジェクト関係者として発言を求められたため、システムの概要について簡単に紹介した。当時、TOP500リストはまだ現在ほどは騒がれておらず、筑波大でもシステム開発が最優先で、システムに関する

論文が海外でほとんど出ていなかったため、多くの人が CP-PACS のことをよく知らないという状況であった。その後、いろいろな論文を発表し、世界での認識が深まっていったが、現在の TOP500 リストの騒がれようからはおよそ考えられないような話である。

CP-PACS の 3D-HXB の最大の危機は、システムがフル稼働してしばらくしてから訪れた。極めて緻密に作られ、クロック同期が取られている 3D-HXB 上で、どうしてもある確率で転送エラーが発生し、プログラムがダウンするという現象が起きていた。特に、CP-PACS での初期プロダクトランは QCDC 計算に集中していたため、256~1024、場合によってはシステム全体を使う 2048 ノードの大規模並列ジョブが絶えず走っており、1つのエラーによって計算全体が落ちてしまうという超並列処理の難しさが如実に現れていた。エラーが起きるたびに、wave-pipelining の調整のためケーブル長を修正し、クロックの位相修正をしていたが、どうしても局所的な修正では限界があることがわかってきた。そこで、日立の提案により、クロックの極性を逆転させ、パケット到着のタイミングがクリティカルにならないような改造を行うことが決まった。詳細は省略するが、これはシステムのハードウェア全体に対する「大手術」で、このためにシステムが長期間停止されたが、結果としてこの作戦は成功し、その後の稼働では見違えるように安定するようになった。

個人的には CP-PACS プロジェクトがきっかけで筑波大に来たわけだが、つくばという土地に移った事は自分を取り巻く研究生活に非常に大きな変化をもたらした。当時、同じように高性能並列処理の研究を進めていた産総研の研究チームと共同研究を行う事になり、毎月の定例ミーティングで産総研・お茶の水大学・筑波大学が協力して様々な HPC 研究を展開した。その中心は当時産総研の方式研・アーキテクチャグループにいた関口智嗣さん(現情報技術研究部門長)で、彼の周りの様々な HPC 研究者と深くかかわることになり、研究環境という意味での人脈が大きく広がった。その集まりの中で、当時同じく産総研アーキテクチャグループだった佐藤三久さん(現筑波大学計算科学研究センター長)と知り合った。最初から妙に息が合い、並列処理性能評価とツール作成というテーマでいろいろな議

論を行ったと思うが、その当時は今日のような繋がりにまでなるとは予想していなかった。

つくばという土地でのもう一つの大きな出来事は、CP-PACS とほとんど同時に始まった、通産省の RWCP プロジェクトの開始であった。RWCP の集中研究室がつくば三井ビルに設置され、ビル内の複数のフロアを占有するほど多数の研究者が集結していた。慶大の学生時代の研究室の先輩に当たる石川裕さん(現東京大学情報基盤センター長)が、産総研から出向して RWCP の立ち上げを行っており、いろいろな議論を行う機会ができた。そして、RWCP の後期に入ると佐藤さんが RWCP に出向し、石川さんのリーダーシップの下で超並列高性能 PC クラスタのための OS として SCore の開発が行われた(SCore はその後、CP-PACS の後継プロジェクトである PACS-CS で採用された)。その頃、私は佐藤さんの研究室から研究再委託を受け、彼との本格的な共同研究が始まった。そして、最終的に RWCP を終えた佐藤さんが筑波大計算物理学研究センター教授として就任することが決まり、研究室の共同運営がスタートした。また学内だけでなく、関口さんや佐藤さん、さらに大勢の HPC 研究者との繋がりの中で、情報処理学会 HPC 研究会の運営に携わるようになり、2006～2009 年度は HPC 研究会主査を任されることにもなった。国内外の数多くの HPC 研究者との繋がりが広がって行ったが、やはり国外では CP-PACS プロジェクトの一人として認識されたことが、様々な人達との繋がりのきっかけになったと思う。

このように、自分の周りに次々に魅力的な研究仲間が急速に増え、人との繋がりが学内から日本国内、さらに世界へと広がって行き、現在の自分の研究生活の基盤が作られて行ったが、その元を辿れば、星野先生と共に PACS 及び PAX シリーズの研究を開始した川合先生の助手になったこと、そしてそれを通じて CP-PACS プロジェクトに参加できたことが、世界一のマシンを作るという作業に参加できただけでなく、まさに自分の人生の転機になったと信じている。その意味で、川合先生・星野先生、そして CP-PACS プロジェクトとそれに関係した全ての方々へ感謝したい。

2005 年 9 月、CP-PACS はその稼働を終了することになり、多くの関係者

が集まる中、システムのシャットダウンと記念撮影、さらに懇親会が行われた。懇親会の場で星野先生が声をかけて下さり、「朴さんを筑波大に呼んで本当に良かった」という暖かいお言葉を頂いた。このエピソードをもって、本稿の結びとしたい。

私に取っての CP-PACS プロジェクト

宇川 彰

筑波大学物理学系(当時)

筑波大学本部(現在)

1. はじめに

私の専門は素粒子物理学の理論的研究である。初めて研究論文を書いたのが 1974 年, 格子 QCD の研究を始めたのが 1979 年, 数値シミュレーションを使った最初の論文が 1981 年であるから, CP-PACS プロジェクトに加わった 1991 年には既に 20 年近い研究生活を送っていた。しかしながら, CP-PACS プロジェクト 5 年間の経験を境目として, 私の素粒子物理学に対するアプローチ, さらにには研究そのものに対する考え方や取り組みは, それ以前とは全く違ってしまった。この一文では, その事情を書いてみたいと思う。

2. CP-PACS プロジェクトへの参加

筑波大学には, 既に 1980 年頃から, 星野力先生を中心とした科学技術向けの並列計算機開発の歴史がある。岩崎洋一氏が加わり, 科研費特別推進課題の研究代表者となった QCDPAX プロジェクトは, 岩崎氏と, 吉江友照氏や金谷和至氏が物理側の中心となって 1987 年から 1989 年の 3 年計画で推進された。私が筑波大学物理学系に着任したのは, それに先立つ 1985 年であるが, この時期, 私自身は高エネルギー物理学研究所(現在の高エネルギー加速器研究機構)の福来正孝氏(現東京大学宇宙線研究所教授)や大川正典氏(現広島大学物理学研究科教授)と研究グループを組み, 高エネ研設置のスーパーコンピュータ(日立 S810 を 1986 年に設置, 1991 年に S820 に更新)を多用した格子 QCD の研究を行っていて, QCDPAX プロジェクトには参加していなかった。

QCDPAX プロジェクトの後継プロジェクトとして, 岩崎氏を代表とする CP-PACS プロジェクトが, 文部省の新プログラム(学術の新しい展開のた

めのプログラム)の課題候補として検討され内定したのは1991年初めであったと記憶する。岩崎氏からプロジェクトへの参加を勧められた当初、私はあまり乗り気ではなかった。当時の私は、計算機開発に十分な理解を持たず、またQCDPAX開発の様子からもその大変さに尻込みの感もあった。しかしながら、高エネ研の菅原寛孝先生(当時)から、文部省の重要施策である新プロに採択されたからには、我が国の格子QCDの関係者が全員で取り組むのが当然ではないかと、ことの重要性和考えの浅さを諭され、参加させて頂くこととした。

中澤喜三郎先生や中村宏氏らの計算機科学側のメンバーと物理側のメンバーが一堂に会した初会合は1991年5月9日であったと思う。この時点では、開発システムの名称は決まっていなかった。CP-PACSの名称が決まったのは第二回会合の行われた5月16日と記憶する。

3. CP-PACSの開発・制作

CP-PACSプロジェクトは正式には1992年度から1996年度までの5カ年計画であったが、1991年春から素粒子物理における格子QCD計算の具体的目標の検討や、計算機工学側の計算機システムに関する技術的検討が行われていた。素粒子物理側では、岩崎氏、吉江氏、金谷氏、宇川に、青木慎也氏も加わり、月一回の物理側と計算機工学側の合同WGを軸として検討が進められた。この形態は、一方では、計算物理学研究センターの同時正式発足を受けて、同センターの研究員会議へと発展し、現在の計算科学研究センターにおいても、教員・研究員の情報共有・意見交換の場として受け継がれている。また、その一方で、1992年にプロジェクトが正式に発足し、システムの開発・制作者が(株)日立製作所と決まったことにより、大学側と日立側の合同WGへと形を変え、その後プロジェクトが終了する1997年まで略月一回のペースで続けられた。

計算機の開発・制作に参加するのは私にとって初めての体験だった。大学・日立合同WGは、毎回5時間も6時間も続くタフなものだったが、最初の内は計算機工学や日立製作所のみなさんの話す内容が判らず、精神的に苦痛であったし、計算機に関する知識の不足を痛感させられた。中

澤先生の講義録「計算機アーキテクチャと構成方式」(後に朝倉書店から1995年に出版され大川出版賞を受賞)を読んで、レジスタやキャッシュ等の計算機の基本中の基本を勉強したこと、また、計算機のみなさんと議論したり、彼らが端末からシステムを操るのを見て、「この人たちには、きっと計算機の中は手に取るように視えているんだろうなあ」としみじみ思ったことなど、懐かしい思い出である。そのような開発・制作の中で私の記憶に残る事柄を挙げてみよう。

(1) 疑似ベクトル機構 PVP-SW

格子QCDの計算の中心部分は、その後サブルーチンMULTという名前で関係者の間で有名になったが、1演算あたり約2.5バイトのデータをメモリからレジスタへロード・ストアする必要がある。当初検討されたPA-RISCアーキテクチャでは、問題サイズがキャッシュ溢れを生ずる大きさとなった時点で実効性能が極端に低下してしまう。今では素粒子関係者の間では常識となっているが、何も知らない当時はこの結果に素粒子一同顔色を失った。当初のシステムのピーク性能目標は300Gflopsだったが、実効性能12.3%では、世界一はおろか、目標としていた計算も夢のまた夢となりかねない。1991年8月10日が、「暗黒のCP-PACSミーティング」として記憶されている由縁である。

この問題は、その後時間を置かずして疑似ベクトル方式により解決の見通しが立った。この間の事情については中澤先生始め、多くの方々が書かれると思うので、これ以上触れることはせず、PVP-SWの経験が私にもたらしたものを書いておこう。

PVP-SWの威力は、1991年の9月から10月にかけて、自分でメモリーとレジスタ間のデータのロード・ストアやレジスタ上での演算を、各操作のレイテンシを数えながら配置し、計算に必要なマシンサイクル数を勘定して納得することができた。一挙に50%を超す実効性能が可能であることが示唆され、興奮を伴う体験だった。私はそれまでにスーパーコンピュータを5年以上にわたって使って来た。しかし、計算機の動作をこのようなレベルで考えたことはなく、それが研究を進める上で如何に重要であるかについて

でも理解していなかった。私は、CP-PACS プロジェクトの期間を通じて、幾度か同様の経験をし、この事を通じて、計算科学者にとって計算機を深く知ること、また計算科学と計算機科学の深い協力が如何に重要であるかを深く認識するようになったと言って良い。

(2) できたてほやほやの CP-PACS

開発・制作された CP-PACSを使うという点からは多くの思い出がある。1996 年 1 月には、開発段階のハードに OS 及び PVP-SW 対応のコンパイラを装備した 8 ノード構成機が Pilot-2.5 システムという名称で計算物理学研究センターの計算機室に持ち込まれ、ユーザ側のプログラム開発に供された。

並列プログラミングは私には初めての経験であったので、統計物理の標準問題であるイジングスピン系を使って練習を積んだ。CP-PACS はノード間通信に、新規開発のリモート DMA 通信機構を備えており、combuf 通信 library と称する一群のライブラリコマンドを使ったプログラミングが必要だった。計算機工学側の朴泰祐氏が作成したプログラミングマニュアルを片手に毎日コーディングに取り組んだことを思い出す。

格子 QCD プログラム本体については、quenched QCD 用の heat bath プログラムと、クォークの動的効果を取り入れた hybrid Monte Carlo プログラム、さらにハドロン伝播関数等の物理量の計算プログラムの三種類の開発が必要だった。私は、ベクトル計算機用の hybrid Monte Carlo プログラムを作成した経験があったので、こちらに取組み、まずは純グルオン系に対するプログラムを書いた。1996 年 3 月には一応動くプログラムが完成していたと記憶する。クォークの効果を取り入れたバージョンは、同年 8 月に金谷和至氏が中心となって書いた。

CP-PACS は、まず 1024 ノード構成機が 1996 年 3 月 25 日に設置され、4 月 8 日から稼働が開始された。Pilot-2.5 の 3 カ月の先行設置のおかげで、格子 QCD 計算は開始の準備が整っており、我々物理関係者は皆が「それ！」という勢いでテストを始めた。単体ノードでのテストから初めて、4 ノード、64 ノード、256 ノード等々と並列プログラムのテストを進めていく。純

グルオン理論用の hybrid Monte Carlo プログラムは本格計算用としては heat bath プログラムに劣るが、このテストの目的には随分と役に立った。

この段階での一番の思い出はパフォーマンスモニタ mon である。このモニタソフトは、調査したいプログラム部分の前後に開始・終了の statement を入れておくだけで、その部分の実行に要したマシンサイクル数、演算数、データのロード数、ストア数、キャッシュミス数等の基本情報がノード毎に採取できた。hybrid Monte Carlo プログラムの各 do loop や combuf 通信部分を mon で調査することにより、できたてほやほやの CP-PACS がどのように動作しているのか、事前に行っていた詳細な検討どおりの動作をしているのか、手に取るように視ることができたことは私に鮮烈な印象を残している。

4 月のうららかな日が続く中、週末の静かなオフィスでキーボードを叩いて、同じプログラムを何度も計算させ、各 CPU の動作確認や combuf 通信ライブラリの性能評価をしたのを思い出す。問題が起こる度に、計算物理学研究センターに詰めていた SE の滝田忠郎氏や日立電子サービスの矢嶋則幸氏に電話をかけた。当時の私は、夕食に一度帰宅した後、ひと眠りして夜 10 時頃に大学に戻り午前 2 時～3 時まで仕事、朝は 9 時前には大学に出て仕事という生活を、平日・週末の別なく続けていた。時間の別なく電話して随分とご迷惑をかけたと今では冷や汗の思いである。

4. 計算物理学研究センター

CP-PACS プロジェクトは、異なる分野や、大学とメカなど異質の組織に所属する大勢の研究者が協力する必要があるプロジェクトの進め方を私に教えてくれたが、同時に新たな研究組織の立ち上げやそのための文部省(当時)への概算要求が如何なるものかも知る機会を与えてくれた。

新プログラムは、採択された場合に、その推進母体として全国共同利用の研究センターを設置する可能性があった。採択が内定した 1991 年の春には、新センター設置を目指して、まずは学内の研究担当副学長や研究協力部、財務部と全体計画や教員振替などの調整、それを踏まえて文部省の研究機関課(当時)との折衝が岩崎さんを中心に行われ、私もそのよ

うな席に幾度か同席する機会があった。

今でもそうだが、文部省との折衝は担当課の課長補佐への説明が中心になる。当時の概算要求は、人員や予算、また建物等について記入する一定の書式があり、その上に要求理由を説明する文書や添付書類を準備する必要があった。「ポンチ絵」と呼ばれる、判りやすい図等の作成も必要だった。これまた週末の物理学系の建物で、当時の最新鋭の PC98 パソコンに一太郎を使って、岩崎氏と書類準備をしたのを思い出す。

その上での研究機関課との折衝になるが、これまた大変だった。小一時間ほどの折衝の間、にこりともしないで、「本当にセンターを設置する必要があるのか、人員要求は過大ではないのか、フロントエンドは何故必要なのか」と次々と厳しい質問を浴びせかけてくる文部省の担当の方々を相手に岩崎氏、中澤先生が丁寧に反論を重ねていたことを思い出す。結果として、センターの 1992 年 4 月 10 日設置、新規教員も教授 3 名、助教授 1 名が認められた。更に、建物についても 1993 年と 1995 年の二期に分けて建設が行われた。

この間の岩崎氏の粘り強い交渉には本当に頭が下がる。私自身も、その後に計算物理学研究センターの運営と概算要求や計算科学研究センターへの改組拡充に関係して、似たような経験を持つことになり、なかなか思うように進まないことに苦しい思いをした。その時に支えとなったのは、CP-PACS プロジェクトや計算物理学研究センターの立ち上げを通じて培った、研究を大きなスケールで進めるということは、単に研究者と研究予算だけではなく、それを支える研究組織や建物等のインフラ整備と表裏一体であるという考えと、概算要求などの経験であったと思う。

5. CP-PACS による素粒子物理学の成果

CP-PACS が、設置当初のテストを経て物理計算に取り組めるようになったのは 1996 年 6 月頃である。既に格子 QCD の quenched hadron spectrum の計算プログラムの準備はできており、本格計算への努力が始まった。またこの夏にはフレーバー数 2 の動的クォークを取り入れた hybrid Monte Carlo プログラムが書かれ、full QCD 計算の準備が整った。

1992年9月にはCP-PACSが2048ノードの最終構成になり、Linpackベンチマークで世界最高速を記録した後、10月4日から運用が開始された。この後、約1年をかけてquenched QCDでのハドロン質量スペクトルを求める計算が行われ、また半ば並行する形でフレーバー数2の動的クォークの系統的計算が進められた。これらの結果は、1997年3月にCP-PACSプロジェクトの活動報告の一環として行われた国際会議”Lattice QCD on Parallel Computers”を皮切りに、国際舞台への報告が始まり、以降数年間の格子QCDにおける世界の研究をリードすることとなった。特にquenched light hadron spectrumの計算結果は1980年頃に始まる格子QCD計算の里程碑の一つとみなされる結果となっている。

CP-PACSプロジェクトが1997年3月に終了した後、同年8月29日に日立側のお世話で、品川の日立和彊館でプロジェクトの打ち上げの催しが行われた。その席で私にも挨拶をする機会が与えられ、私はquenched light hadron spectrumの結果を「美しい」という言葉で表現した。CP-PACSに込められた様々なアーキテクチャ上の工夫、それを実現した実装技術、その上に立った格子QCDの計算とその結果は、それぞれに多くの関係者の多大な労苦を秘めながらも、beautifulという言葉が一番ふさわしいというのが私の実感だったのである。

6. おわりに

CP-PACSは1996年10月4日に本格稼働を開始し、その後9年間稼働を続けて、2005年9月29日に稼働を終了した。この間のパワーオン日数は2813日、平均稼働率は86%であった。CP-PACSの開発・制作とCP-PACSによる計算から生み出された研究成果は、学術論文80編、国際会議発表147件に上る。稼働終了の当日には大学側と日立側から関係者約40名が集まり、岩崎氏が最後のshutdownコマンドを投入するのを見守った。この後、CP-PACSは解体されたが、事前の相談に基づき、PU、ネットワーク、IO、電源各1筐体と開発関係書類を国立科学博物館に寄贈し、計算科学研究センターには、PU 筐体とネットワーク筐体各1を展示している。

CP-PACS の思い出

金谷和至

筑波大学物理学系／計算物理学研究センター(当時)

筑波大学数理物質科学研究科(現在)

CP-PACS プロジェクトには、その開発計画の最初の立ち上げから、物理の実計算で実効性能を出すためのプロセッサ・アーキテクチャの検討、マシンとしての完成、それを使っての格子 QCD の大規模シミュレーション、最後のシャットダウン、さらには科学博物館への寄贈や、現在計算科学研究センターのエントランスホールに設置されている展示モデルの製作に至るまで、思えば18年間も参加させていただきました。私にとってこの18年は、研究者としての業績の大部分を積み上げた重要な時期であり、CP-PACS はいつもその中心にありました。

私は、CP-PACS の先代の QCDPAX から、物理と計算機工学が交錯したこの奇妙な世界に住んでいます。元々は計算機そのものが特に好きというわけでもなく、まわりでPC98 や Windows を使っている人たちもいましたが、その場でしか使えない呪文だらけの世界には馴染めませんでした。研究のために、純粋にユーザとして Cyber 205 や CRAY X-MP を使って、格子ゲージ理論のシミュレーションをしていたのですが、なかなか職が見つからず、ポスドクとしてドイツやスイスを放浪していたところを岩崎先生に拾っていただき、1988 年の末に筑波に来ました。そこで、QCDPAX プロジェクトに途中から参加し、生まれて初めて、計算機も誤動作しうることを学びました。QCDPAX をプログラムどおりに動作する計算機にするために、四苦八苦しながらコンパイラのバグや確率的に出現するハードの誤作動と戦い、また、計算機のハードウェアの制限内でどうすればやりたい計算を最も高速に実行できるかのパズルに挑戦したり、毎日新しいことと格闘しました。そして、正しく動作するところまで持ってきた後に初めて、それまで押さえていた物理の計算を走らせて、誰も出来なかった規模でいろんなアイデアを試してみる。その結果を発表したときには、本当にゼロから研究したの

だという、単なるユーザとしてスパコンを使っていたときには感じられなかった強い充実感が得られたのを憶えています。

と、このように、住み慣れた物理の世界の外側で、命令に従わない電子回路や想定外のアセンブラをはき出すコンパイラと格闘することにも結構楽しんでいましたが、QCDPAXの次のプロジェクトとしてCP-PACSプロジェクトが開始したときには、こうした苦労はほとんど無くなるのだろうと思っていました。計算機としての規模と集積度が格段に大きくなり、QCDPAXにまだ残っていた手作的な要素が排除されて、大メーカーが完璧なものを持ってきてくれるだろう。だから、物理のプログラムで実性能が出るよう、ハードとソフトの基本設計の所さえ注意して押さえれば、あとは物理を考えるだけで良いだろう。

実際は、技術の限界に挑みながら最高性能を引き出そうとしているのだから、思いどおりに動くものがそのまま出来るはずもありませんでした。むしろ、規模と集積度が大きくなった分だけ、ハードやソフトがよりブラックボックス化し、誤動作の原因をさぐるのがもっと難しくなりました。幸い、日立の有能な技術者集団がいつも近くでインテンシブに取り組んでくれたので、いっしょに一つ一つ問題を解決してゆくことが出来ました。

関連するエピソードはいっぱいありますが、特に印象に残っている中では、こんな事がありました。開発の初期には、日立のハード・ソフトの専門家と、2週間に1回くらいのペースで、濃厚な技術的検討会議をもっていたのですが、あるとき、私がやった試験計算の報告の中で、通常の組み込み関数を使った結果と、疑似ベクトル化された高速なライブラリ関数を使った結果とで、有効桁数の最後の方がわずかにずれていたことに言及しました。私自身は、関数によって内部の計算の順番が違うだろうから、こんなこともあると思って、バグを指摘したつもりではなかったのですが、それを聞いていた日立の方がすぐに、全ての桁で同じ結果になるはずだと気付かれて（演算の順番が同じという意味か、内部の有効桁数に余裕を持たせてあるという意味か忘れましたが）、それから、疑似ベクトル関数ライブラリのバグがいくつか見つかりました。

こんな事もありました。開発の終わり頃だったと思いますが、コンパイラが

新しい版に変わったときに、私の QCD テストプログラムの結果がおかしくなってしまう。この頃には QCD プログラムの高速化もかなり進めていて、プログラムセットとして複雑なものになっていたのですが、シンプルな初期のプログラムでは、新しいコンパイラでも正しい答えを出すことがわかった。そこで、それまでのバグ取りの定石に従って、まずプログラムを段々と元に戻して、おかしくなるぎりぎりをみつけ、次にそのプログラムのどの計算が変なのかを特定するために、新旧両方のプログラムセットで、プログラムを削って段々とシンプルな物にしてゆく。そんな作業を1日中やって、かなり簡単な計算でも結果が違うところまで来た。簡単な計算で結果が変わる理由がわからなくて、ここから先はインスピレーションの世界と、いろんな試行錯誤の結果、明け方近くの朦朧としはじめた頃にやっと、プログラム本体ではなく、そのセットに付いている Makefile のコンパイルオプションの違いが原因になっていることを見つけました。高速化のために、ループの書き方とコンパイルオプションの組み合わせを、ある版から変えていましたが、定石に従ってプログラムを小さくしているときには、そこまで気が回らなかった。元々のプログラムでコンパイルオプションをどうすれば変になるかまで確認して、結果を宇川さんに報告して、後は力尽きて研究室のソファに倒れ込みました。この後で、バージョンアップで改良された部分に、コンパイラのバグがみつかりました。

結局、一杯苦労もしましたが、周りにいろんな分野の優秀な人たちが揃っていて、お互い得意なところで密接に協力して、全体として創造性と生産性が極めて高い開発が進められたと思います。実際、その後に動き出した CP-PACS は、プロジェクトで開発された計算機としては格段に安定しており、世界と戦う上で強力な武器になりました。

物理としては、プロジェクトの最初の立ち上げ時の目標に従って、クエンチ近似(クォークの対生成・対消滅を無視する近似)におけるハドロン質量の計算をやりました。王道の真ん中を圧倒的な計算力で押し開きながら進むような研究で、物理の進め方として私自身も一杯勉強になりましたが、その成果も、CP-PACS の名前を不動のものにするに十分なものでした。クエンチ近似は、格子 QCD を現実の計算機で計算可能にするために、格子

研究の最初期から広く用いられてきた近似法で、現実とそれほどずれないだろうと思われてはいましたが、どこまで正しいかは不明でした。CP-PACSにより、クエンチ近似の有効性と限界が同時に実証され、クエンチ近似の時代にある意味幕を下ろす事が出来ました。

私にとってこのクエンチ近似の研究以上に誇りに思える研究は、それに引き続いて行ったフル QCD の研究です。CP-PACS プロジェクトを立ち上げた当初は、CP-PACS の計算能力ではまだまだ難しいだろうとの判断で、クォークの対生成・対消滅の効果を正確に取り入れた格子 QCD (フル QCD) の本格計算は、プロジェクトの目的には含まれていませんでした。しかし、クエンチ QCD の研究と同時に進めた有限温度 QCD の研究などから、改良格子作用と新しいシミュレーションアルゴリズムを組み合わせれば、CP-PACS の計算能力でも、フル QCD で最初の系統的研究を実現できるかもしれないと考えるようになりました。そこで、クエンチ計算を完了させるとすぐに、フル QCD 計算に取りかかりました。そして、フル QCD でも、王道とは何かを示すような、最初の系統的大規模シミュレーションが行われました。つまり、1台の計算機により、その分野のマイルストーンとも言える業績を2つ続けて出すことができました。

これは、いくつか時代の幸運もプラスしていますが、岩崎先生と宇川先生がまとめた物理チームの創造性が極めて高かったことと、物理・計算機工学・メーカーの合同チームが開発した CP-PACS が高い性能を最初から安定して出し続けていたことの両方が、最良の形で組み合わせさせた結果と思います。このようにアクティブなチームに最初期から参加し、私にもいくらかは貢献することができたのは、幸福な経験でした。次の世代に、その経験のいくつかでも伝えられればと思っています。

CP-PACS プロジェクト終了にあたって思うこと

吉江 友照

筑波大学物理学系／計算物理学研究センター(当時)

筑波大学数理物質科学研究科／計算科学研究センター(現在)

私は計算素粒子物理学の研究者の一人として、CP-PACS プロジェクトに参加させて頂いた。CP-PACS の記念冊子への寄稿をしたためようと、十余年にわたる CP-PACS プロジェクトを振り返ると、実に色々な形でこのプロジェクトに関わってきたものだと、感慨深いものがある。CP-PACS の開発初期段階では、計算素粒子物理学のアプリケーションである QCD シミュレーションプログラムを分析しベンチマークプログラムを纏めた。設計段階では、計算機工学者やメーカーである日立製作所の提示するアーキテクチャに基づいて演算・通信・ディスク 10 の各側面での机上性能評価を行い、CP-PACS のアーキテクチャの決定に寄与した。CP-PACS 完成後は、プログラムの開発チューニングに携わり、QCD シミュレーションのマイルストーンとも呼ぶべき『クエンチ近似のハドロンスペクトラム計算』を自ら実行した。また、CP-PACS 稼働期間中は一ユーザーとしてのみならず、計算機システム運用委員の一人として、主として素粒子物理ユーザーと計算機システム及びメーカーとの橋渡役も行なわせて頂いた。実働部隊の一人として、プロジェクトの多くの側面を勉強させて頂き、微力ながらも各側面でプロジェクトに寄与できたのは、望外の幸せであった。

私にとっての CP-PACS プロジェクトのハイライトは、上述の『クエンチ近似のハドロンスペクトラム計算』である。その素粒子物理学に於ける意義や成果の詳細は、本冊子の『CP-PACS の成果、計算素粒子物理学』の章に纏められているので、ここでは、スペクトラム計算と計算機としての CP-PACS との関わりについて、既に種々の報告書にまとめた事柄も含むが、本記念冊子の本文で詳細に述べる事が出来なかった内容を、雑記風にまとめておきたい。

スペクトラム計算の主要なサブルーチンである QCDMULT や QCDMR

の演算の特徴は、ひとことで言えば、『ベクトル計算機向き』である。つまり、ロードストアの割合が加算乗算演算のそれに匹敵する。(コーディング手法にも依存するが、QCDの場合、概ね、加算:乗算:メモリアクセスは、1:1:1である。)キャッシュに頼らず高いメモリスループットを確保し、メモリアクセスレイテンシを隠蔽する為に、CP-PACSでは、バンクメモリとスライドウィンドウによる疑似ベクトル機構が導入された。CP-PACSプロジェクトでは新たなベクトルプロセッサを開発するのは不可能であったので、現在でも最善の選択であったと思っている。但し、「疑似ベクトル計算機はベクトル計算機に比べて、高い効率を実現するのにユーザーの努力をより必要とする計算機である」というのが、正直な感想である。ベクトル計算機では、フォートランレベルでのコーディングの工夫とチューニングで、これらのサブルーチンで80%程度の効率を実現するのはそれ程困難ではないが、CP-PACSでは30%の効率を実現するのがやっとであった。結局、我々は、QCDMULTやQCDDMRのアセンブラコードを開発し、それを実際の計算に用いることになったわけである。この場合、60%以上の効率を実現できた。参考までに、フォートラン版とアセンブラ版のQCDMULTやQCDDMRの性能の時間方向格子サイズ依存性を図1に再録した。アセンブラ化による2倍以上の高速化は、CP-PACSで多くの素粒子物理の成果を得る事におおいに寄与したと思っている。なお、QCDMULTのアセンブラコードは、最初のバージョンを日立製作所に作製して頂いた。この場を借りて、改めて深謝したい。QCDDMRは私が作製した。実は、私が実アプリケーションで使われるアセンブラコードを書いたのはこれが唯一であり、貴重な体験をしたと思っている。

並列計算機としてのCP-PACSの特徴の一つは、ブロックストライド転送をサポートしたリモートDMA機構であるが、これもQCDの通信パターンを分析し、それを高速に実行する為に開発された方式である。その過程で私が一番興味深かったのは、「メモリコピーが意外に重い」という点である。ブロックストライド転送はユーザーレベルのgather/satterのコストを削減し、リモートDMA機構はユーザーシステム間のメモリコピーを削減する。この様な工夫によって、QCDの実アプリケーションでピーク性能の50%の

性能を達成できた(時間方向の格子サイズが 40 以上の場合). 今後も幾つかの並列計算機の開発プロジェクトに参加する機会もあるかと思うが, その際には, CP-PACS での経験を生かして, ユーザーの立場から, 「メモリコピーをギリギリまで削減した通信」の提案をしていきたいと思っている.

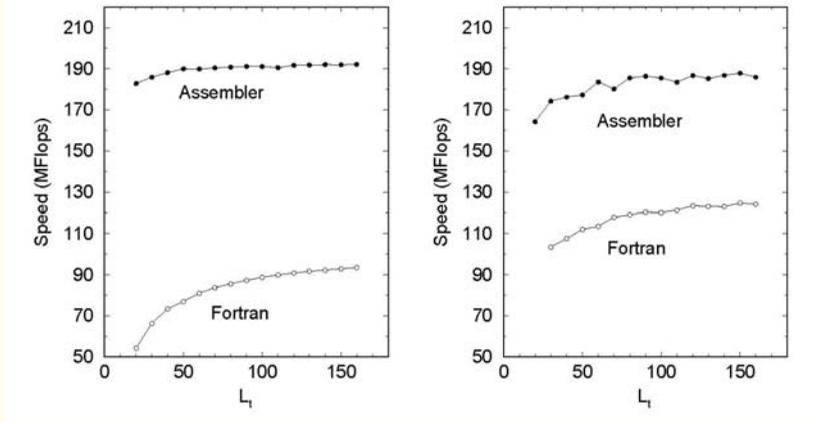


図1. QCDCMR(左)とQCDCMR(右)の実行性能の時間方向格子サイズ L_t 依存性. 格子の時間方向のdoループを疑似ベクトル化し, 空間方向はその繰り返しとした. Even/odd algorithmを用いているので, ベクトル長は $L_t/2$ である.

計算機の性能の指標の一つはディスク I/O 性能である. 特にクエンチ近似のスペクトラム計算では, 問題規模やパラメータにも依るが, 実行時間の 15-20%がディスク I/O に費やされ, これを如何に削減するかも CP-PACS の課題であった. 実際, CP-PACS の設計段階から QCD プログラムのディスク I/O を分析し, I/O ノードの設計に反映した訳であるが, 私がもっとも興味深かったのは, 「I/O 性能を引き出す為には, ディスクへのシーケンシャルアクセスが必須である」という点である. CP-PACS では 16 個の PU からのディスク I/O が一つの 10U に集中するが, クエンチ計算では, 各 PU からの I/O 命令をシリアルライズする事によって, 同一ファイルがディスク上に連続に配置される様工夫した. その他, あらかじめディス

ク上の全てのファイルを消去しておく、処理の軽い C の 1/0 関数を大きなバッファを確保して用いる等の工夫によって、クォーク伝搬関数の出力段階のシーケンシャルな書き込みでは、約 5MB/sec/10U の性能を達成した。これは、ディスク 1/0 の理論ピーク性能の 25% である。クォーク伝搬関数からハドロン伝搬関数を構成するステップは、論理的にランダムな読み込みが発生し、1.3MB/sec/10U の実効速度となった。

以上の様に、クエンチ近似のハドロン質量計算では、演算、通信、ディスク 1/0 の全ての側面で、殆んどギリギリまでチューニングしたプログラムを作製した。よくここまで効率的にシステムを使い果たしたものだ、感慨深い。

並列計算機としての CP-PACS の特徴の一つは、パイパークロスバネットワークである。この方式の利点は多々あるが、QCD 計算に限れば、システム全体を複数のメッシュトラスのパーティションに分割しての運用が可能にした点が、最大の利点であったと思う。実際、CP-PACS 全体を一つのジョブで占有した実アプリケーションは、クエンチ近似の最大サイズの計算で、それは約半年の期間だけであった。運用期間の殆んどすべてで、CP-PACS は計算の粒度に応じて臨機応変に種々のサイズのパーティションに分割して運用された。『CP-PACS の成果、計算素粒子物理学』の項に纏められている様に、CP-PACS では、QCD に限っても実に色々な計算が行われたが、それらを効率良く実行できたのは、「柔軟な分割運転が可能であった」ことに依るところが大きかったのではなかろうか。この点は、なかなか定量的に評価できない事であるが、パイパークロスバという QCD には一見贅沢なネットワークに費用を投じるのは、正しい選択であったと思う。

最後に、CP-PACS の運用に関して感ずることを述べたい。計算物理学(科学)研究センターでは、CP-PACS やそのフロント計算機システムの運用を『計算機システム運用委員会』が担当してきた。特筆したいのは、「運用委員会には、計算機工学者だけでなく、CP-PACS ユーザーである物理サイドの各分野の代表も加わった」点である。システムの問題点や改善すべき事項に最初に気づくのは、多くの場合、実際に計算機を使って

いるユーザーである。その分野のユーザの研究や計算をよく知っている運用委員会のメンバ(ユーザー自身の場合も多々あった)が問題を的確に把握-整理し、計算機工学者やメーカ担当者と協力して問題解決にあたった。ユーザーは我儘なもので、自分達の研究 / 計算の進行がシステムの問題が原因で遅延するのは我慢ならないと感ずるものである。問題の解決を人任せにせず、問題解決の責任の一端をユーザー自身が負っているという方式は、問題解決にかかる時間を短縮するのに、おおいに寄与したのではないかと思う。CP-PACS の成功の一因は、計算機的设计段階から「物理研究者と計算機工学者とメーカが同じ土俵にたって緊密に協力した」ことであるが、緊密な協力関係が10年近い長い運用期間に渡って継続的に維持されたことが、CP-PACS の高い稼働率に、ひいては、CP-PACS による多くの物理学上の成果につながったのだと思う。

2006 年 4 月吉日

CP-PACS プロジェクトの思い出

青木 慎也

筑波大学物理学系(当時)

筑波大学数理物質科学研究科(現在)

私は1991年8月半ばに筑波大学の物理学系に素粒子理論の助手として赴任した。その時点で CP-PACS プロジェクトは(もしかしたら非公式の検討段階だったかもしれないが)既に立ち上がっていたようであり、(正確な日付は覚えていないが)私もすぐにこのプロジェクトのメンバーに加わった。未だに計算機に関しては素人同然であるが、当時はコンピュータのハードのことはほとんど知らず、CPUという言葉は聞いたことがあったが、レジスタ、キャッシュなどはもちろん、メモリやディスクの区別なども出来ていなかった。そのような素人が超並列計算機の製作という(今ではどこでもやっているが)当時の最先端の研究プロジェクトに参加したのであるから、いろいろ分からないことが多く苦勞した。この寄稿文では、CP-PACS プロジェクトにおいて、計算機の素人である私がどのようにプロジェクト内で居場所を見つけ、何らかの貢献が出来たのかを振り返っていきたい。

私がプロジェクトに参加する前に、岩崎さん、中澤さんらを中心に既にいろいろな検討がなされていたようであり、特に、CP-PACS の CPU として PA-RISC を使うことが決まっていた。この CPU は当時の量産 CPU の主流であった単精度の演算器ではなく倍精度の加乗算演算器を持っており、150MHz で動作し、1つの CPU あたり倍精度で 300MFlops の性能であった。この CPU を 1024(後から 2048 に倍増)個並列に繋げたものが CP-PACS である。PA-RISC を使った場合に、格子 QCD(以下では単に QCD と呼ぶ)の計算がどのくらいのスピードで出来るか、が問題であり検討が行われた。岩崎さんは、「QCD の主要部分の計算性能を検討したところ、ピークの10%程度(この数字はうろ覚えなので間違っているかもしれませんが)しか出ないと分かり非常にがっかりして、検討会の後のビールがとても苦かった。」と当時のことを述べられている。初めて私が参加した検討会は、プロジェクトの正否を握る CPU の性能が出ないことが認識された直後のも

のであり、この問題の解決策として、スライド・ウインドによる疑似ベクトル化のアイデアが中澤さんから提唱された。前の検討の背景を知らなかったのでこの提案のポイントや重要性は分からなかったが、岩崎さんが「これでこのプロジェクトはなんとかかなりますね。」という言葉に印象深く覚えている。スライド・ウインドに関しては他の専門家の正確な説明があると思うので、ここでは素人からの“解釈”を述べる。QCD 計算は、他のアプリケーションに比べて、演算に対するメモリアクセスの比率が格段に高い。演算器は早くて安価ものがどんどん出てくるが、それに比べてメモリの性能は向上せず、QCD では実行性能がでない。今までの高価なベクトル計算機では、ベクトル化によりQCD でも性能が出たが、PA-RISCというスカラCPUはその性能をキャッシュに頼っている。しかし、QCD ではキャッシュが機能しない、というのが前回の検討結果であった。1つの解決策は、イタリアの APE グループが採用したように、多くのレジスタを用意して、そこに前もってデータを送り込んだり、また、計算後のデータをレジスタに置いて再利用したりすることで性能向上を図るものである。しかしながら、PA-RISC は互換性のために使えるレジスタの数は32と決まっており、APE のように128ものレジスタを使うことが出来ない。その解決策がスライド・ウインドである。まず、物理的なレジスタを128用意する。PA-RISC が演算に使える論理的レジスタは32であるが、その32の連続した論理レジスタを128の物理レジスタの中から動的に選ぶのがスライド・ウインドである。これは128のレジスタのどこを論理レジスタの先頭にするかで実現される。計算に必要なデータは計算に使われていないレジスタに前もって送っておく(プレロード)、あるいは、計算した結果はレジスタに残しておき後からメモリに戻す(ポストストア)ことで、メモリアクセス時間の遅れを隠蔽するのである。32の論理レジスタのウインドが次々とスライドして演算を行うことから、スライド・ウインドという名前がついた。レジスタに前もって置いたデータを次々処理していくので疑似ベクトル化とも呼ばれる。スライド・ウインド機能を PA-RISC に付加することで、PA-RISC との互換性を損なうことなく、また、比較的安価に性能向上が図れる。

スライド・ウインドによる疑似ベクトル化により QCD の実行性能の向上が

見えてきたが、どの程度早くなるかを詳細に調べる必要があった。通常であれば、コードをコンパイラにかけて性能を計測し、演算順序や配列の並べ方などを試行錯誤してチューニングしていくのであるが、スライド・ウインドという新しい機能が加わったため、コンパイラ自体も新たに作成する必要があった。また、スライド・ウインドの説明からも分かるように、データのやり取りを計画的にスケジュールしないと性能が出ない。そこで、QCD のもっとも主要な計算の1つである Mult と呼ばれる部分をアセンブラでチューニングすることになった。Mult は、大規模疎行列をベクトルに掛ける演算である。フォートランしか使ったことのない私にはアセンブラの敷居は非常に高いように見えたが、当時、筑波大にいた山下さんによるコンパイラのためのアルゴリズムの解説が転機となり、物理としての意味が分かっている Mult 部分のチューニングに参加することにした。山下さんの講義では、いろいろな条件(演算、データのロード/ストア、ウインドの切り替え、などに必要なマシン・サイクル数や演算器のレジスタ数など)を与えた時に最適なコードを生成するためのアルゴリズムや、それを実現するための時間遷移のチャート図などの書き方を教わった。この方法では、試行錯誤ではなく、論理的に最適なコードを生成できるので、私の好みにぴったりであった。いろいろな演算パターンにこの方法を適用し最適化の練習をした。そして、その使い方に慣れたところで、“人間コンパイラ”として Mult の最適化に取りかかった。とはいえ、アセンブラ未経験の私がスクラッチからコードを書くのは不可能なので、日立側の作成したかなり性能の良いアセンブラ・コードを基に、いろいろな条件の変化に対応できるアセンブラ・コードを作成した。山下さんに教わったアルゴリズムでまずレジスタの割当をスケジューリングし、それをアセンブラに直していくのであるが、難しいパズルを解いていくような感じで苦労した反面、非常に楽しかったことを今も覚えている。当時は実機がなかったが、スライド・ウインドやメモリ構成を反映したシミュレータが存在したので、出来上がったアセンブラ・コードをシミュレータに掛けて性能を測定した。コード上は単体で90%近い性能が出るはずだったが、D-RAM をいくつかのバンクに分けたメモリのバンクコンフリクトの影響などで80%前半の実行性能であった。その後、CPU やメモリの仕様の変更の

ため、単体で70%の後半ぐらいまで性能が落ちたが、これは現在のインテルなどのCPUに比べると驚異的な実行性能である。Multの実機での性能はシミュレータでの予測とほぼ同じであり、並列化による性能低下を入れても実行性能60%の後半を達成できたように記憶している。(ここは美化して覚えている可能性あり。)このように、計算機の素人である私が少しでもCP-PACSプロジェクトに貢献できたことは、楽しい思い出として残っている。

残念ながら、最近の超並列計算機ではコストの関係上、汎用CPUをそのまま使うので、CP-PACSでやったようなCPUの変更はほぼ不可能である。また、CPUの性能向上に比べてメモリの性能は伸びていないので、CP-PACSの開発時に比べてQCDにとってはますますつらい状況である。並列計算機では並列化による性能低下が大きく問題にされるが、QCDの場合は、まず、単体性能の向上が必須である。現在の汎用CPUでのMultの単体の実行性能は最高でも30%がせいぜいで、通常は20%台、下手すると10%台になる場合もある。逆に、単体性能が悪いので並列化による低下はそれほど顕著ではない。汎用CPUの演算性能もキャッシュに頼る部分が多く、スライド・ウィンドで経験したような精緻なスケジューリングは望むべくもない。(GPGPUは少し違っていて、ベクトル計算機でやっていたチューニングが必要なようである。)当時はいろいろ苦労したが、今から振り返ると、いろいろな意味でなんと幸福な状況であったことか！

CP-PACS プロジェクトと宇宙物理学

梅村 雅之

筑波大学物理学系／計算物理学研究センター(当時)

筑波大学数理物質科学研究科／計算科学研究センター(現在)

私と CP-PACS プロジェクトの関り合いは、1993 年6月の1本の電話から始まった。当時計算物理学研究センター長をされていた岩崎先生から突然お電話を頂き、筑波大学に宇宙物理分野を新設することになったので、筑波大に来ませんかというお誘いを受けたのである。当時、私は国立天文台理論天文学研究系で助手をしており、銀河形成の数値シミュレーションを行いながら、研究系主幹であった観山正見先生(現国立天文台長)と共に、宇宙分野における輻射輸送専用計算機製作プロジェクトに携わっていた。実は、それまで岩崎先生とは面識がなく、恥ずかしながら、CP-PACS プロジェクトのことも不案内であった。電話から1週間後、三鷹の航空宇宙技術研究所に岩崎先生が出張に来られるということで、その近くで初めて岩崎先生にお会いした次第である。そこで、岩崎先生が進められているご研究のことや、今回のポストのことについて小一時間お話を聞かせて頂いた。新天地で宇宙グループを作るということには大変魅力を感じ、赴任の件をお受けした。

また、当時天文台で同じく輻射輸送専用計算機製作プロジェクトに携わっていたのが中本さん(現東工大准教授)で、翌年(2004年6月)に筑波大の宇宙グループの助手に着任した。このため、天文台の輻射輸送専用計算機プロジェクトは頓挫する形となってしまった。

筑波大に着任当時、CP-PACS プロジェクトは既に2年目の最後に入り、計算機アーキテクチャについての詳細な検討が進んでいた。天文台で輻射輸送専用計算機なるものを考えていたとはいえ、知識・経験の面で CP-PACS と比較すると、紙飛行機とジェット機製作ほどの差があり、PVP-SW, Slide Window といったこれまで耳にしたことがない話が交わされ、当初からすっきり落ちこぼれとなってしまった。しかし、CP-PACS を使って

どういふ計算を行えば新しいことができるかということの中本さんと試行錯誤する中で、天文台の輻射輸送専用計算機の青写真が生きてきた。光の伝播を扱う輻射輸送計算は6次元の計算を必要とし、宇宙分野では前人未踏の分野であった。CP-PACSを用いることで、天文台で考えていたアルゴリズムの応用で輻射輸送計算が世界に先駆けて実現できる可能性が見えてきたのである。しかし、当時輻射輸送計算のノウハウはほとんど持っておらず、また実際に計算したとしてもその収束性すら保証されていないという代物だった。従って、輻射輸送計算について、まずは低次元の問題から始めて、これらの経験を積むという段階が必要であった。そして同時に、CP-PACS のアーキテクチャの上で、6次元輻射輸送計算をどのように実現するかという検討も必要になった。これらには、思いのほか時間を要し、何年間かがあつという間に経ってしまった。途中から須佐さん(現甲南大准教授)も宇宙グループに加わり、1999年になってCP-PACSを使った世界で初めての6次元輻射輸送計算が実現し、最初の論文が発表された。この計算により、我々は、非一様宇宙における電離光子の6次元輻射輸送を扱い、宇宙再電離過程の大規模数値シミュレーションを実現することができた。この輻射輸送計算は、その後も発展し重要な論文を製作することができた。また、その頃からCOE研究員の森正夫氏(現筑波大准教授)と共に、超新星爆発の集積効果を採り入れた銀河進化に関する高精度数値流体力学計算をCP-PACSで始めた。これは、地球シミュレータを使った結果と合わせて、銀河が原始銀河から楕円銀河へと進化する過程を解明することができた。この結果は、2006年3月のNatureに掲載された。これらはすべて、長い道のりを経ての成果であったが、共同研究者の協力を得て成し得た価値のある到達点であると思っている。

CP-PACSの輻射輸送計算は、その後も発展し現在の研究につながっている。輻射輸送の計算に流体計算を結合した輻射流体計算への取り組みは、未来開拓学術推進研究事業「次世代超並列計算機の開発」プロジェクト(1997年度～2002年度)で推進したHMCS(Heterogeneous Multi-Computer System)によって、世界に先駆けて実現することができた。HMCSは、CP-PACSに重力計算専用機であるGRAPE-6をネットワークを

介して結合した異機種融合型計算機である。流体計算を結合するとき、流体計算部分そのものは輻射輸送計算より軽いので、ホストで行うことが出来るが、重力計算部分がかかなり重くなる。そこで、重力計算をGRAPEに任せることで、宇宙輻射流体計算が実現できるようになるのである。考えてみると、GRAPEプロジェクトとは、概念設計当時から長い付き合いである。概念論文がNatureに掲載されたのが1990年である。私自身は、GRAPE製作に対してのコントリビューションはほとんどなかったが、GRAPEに近接粒子を返す機能を加えることで、粒子法流体力学の計算が $O(N^2)$ から $O(N)$ にできることを提案し、これを採り入れてもらった。GRAPE(当時は1号機GRAPE-1A)を用いた粒子法流体力学のシミュレーションを初めて行って論文を書いたのは1993年である。聞くところによれば、朴先生は慶応大学にいらした頃から、GRAPEプロジェクトとの関わりがあり、1号機のボードのラッピングのノウハウなども指導して頂いたそうである。筑波大で、HMCSという形で、一緒にプロジェクトを推進することになったのは、何かのご縁であると思う。HMCSプロジェクトは、現在FIRSTプロジェクトという形で受け継がれ、大規模輻射流体力学計算を実現するための新たな融合型並列計算機(FIRST)を製作し、大規模シミュレーションを推進している。天文台から筑波大に移ることになった1993年末から現在まで、この10数年は、宇宙輻射流体力学実現に向けての長い戦いであったように思う。縁あって、CP-PACSプロジェクトに加わったことは、宇宙輻射流体力学実現に向けての重要なステップとなった。現在の成果は、多くの共同研究者に恵まれたお蔭である。中本さん、須佐さんとの協働がなければ、宇宙輻射流体力学は実現しなかったであろう。この場を借りて、深く感謝したい。そして、岩崎前学長、宇川副学長をはじめ、これまで宇宙物理分野の研究を支援し続けてきて頂いた全ての方々に深く感謝申し上げる次第である。

超並列計算への挑戦

～宇宙物理学の多次元輻射輸送問題における一例～

中本 泰史

筑波大学物理学系／計算物理学研究センター(当時)

東京工業大学理工学研究科(現在)

1. はじまり

私が CP-PACS プロジェクトに参加させていただいたのは、宇宙物理グループの立ち上げとほぼ同時の 1994 年です。当時は実機の製作段階にあり、計算物理学研究センターの新しい建物の工事も始まろうとしていた頃でした。私たちの任務は、大規模数値シミュレーションを用いた宇宙物理学の推進でした。その頃の天文学・宇宙物理学における数値シミュレーションと言えば、圧縮性気体の流体力学(あるいは電磁流体力学)シミュレーションと、重力多体系のシミュレーションが主でした。そのような中、CP-PACS というせっきくの大型計算機を生かしてどんなシミュレーションをやるか、宇宙物理グループの梅村さんと検討しました。そして得た答えは、多次元の輻射輸送計算でした。

輻射というのは、ひらたく言えば光とか電波とかのことです。天文学・宇宙物理学において輻射は非常に重要な役割を担っています。まず第一に、天文学的観測によって得られる情報のほとんどは輻射によってもたらされています。望遠鏡で天体を観測するというのは、天体からやってくる輻射を望遠鏡で集めて見るということです。ところが日常経験でも分かるように、遠くにあるものの真の姿を、見た目だけで判断することは実は容易ではありません。霧がかすんでいるかもしれないし、手前と奥の前後関係の判定すら怪しいことがあります。遠方のものの実体を見た目から知るためには、実体と見た目との間の関係が理解できればいいのですが、日常生活で私たちはこのことを一瞬のうちに頭の中でシミュレーションしているわけです。ところが宇宙の現象では地上での経験や直感が成り立たないことがあります。したがって観測天体の見た目から実体を理解するために、輻射の伝

播の仕方をシミュレーションすることが必要になるのです。さて、輻射は宇宙物理学においてもっと物理学的な意義も持っています。輻射は一般にエネルギーを運びます。日の光を浴びれば暖かく感じるのは、エネルギーが太陽から輻射によって運ばれてきているからです。輻射のエネルギー輸送機能により、天体は冷えたり暖まったりします。天体の温度というのはその天体の物理的性質を特徴づける極めて重要な意味を持っていますので、その温度を左右する輻射というのもまた、宇宙物理学において極めて重要な役割を担っているのです。さらに輻射は運動量も運びます。地上ではこれは完全に無視できますが、天体現象においては輻射圧が無視できない場合も多数存在するのです。したがって輻射の運動量輸送機能もまた、宇宙物理学における輻射の重要性を裏付ける根拠の一つとなっています。

さて、このように重要な輻射ですが、それまでの宇宙物理学では2次元・3次元空間内の輻射輸送シミュレーションはほとんど存在しませんでした。そのような計算が困難だったからです。困難の最大の原因は、まともに扱うと計算量が膨大になるということでした。しかし、CP-PACSのような大型計算機を用いればこの困難は克服できるかもしれない。私たちはそう考え、多次元輻射輸送シミュレーションをCP-PACSを用いた宇宙物理学シミュレーションの柱とすることにしました。

2. どうやって計算しようか

目標を定めたままでは良かったのですが、いざそれを実現しようという段になると、いろいろと問題が出てきました。

私たちが計算しようとしていた問題では、空間内の全ての点が他の点と輻射を介して相互に関係合っています。南の端のある点が明るくなったという影響は、北の端の点にも及びます。しかもそれは、途中の伝播の仕方にもよります。光源に近い上流から下流に向かって順に伝播を計算して初めて、北の端における南の端の変化の影響を知ることができるのです。さらには、北端からの輻射が南進して南端にも影響を及ぼします。つまり、全空間点同士の輻射を介した相互作用を、上流・下流関係を考慮しながら、全て計算する必要があるのです。これをCP-PACSという超並列計算機

を使ってどのように計算したらよいか、効率的な並列計算のアルゴリズムがすぐには思いつきませんでした。

もうひとつの問題は先の問題と関連しますが、輻射輸送計算問題をどのように分割して並列化するかということでした。これは、CP-PACSのメモリサイズとも関係しています。CP-PACSの各PUに付随するローカルメモリでユーザが利用できるのは40MB程度でした。扱う問題をこのサイズ以下に分割する必要があったわけです。ところが輻射輸送問題を単純にプログラミングすると、メモリを大量に消費してしまいます。輻射を表す物理量である輻射強度というものが、時間以外に6つの独立変数に依存する量だからです。6つの独立変数というのは、空間の座標 (x, y, z) と光子の伝播方向 (θ, ϕ) 、それに光子の振動数 ν です。これら全ての次元を適当な格子数、たとえば100ずつに分割したとすると、総格子数は $100^6 = 10^{12}$ 個にもなります。これをたとえ2048個のPUに分配したとしても、輻射強度という1変数のためだけに必要な1PUあたりのメモリは3.2GBにもなってしまいます。したがって、何らかの方法によって使用メモリを減らすことが必要でした。

問題は見極められたのですが、その解決策はなかなか見つかりませんでした。ああだこうだと机上テストで悩むうちにどんどん時間が過ぎていきました。状況によっては当初かかげた目標問題を、計算の実行が可能になるまで縮小する必要があるとも考えました。しかし、pilot-3やCP-PACSの実機を利用して並列計算のテストを始めると、少しずつですが並列計算の特徴が実感できるようになってきて並列計算の考え方に頭が慣れ、並列アルゴリズムも思い描けるようになっていきました。そしてようやく、問題の解決策が見えてきました。私の場合は、対象問題の特徴はこう、CP-PACSの特性はこう、だからアルゴリズムはこうあるべき、という理詰めで解決策を見出したわけではありません。それぞれの特徴を頭の中に入れ、ガチャガチャとかき混ぜていろんな角度から眺めているうちに何となくひらめいたアイデアを拾い上げてみたらうまくいった、という感じでした。

3. 問題の解決

問題となっていたうちの2番目の方、すなわち、使用メモリが大きくなっ

てしまうという問題は比較的簡単に解決できました。私たちが取り組んでいた輻射輸送問題では、実は輻射強度という物理量そのものを全空間格子点上で保持しておく必要はありませんでした。必要なのは輻射強度を方向と振動数に関し積分した量だけだったのです。そこでまず、輻射輸送計算に必要な最小限の輻射強度だけを保持しておき、それ以外のものは各点での方向積分に繰り込んだ後はすぐに捨ててしまうことにしました。これによって使用メモリサイズを、先の例で言うと 100^2 分の1に縮小できました。さらには振動数積分も、輻射強度の絶対値が必要な部分とそれ以外の部分に切り分け、それ以外の部分を予め計算しておいてテーブルとして参照することにより、積分時に必要な振動数格子点数を6（要求精度が低い場合は3）にまで縮小できる新しい振動数積分法を考案しました（6点近似法と呼ぶことにしました）。これによりさらに使用メモリを減らせたと同時に、うれしい副産物ですが、振動数積分の精度を上げることもできました。

もう一方の問題、すなわち、空間の上流・下流関係を保って全空間格子点同士の相互作用を計算するという問題は、もう少しやっかいでした。私たちの対象はメモリの問題もあり、並列化の際には方向・振動数次元ではなく、位置の3次元空間を分割して並列化せざるを得ませんでした。その場合、下流側の空間を担当するPUでは上流側のPUでの計算の終了を待ってから計算を始める必要があります。さらには、輻射の伝播方向は上下・左右・前後・斜めすべて（全立体角方向）ですから、方向毎に上流・下流の関係が変わります。このような状況を超並列計算機で効率よく並列計算するにはどうしたらよいか。なかなか難しい問題でした。解決策は、並列計算の際のPUの動作の様子を見ていて思い付きました。

実際の対象は空間3次元ですが、簡単のために2次元平面問題で説明しましょう。計算領域の南西の端にある点から出た北（からほんの少しだけ東）向きの輻射の伝播を計算しようとする、まず最初に南西端の領域を担当するPUで計算し、その後、それに隣接する北側と東側のPUで計算し、以下それを北東端に達するまで繰り返す、となります。こうすることによって、上流・下流の計算順序を守ることができます。次に同じ点から出るもう少し東向きの輻射、例えば北北東向きの輻射の伝播を計算しようとする

と、PU の動作は北向きの場合と全く同じになることがわかります。つまり、最初のステップでは南西端担当の PU が動作し、次のステップではそれに接する北隣と東隣の PU が動作するわけです。物理的には北向きの輻射と北北東向きの輻射は別物なのですが、空間分割された領域が割り付けられた PU 空間内での稼働 PU 群に色づけしたら、その色の動きは全く同じになるのです。調べてみると、2 次元平面問題の場合には稼働 PU 群の動作パターンは 8 つに、3 次元問題では 24 になることがわかりました。私たちは対象の 3 次元問題を扱うために物理的方向を 100^2 もの格子に分けていますが、PU の動作パターンはわずか 24 しかないのです。ですから、 $400 \sim 500$ ($\doteq 100^2/24$) もの方向格子が同じ PU 動作パターンで並列計算を行うのです。このことに気づいたことが、問題解決の糸口になりました。

再び 2 次元平面問題で説明します。北向き(よりほんの少し東向き)の輻射の伝播を計算する場合、まず最初のステップで南西端の PU での計算のみを行い、その計算終了後、その北隣と東隣の PU にデータが送られ、2 ステップ目でそれら 2 つの PU での計算が行われる、というふうに計算が進みます。このとき、1 ステップ目の稼働 PU 数は 1、2 ステップ目でのそれは 2 です。3 ステップ目では、さらにそれぞれの北隣と東隣の PU での計算が行われ、このときの稼働 PU 数は 3 となります。南北、東西、各方向に並べられた PU 数を N_{PU} とすると(この場合、全 PU 数は N_{PU}^2 です)、北向き輻射の計算は $2 N_{PU} - 1$ ステップで北東端に達し完了します。ある瞬間に計算をしている PU 群を PU 空間内で色づけしたとすると、北西から南東に向かう 1 本の線上に並ぶことが分かります。そして計算ステップを進めるとこの線が南西から北東に伝播し、それによって上流から下流に向かう計算順序を守りつつ全空間での計算が行われることとなります。

しかし、このままでは並列化率が極めて低いことがわかります。ある瞬間における稼働 PU 数の全 PU 数に対する割合は、最大でも $1/N_{PU}$ 程度にしかなりません。1 ステップ目などではわずかに $1/N_{PU}^2$ です。しかし実は、先に述べたように私たちの問題ではたくさんの方向格子に対する計算が同じ PU 動作パターンのできるので、このことを用いて稼働 PU 数を増やすことができます。北向き方向の計算の 2 ステップ目の時、それより少し東向きの

別の方向の計算を南西端の PU で始めればよいのです. この方向は北向きとは別の方向ですが, PU 空間内における稼働 PU 群のパターンは北向きの場合と同じなので, このようなことができるのです. さらには, 最初の北向き方向の計算が 3 ステップ目に入ったとき, 2 つ目の方向の計算を 2 ステップ目に進め, さらに 3 つ目の方向の計算の 1 ステップ目を南西端 PU で始めるのです. これを繰り返すと, 最初の計算を始めてから $2 N_{PU} - 1$ ステップ目では北東端の PU で北向き方向の計算をしていることになりますが, その南隣と西隣の 2 つの PU では 2 つめの方向を, それらのさらに南隣と西隣の 3 つの PU では 3 つ目の方向を計算しています. そしてそのとき南西端の PU では $2 N_{PU} - 1$ 番目の方向の計算を行っています. つまり, $2 N_{PU} - 1$ 個の方向が同時に計算されている状態になるのです. ここで, $2 N_{PU} - 1$ の値はせいぜい 80 程度(3 次元の場合は $3 N_{PU} - 2$ となって, せいぜい 40 程度)であり, 先に求めた PU の 1 動作パターンあたりの方向格子数(400 ~ 500)に比べればずっと小さいということに注意すると, これからしばらくの間はそれぞれの PU は別々の方向を計算しながらも, 全ての PU が稼働し続けることがわかります. こうして長い時間にわたって全 PU が稼働する状態が続くため, 問題全体として高い並列化率が得られました.

2 次元平面問題の場合, ある瞬間にある方向を計算している PU 群を PU 空間内で色づけしたとすると, 2 次元 PU 空間全体は全部で $2 N_{PU} - 1$ 本の線に色分けされて埋め尽くされます. 3 次元問題の場合は PU 空間も 3 次元となり, 同じ方向を計算している PU 群は平面を形成します. そして 2 次元の場合と同様に, $3 N_{PU} - 2$ つの平面で 3 次元 PU 空間を埋め尽くします. 計算ステップを進めると, この面が順に下流に伝播していきます. この計算法では PU 空間を複数(Multiple)の波面(Wave Front)が伝播しつつ埋め尽くすことによって, 上流・下流の計算順序を守りつつ, 高い並列化率が得られています. そこで私はこの計算法を, Multiple Wave Front 法と呼ぶことにしました. PU 空間内を 1 つの波面を伝播させることによって上流・下流の計算順序問題は解決できたわけですが, さらに複数の波面を同時に存在させて PU 空間を埋め尽くすことができることに気づいたことで, 並列化効率を上げることができたわけです.

余談になりますが Multiple Wave Front 法の原理を思いついたあと、プログラミングして実装し、テストを行って有効性を確認するのに、ある年の夏まるまる 1 ヶ月をかけました。このとき私はまだ独身でしたが、ほとんど家に帰らず研究室で寝泊まりし、根を詰めて作業していました。プログラムが物理の実問題に応用できるようになったのはさらに数ヶ月後でしたが、プログラム開発にとってはあの夏休みの集中的作業が最も重要であり、鍵になったと思います。その後、私自身はこのようなやり方でのプログラム開発は、体力的、精神的、社会的な理由により、なかなかできなくなりました。

4. その後

なんとか問題が解決し、当初の目標であった 3 次元輻射輸送シミュレーションを実行することができました。CP-PACS を利用した結果、シミュレーションの規模は最大 $128^3 \times 128^2 \times 6$ 格子にもなりました。この数字は自分たちで言うのもなんですが、驚異的な値です。宇宙物理分野で輻射輸送シミュレーションを行っている人に言うと、今でもびびりされます。この計算の規模は、世界の数年先を行っていたと自負しています。

さて、振動数積分に関する 6 点近似法も並列化効率を上げるための Multiple Wave Front 法も、CP-PACS を有効に使って大規模計算をしようという必要に迫られて考え出されたものです。(またまた余談ですが、宇宙物理グループの人達は格子 QCD グループの人々と違い、計算機を効率良く使い尽くそうという厳しさがあまりありませんでした。実際、計算機を少し無駄に使っていて、お叱りを受けたことがありました。そういうことも、私たちの新アルゴリズム開発の強い動機になっていたように思います。)しかし、ここで考案した手法は CP-PACS 以外の計算機にも応用することができました。また、宇宙物理の対象問題が変わっても類似の多くの問題に応用できます。実際、その後現在に至るまで、利用計算機が変わっても私たちはこれらの手法を利用し続けています。さらには、外国のグループもこれらの手法を取り入れるようになってきました。このような状況を見るにつけ私は、並列プログラミングと格闘したあの夏のことを思い出しつつ、ちょっとした喜びに浸っています。一方で、あのアルゴリズムはもっと改良できそうだなあ、

改良してもっといろんな問題に応用したいなあ、とも考えています。実際には、なかなか思うようには動き出せないのですが。

5. 御礼

最後になりましたが、計算物理学研究センターでの宇宙物理シミュレーションの機会を私に与えていただき、陰に陽にご支援をいただいた、岩崎先生をはじめとする CP-PACS プロジェクトの皆様、日立製作所等関係企業の皆様、計算物理学研究センターの皆様、筑波大学の関係者の皆様、天文学関係者の皆様、そして宇宙物理グループの梅村雅之先生と須佐元氏(現 甲南大)に、この場をお借りして心よりの謝意を表したいと思います。ありがとうございました。

QCDPAX から CP-PACS へ

小柳義夫

東京大学理学系研究科情報科学専攻(当時)

工学院大学情報学部(現在)

1. 並列計算機の誕生

現在の計算科学研究センターの前身である計算物理学研究センター(1992~2004)ができるきっかけになったのが、QCDPAX という QCD(量子色力学)専用の並列計算機である。CP-PACS を記念して、その前史を語りたい。

1970 年代後半といえば、世界初の商業的に成功したベクトル計算機 Cray-1(1976)が登場し、世界初の本格的並列計算機 ILLIAC-IV(1975)が稼働し始めたころである。日本ではそのころ、富士通の FACOM230-75APU(1977)、日立の M200H-IAP(1979)などの、メインフレームへの付加ベクトル演算器が登場した。星野力と川合敏雄両氏は、1977 年頃、当時登場し始めた CMOS の汎用 CPU チップを用いれば、安価に科学技術計算用の並列計算機が構成できることを主張した。2 人の考えた対象は原子炉などの偏微分方程式で記述される連続系であり、必要なノード間通信は隣接通信が大部分であることから、2 次元メッシュネットワーク(両端を閉じるので実際は 2 次元トーラス)が最適であり、隣接ノード間に共有メモリを置くことにより簡単に実現できることを示した。

このアイデアにより、PACS-9(1978)、PACS-32(1980)、PAX-128(1983)、PAX-64J(1984)などが星野研究室で製作された。筆者が初めて見たのは、PACS-32 であった。PAX-128 で 2 次元 Poisson 方程式を解くと、当時の学術情報センターのメインフレームとほぼ同じ性能を出すことができ、商品と実験機の違いはあるが価格性能比は 2 桁優位であった。汎用チップを用いた並列コンピュータとしては、有名な Caltech Cosmic Cube(1981~3)よりも早く、先駆的なコンピュータであった。筆者はこの計算機が QCD のシミュレーションにも利用できることを直感し、共同研究を始めた。1983 年には東

大の大型計算センターにベクトルコンピュータ HITAC S810 (ピーク 800MFlops) が設置され、これを利用して研究が大幅に進展したが、それでも QCD の研究は計算性能がネックであり、少なくともさらに1桁から2桁高性能なコンピュータが必要と思われた。

2. QCDPAX プロジェクトの発足

価格性能比がいくらいいと言っても、実用的な並列コンピュータを建設するには億の単位の予算が必要である。当時の科研費は高々数千万であり、これでは作れない。なにか方策がないかと星野氏を中心に必死に文部省(当時)などを回って可能性を探った。

そのころ科研費に、従来よりは大型の予算が与えられる特別推進研究という新しい種目ができた。しかも故西島和彦先生の推薦で書類審査をシードできるという幸運に恵まれた。しかし道はそれほど簡単ではなかった。1984 年から出し続けたが、86 年まで 3 回連続で不採択となった。ヒアリングで出された質問は、「いったい君たちは新しい計算機を作りたいのか、それとも物理の研究をしたいのか」ということであった。しかしこの質問は一種の落とし穴(ジレンマ)である。もし「物理の研究をしたい」と言えば、そんならコンピュータを自作することなど考えずに買ってあげればいい、ということになるし、もし「計算機を作りたい」といえば、計算機屋のおもちゃに金は出せない、ということになる。86 年のヒアリングでは、筆者が「両者が分かちがたく結びついているところにこの提案の独自性があります。」と述べたところ、審査委員の N 先生がどういわけか激怒され、「君たちを査問しているわけではない」などと怒鳴り出し、当惑したことを覚えている。筑波大学の計算科学の基本路線は当時も今も Application, Algorithm, Architecture の codesign であり、筆者の発言は単なる言い逃れではない。ちなみに、2009 年 11 月 13 日の事業仕分けで日本の次世代スーパーコンピュータ計画が死にかけたが、その背景にもこのジレンマがあるのではないかと思われる。

87 年には、物理に重点を置き、代表者を星野氏から岩崎氏に変えて、4 度目の正直で提案し、めでたく採択となった。与えられた予算、時間、人的資源を考慮して、過去の PAX の研究成果を基に、目的とする QCD モデ

ルのシミュレーションに最適な QCDPAX の基本設計を行った。更に、その基本設計に基づき、並列計算を分担する演算要素(PU)のアーキテクチャ、ホスト計算機とPUとのインターフェイス、ホスト計算機システム、ソフトウェア(コンパイラ psc, アセンブラ qfa)の設計を行った。アンリツ株式会社の協力を得て、87年には、4台のPUからなるQCDPAX-4を製作し、88年度には、約半分のサイズの288台からなるQCDPAX-288を製作し、最終の89年度には432台のフルサイズで完成させた。

CPUはM68020、FPUとしては当時最高速(単精度)のLSIロジック社のDSPであるL64133を用い、疑似ベクトル演算や基本関数のためのゲートアレーチップFPUCを自作した。L64133はスカラ演算チップであるが、データをCPUと独立にメモリからFPUに供給するパイプラインの疑似ベクトル機構をFPUCに組み込んだ。また、FPUが加減乗しかサポートしないので、除算のための逆数、平方根逆数、三角関数、対数・指数関数、それにM系列逆数のためのハードウェアをも組み込んだ。パイプライン機構の設計は白川氏、関数関係のアルゴリズムやアーキテクチャの設計は筆者が行った。FPUCの機械命令は一種の水平マイクロコードであり、独自のプログラムカウンタとデコーダを持つ。

3. QCDPAX の苦労

しかし物理学の研究の道具として使えるコンピュータ、すなわち1日24時間、週7日間走り続けるコンピュータを製作することは容易ではなかった。まず、FPUCの設計および基板の設計がある。いずれもアンリツの技術者と協力して行ったが、いろいろ不手際があった。予算の関係で作り直すことができなかったので、一発勝負で製作した。FPUCでは、疑似ベクトル機構のカウンタ回路に誤りがあり、この部分は使えなくなった。このため疑似ベクトル演算のステップ数が増えることになった。また、基本関数回路では、指数関数計算のために指数部と仮数部を合成する回路を忘れ、他の演算で補った。また、プリント基板の設計ミスは、ジャンパー線で配線直した。並列計算機では、1つのミスを修正するためにすべての基板を補修することになり、その手間は膨大であった。また、当然のことであるが、1枚の基板

を動かしているうちは見つからなかったバグが、200枚動かすと200倍の頻度で見つかるのにも苦労した。

何より手こずったのは、CPUの25MHz、FPUの16.7MHzというクロックが当時としては高速でわれわれの手に負えなかったことである。しかも、この2種のクロックが基板上に共存し、さらに通信のために隣りのボードの配線も入り込んでいて、いくつものクロックが共存し、雑音やクロストークに弱い構造だったことがある。あるとき、疑似ベクトル演算とスカラ演算とで結果が異なるエラーが起こった。筆者が両者を分析し、違いがビット16にあることを心眼で見出した。配線図を調べたところ、ビット16の配線だけが遠回りをして雑音を拾っていることを見出した。これもジャンパー線で配線し直した。

ソフトウェアでも苦労があった。cライクな言語 pscのコンパイラは筑波大学とアンリツとで自作した。特に疑似ベクトル演算は vfor 文で指定することにした。しかし生成されたオブジェクトコードの性能は十分ではなかった。このため、一番重要な演算部分は FPUCのアセンブラ qfa で書く必要があった。これはロード、演算、ストアのパイプラインを自分でスケジュールする言語であり、手で書くにはアーキテクチャを熟知していなければならなかった。

これらハード・ソフトの問題点の解決のために、1989年度にハードは完成したが、本格的に稼働するにはさらに約1年を要した。完成後10年間も物理学の研究に用いられた。

4. 当時の日本や世界の情勢

1980年代後半、日本では HITAC S820 (1987)、FACOM VP2600 (1989)、NEC SX-3 (1990)などの第2世代のベクトルスーパーコンピュータが登場していた。アメリカでも ETA-10 (1987)、Cray YMP (1988)などのベクトル機が登場したが、同時に数多くの汎用的な並列計算機がさまざまな企業から登場した時期でもある。日本では、第5世代プロジェクトやスーパーコン大プロで並列計算機が試作されたが、位置づけとしては特定目的のための専用計算機であった。実用のための並列計算機としては、QCDPAXのほか

東大の Grape-1 (1989)など数えるほどしかない。

QCDのために専用の計算機を物理学者が自分で建設するという動きは、この頃アメリカやヨーロッパにもあった。アメリカではコロンビア大学を中心に開発されている。QCDPAXを製作している頃3号機が製作中で、256台のノードから成り、ピーク速度は16GFlopsである。IBMのワトソン研究所では、GF11という大きなSIMDが製作されている。

イタリアではApeというプロジェクトがローマ大学を中心に進められており、1988年には16台のノードで1GFlopsで稼働している。その後Ape100という100GFlopsのマシンが完成した。エジンバラ大学にもTransputerを用いた計画があった。

これらは、QCDPAXと同程度か小規模な計画であり、コンパイラを作らなかったために応用のソフトを開発する生産性があがっていないものがある。QCDPAXが曲がりなりにもコンパイラに力を入れたことは先駆的であったと思われる。

5. CP-PACS へ

これらの業績を基礎として、岩崎氏は次の計画への動きに入った。計画が採択される見通しの立った1991年初頭に、内外のコンピュータベンダ14社に手紙を送り、この計画に参画する意思があるかを問い合わせた。この手紙にはこう書かれている。

わたくしどもは、QCDPAXの成功に勇気づけられ、その経験を活かしてさらに数十倍の計算速度をもつ並列計算機を製作する計画を進めております。別紙に、現在考えております目標の概略を示します。これは決して生易しい計画ではありませんが、半導体技術の発展により実現可能の領域に入ってきたと考えられます。幸い、文部省でもわたくしどもの計画を推進しようという機運が生まれつつあり、諸般の事情が許せば平成4年度より5年計画で予算化される可能性が高くなって参りました。予算化された場合は、筑波大学に「計算物理

学研究センター」(仮称)を設置し、そこで計算機の製作と物理学の研究を推進することになるものと思われます。

1992年には物理学の研究に適した超並列計算機 CP-PACS の開発プロジェクトが「学術の新たな展開のためのプログラム」の研究課題「専用並列計算機による場の物理の研究」(予算総額 22 億円)として認められ、同時に筑波大学計算物理学研究センターが設置された。センターでは共同開発の相手として日立製作所と契約を交わした。

筆者はこの前年筑波大学を離れたが、月 1 回開かれたセンターの研究員会議にはできるだけ出席して、開発の議論に参加した。

日立の最初の提案は、4 個の HP 社の CPU をバスで共有メモリと結合したノードを、ハーバークロスバーで相互接続した超並列マシンであった。QCD のテストプログラムで詳しく検討したところ、バスネックでノードの実効性能が全然出ないことがわかった。これを解決したのが中澤研究室から提案された、浮動小数レジスタのリネーミングによってパイプライン処理を行う、疑似ベクトル処理のアイデアであった。

1992 年から 5 年をかけて CP-PACS が開発・製作され、同機は 1996 年 11 月の完成時点には 614Gflops を達成し、世界のスーパーコンピュータ トップ 500 リストの第一位と占めたことはご存じのとおりである。

CP-PACS は完成後 8 年にわたって稼動し、素粒子の強い相互作用の基礎理論である QCD に基づく素粒子と基本粒子クォークの性質の解明、量子力学第一原理に基づく固体水素の相構造の解明、宇宙輻射と物質の相互作用を取り入れた宇宙の階層構造形成のシミュレーション等、基礎物理学諸分野において世界的な業績を挙げた。

他方、日立製作所は CP-PACS で開発した技術で商用機 SR2201 を製作し、東京大学大型計算機センターなどに納入した。CP-PACS だけでなく、富士通が開発した NWT の技術が商用機 VPP-500 として実り、NEC が開発した地球シミュレータの技術が商用機 SX-6 として実っている。このような世界一級のコンピュータが応用分野からのイニシアチブで開発されたことは、日本の開発モデルとして注目される。現在開発中の次世代スーパー

コンピュータも、その技術が商用機として広く国内・国外で活用されることを期待している。

CP-PACS プロジェクトに携わって ～プロセッサアーキテクチャについて～

河辺 峻

(株)日立製作所汎用コンピュータ事業部 RISC 開発部(当時)
明星大学情報学部(現在)

1. はじめに

CP-PACS プロジェクトに携わった方から多くの原稿が寄せられていますので、私は関係者の一人として、プロセッサアーキテクチャ周辺についての思い出を述べたいと思います。厳密に言いますとプロセッサアーキテクチャという定義はないのかも知れませんが、ここではコンピュータアーキテクチャの中で主にプロセッサに関する部分を指しています。思い出話なので、記憶違いなども多々あるかも知れませんが、その点はお許し願います。

2. Bipolar ベクトルプロセッサから CMOS マイクロプロセッサへ

CP-PACS プロジェクトが始まった当初、私は日立のスーパーコンの3機種目にあたるベクトルプロセッサ S-3800 の仕上げで悪戦苦闘しておりました。スーパーコンとして日立はベクトルプロセッサ S-810 を 1983 年に出荷し、その後継機種 S-820 を 1987 年に出荷しました。その次の機種が S-3800 で、1992 年の年末から 1993 年の年始にかけては大晦日も元旦も無い状況でした。これらはいずれも Bipolar を使用した高速マシンでした。S-3800 はより高い周波数を狙ったメモリ共有型のベクトルプロセッサでしたが、チップから発生する熱(勿論水冷でしたが)やチップの生産歩留まりなどの問題が顕在化し、Bipolar の限界を感じていました。

従いまして、次のスーパーコンの候補としては、CMOS でメモリ分散型のいわゆる並列プロセッサを漠然とイメージしていました。プロセッサアーキテクチャとしては、前記のベクトルプロセッサはメインフレームにベクトル命令を追加したものでした(国産3社のスーパーコンは皆このアプローチでした)。これは従来システムとの互換性や OS の開発工数などによるものでし

たが、時代は次第に UNIX の流れになってきており、プロセッサアーキテクチャも変えても良い時期になっていました。

そこでどういうプロセッサアーキテクチャが良いのか考えようとしたのですが、これはあつけなく HP 社の PA-RISC に決まりました。当時の日立の WS 部門は HP 社と提携関係にあり、日立独自で PA-RISC の CMOS マイクロプロセッサを開発していました(HARP-1 という開発ネーム)。この開発がなかなか収束しない状況でした。ある日上司に呼ばれて、「S-3800 の開発のメドが立ったらその設計者をまわしなさい。次のスーパーコンとして CMOS の並列プロセッサを考えているのならちょうどよい、RISC 開発部というのを創るからその責任者になって両方やりなさい。」重要な決定は意外と簡単になされるものです。

それで HARP-1 をプロセッサに用いた並列プロセッサ SR2001 の開発が始まったのです。

これは日立としては初めての並列プロセッサで、後の CP-PACS のプロトタイプというべき重要な開発で、HyperCrossbar や OS のマイクロカーネルなどはここで検証されました。

ただ PA-RISC はアドレッシングが 32 ビットでしたので、スーパーコンとしては将来どこかで 64 ビットに拡張することになります。そんな時に HP 研究所で 64 ビットアーキテクチャの開発をやっており、日立も一緒にやらないかという話が来ました。この開発の責任者は William Worley さんという方で、RISC の元祖である IBM の John Cock さんと IBM801 を開発した後、HP 社に移り PA-RISC を開発した方でした。話を聴くと非常に温厚な方で、新しく開発している 64 ビットアーキテクチャは VLIW(Very Long Instruction Word)をベースにしたもので PA-WW という開発ネームで呼ばれていました。将来の 64 ビットアーキテクチャとしてはとても魅力的な内容であり、当時の開発日程では CP-PACS の開発日程にも間に合うように計画されていたので、私としても大変乗り気になって中央研究所の方や S-3800 の設計者を HP 研究所に派遣して共同プロジェクトに参加しました。

しかしこの計画は2つの理由で頓挫しました。1つは HP 社としては PA-RISC の 64 ビット化を内部で進めており、PA-WW を急いで開発する必

要が無くなったことにより、開発計画がずるずる延びて CP-PACS に間に合わなくなったこと。もう1つは Intel 社が IA-32 の後継アーキテクチャとして HP 社と組んで共同開発を始めたことでした。このアーキテクチャは現在では IA-64 と呼ばれ、プロセッサは Itanium シリーズとして Intel 社から製品化されています。

当時日立が独自に開発できるマイクロプロセッサは、PA-RISC では 32 ビットのみでしたが、並列プロセッサですので、1つのプロセッサのメモリ容量が 4GB を超えるのは当分先とみて、このプロセッサアーキテクチャをベースに新しいマイクロプロセッサを開発し、それを CP-PACS のプロセッサにすることにしました。これが HARP-1E という開発ネームのプロジェクトでした。

3. CP-PACS プロジェクト

このプロジェクトで開発された技術は、その後並列プロセッサ SR2201 として製品化されました。この中のプロセッサ関連で想い出に残る技術として次の2つがあります。

(1) スライド・ウインドウ(Slide Window)

科学技術計算の性能に関しては、特にメモリからのデータプリフェッチ技術が重要で、ベクトル命令によるベクトル処理は、その代表的な技術でした。しかし当時のマイクロプロセッサは、数百 KB の大容量のベクトルレジスタを実装する余地はとてありませんでした。

そのような状況の中で、レジスタを 32 個から 128 個に拡張してデータプリフェッチを行うスライド・ウインドウ方式のアイデアを初めて聞いたとき、うまいやり方だなあと思いました。

アーキテクチャ上、演算は 32 個のレジスタしか指定できませんが、ロード・ストア命令だけ特別に直接 128 個のレジスタを指定できるようにしておき、32 個単位のレジスタを1つの窓にして、演算時はどの窓を使用するかを指定するというやり方です。1チップマイクロプロセッサでしたので、メモリを比較的近い場所に置く事ができ、128 個のレジスタがあれば CP-PACS のアプリケーションでは十分に性能が発揮できるというのです。

そこで1チップの中のレジスタを 128 個に拡張することさえできれば、制御部分は比較的単純にすみそうなので、これはうまいやり方だと思ったのです。

科学技術計算の性能とアーキテクチャに関する短絡的な議論の1つに、「ベクトルかスカラか」というのがあります。CP-PACS の例でも判るように、むしろどのようなデータプリフェッチ技術を採用して、それとアプリケーションとの整合性をまず議論すべきと考えます。CP-PACS の時代と比較して、現在は演算器とメモリとの距離が非常に離れていますので、例えば大容量のキャッシュを導入した場合のスライド・ウインドウ方式などは興味あるテーマの1つでしょう。

(2) ハイパークロスバー(HyperCrossbar)

並列プロセッサを接続するネットワークは、いろいろな方式が考えられています。使用する立場で考えるとノンブロッキングクロスバー方式が一番良いと思います。しかしながらハードウェアの点から見ると、2000 台以上のプロセッサをこの方式で実現しようとするのは無理があります。

CP-PACS で実現した3次元クロスバー方式は、X軸が8、Y軸が16、Z軸が16の $8 \times 16 \times 16 = 2048$ 台のプロセッサを接続する方式でした。それぞれの軸に関してはノンブロッキングクロスバー方式になっていて、軸を渡る時だけブロッキングが発生する可能性があるというものです。

何よりも感心したのは、X軸はプロセッサボード上にあり、Y軸はバックボード上にあるため、 $8 \times 16 = 128$ 台までの並列プロセッサ構成では、接続するネットワークのケーブルがまったく必要無いということでした。CP-PACS は 2048 台のプロセッサ構成ですのでZ軸を構成するためのケーブルが山のようにありますが、製品化された SR2201 ではネットワークケーブルをほとんど必要としませんでした。

CP-PACS で実現したこの2つの技術は、アプリケーションと当時のハードウェア技術が見事に調和してできたもので、このプロジェクトの醍醐味の1つであると言っても過言ではないと思います。

4. おわりに

1996年のSC96はPittsburghで行われました。そこでCP-PACSがTop500のNo.1になったことが発表されました。その場に居合せた私たちは、その晩Pittsburghの最も高いビル(確か64階)の最上階のレストランでささやかな祝杯を挙げました。出席者は、筑波大側からは小柳先生、朴先生それに当時東大にいらっしやった東工大の松岡先生など、日立側からは澤本さんや菊池さんなどでした。それが下の写真です。

2006年より2年間、私は文部科学省の兼職(技術参与)となり、次世代スーパーコンピュータに参画することになりました。CP-PACSは、過去日本で開発したスーパーコンピュータで世界をリードした成功3プロジェクト(他の2つは、数値風洞(NWT)と地球シミュレータ(ES))の1つとして位置付けられており、プロジェクトに携わった一人として大変嬉しく思います。

次世代スーパーコンも同様な期待がかけられています。CP-PACSプロジェクトのように、最先端のアプリケーションとハードウェアおよびソフトウェアが調和してできる技術が最も重要なものと思っております。



CP-PACS システムの開発を省みて

和田 英夫

(株)日立製作所エンタープライズサーバ事業部
RISC 開発部(当時)
第1サーバ本部第3部(現在)

2005年9月にCP-PACSが約10年間にわたる稼働を完了しました。私は、初期検討から稼働後の事故対策まで、ハードウェア設計者として、CP-PACSを担当させていただきました。至らない技術者でしたが、筑波大の先生方は、よく付き合っていたいただいたと思うとともに、10年もの長い間CP-PACSを使っていたいただいたことに感謝いたします。

2005年8月に、開通したばかりの、つくばエクスプレスに乗ってみようということで、数年ぶりにつくばを訪れました。地下の改札口から外に出てみると、そこは、見慣れたつくばバスセンターでした。「ここがつくば駅になったのか」と初めて知りました。駅周辺もずいぶん様変わりして、華やかな街になったように思えました。私は、CP-PACSの件で、つくばに5年くらい定期的に通っていたので、郷愁に似た感覚を覚えました。

さて、その懐かしいCP-PACSの開発ですが、以下、年度毎に、私が経験したこと、思ったこと等、意の趣くまに記します。

(1) 平成4年度

最初にぶつかった難問は、長大データを処理するときに起きる、キャッシュ溢れによる、実効性能低下でした。CP-PACSは、従来のようなベクトル型スーパーコンピュータでなく、汎用RISCチップを並べた並列型スーパーコンピュータであり、汎用RISCチップは、キャッシュにデータがあるときに高性能が出るような構造になっているので、この問題が発生したわけです。

ここで、考案された機能が、かの名高い「擬似ベクトル処理機構」です。この機構によって、後続命令を止めることなく、メモリーからデータをパイプ

ライン的に取り込むことにより、ベクトル型スーパーコンピュータと同様に、メモリー上の大規模データを演算器へ高速に供給します。

また、PU(Processing Unit)間の通信の高速化が問題となり、Remote DMA 転送が考案されました。Remote DMA 転送では、送信側/受信側ともにユーザの仮想アドレス空間の一部をリアルメモリー上に固定的にマッピングしておき、それらのメモリー間でデータ転送を行いません。これにより、異なるPU上のユーザ仮想アドレス空間の間で直接データ転送ができます。カーネル空間とユーザ空間の間でデータコピーが発生しないため、高速な転送ができます。

さらに、PU 間ネットワークとしては、「3次元ハイパークロスバネットワーク」を採用しました。「3次元ハイパークロスバネットワーク」は、以下の構造を有します。

1. PU を、x 方向、y 方向、z 方向に、 n_1 個、 n_2 個、 n_3 個、直方体状に並べる。(すなわち、総 PU 数= $n_1 \times n_2 \times n_3$)
2. x 方向、y 方向、z 方向に並んだ各列の PU を各列ごとに完全クロスバで結ぶ。
3. 各 PU はスイッチ(router)を持ち、転送方向を変える(たとえば、x 方向 → y 方向)ことができる。

「3次元ハイパークロスバネットワーク」の特長は、以下の通りです。

- a. 短距離通信:最大3回のクロスバスイッチ乗り換えで任意のPUと通信可能。
- b. 柔軟なトポロジ:演算プロセスのPUへのマッピングに対する自由度が高い。

(2) 平成5年度

CP-PACS の構造も固まりつつあり、QCD 計算の性能評価を行なうために、メモリーシミュレータを作成しました。このシミュレータは、筑波大の先生方に、重宝がられたようで、光栄でした。

次に、大問題が起きました。2nd cache のアクセスに関する制御の問題

で、プリロード命令(メモリー先読み命令)が、1サイクルピッチで実行できないことがわかりました。どうやって、この問題を解決するか、長時間かつ激しい議論が続きました。ユーザデータをアンキャッシュャブル領域(キャッシュに格納しないメモリー領域)に置こうか、2nd cache を使用しないモードを作ろうか、等々、種々の案が出されましたが、結局、ストアイン方式をストアスルー方式に変更することによって、解決しました。

また、ディスクに関して、信頼性の問題から、通常のディスクを並べるのではなく、RAID(Redundant Array of Independent Disks)の採用が決まりました。

(3) 平成6年度

メモリー素子の件で問題が起きました。当初、SDRAM(Synchronous DRAM)の採用を考えていましたが、供給が困難になりました。そこで、通常のDRAMに変更になりました。SDRAMは現在では、多くのマシンで使われ、popular になりましたが、この時点では、時期早尚だったようです。SDRAMにチャレンジしたかったのですが、残念に思いました。

次に、CP-PACSとFCS(センタフロントシステム:ユーザのCP-PACS使用を支援するシステム)の間の実効性能が問題になりました。CP-PACSとFCSはピーク転送速度100MB/sのHIPPIで接続しますが、通常のプロトコルで転送すると実効数MB/sの転送性能しか出ません。これでは、大量のデータを扱う、CP-PACSユーザには不足です。筑波大側から、「実効50MB/s」という要求が出ました。これに対し、メモリー間転送の回数を減らしたHFTPという転送プログラムを開発しました。システム完成後、平成8年に、実際に実機で実効66MB/sの性能が出ることを確認しました。

また、ネットワークレイテンシ、メモリーレイテンシといった基本性能諸元や、バリア同期、ブロードキャスト、分割運転、ブロックストライド転送といったPU間ネットワーク転送の詳細仕様が固まったのもこのころです。

実装構造が固まったのも、このころです。最大1GBのメモリーをもったプロセッサ(0.3GFLOPS)を約15cm×20cmの面積に凝縮し、45.6cm x 62.5cmのパッケージ当たり8台のプロセッサを搭載するという、世界最高ク

ラスの実装密度を実現しました。

(4) 平成7年度

CP-PACS の製造/評価が行なわれました。毎 WG(ワーキンググループ) で、どの工程まで進んだかを報告しました。

結局、設備は、センターの用意した設備の容量内で収まりました。

平成8年3月に、1024PU 機の検収も無事に終わり、達成感を味わいました。

(5) 平成8年度

システムが稼働開始しました。いくつか問題も起き、また、いくつか喜ばしいこともありました。

まず、電源ノイズにより誤動作が発生しました。これは、PU 間ネットワークのクロスバ部の電流の変動が、予想以上に大きかったためでした。我々は、Bi-polar のマシンの経験はありましたが、CMOS の並列機は、CP-PACS が最初ということもあって、電流変動量を読み誤りました。Bi-polar は論理回路の動作率にかかわらず、ほぼ同じ電流が流れますが、CMOS の電流量は論理回路の動作率に正の相関があります。CP-PACS は並列機であり、そのためにクロスバの部分が極端な論理回路の動作率の変動を起こすということを予想していませんでした。この点については、平成8年9月の2048PU 化の時に対策させていただきました。

この2048PU 化によって、CP-PACS は、ピーク性能 614.4GFLOPS の、世界に冠たるスーパーコンピュータシステムになりました。

さらに、「CP-PACS が1996年11月のTOP500で1位をとる」という偉業を成し遂げました。このLinpackの測定は、1996年9月下旬に行なわれました。朝から測定を開始したのですが、OS パニックは起きる、ディスクはパンクする、等々で、結局、測定値が得られたのは、翌日の明け方でした。マシン室で「世界記録達成！」と仲間で喜びをわかちあったのを記憶しています。このときの記録(Rmax)が 368.2GFLOPS でした。今は500位以内に入るのに、Rmax 20.05TFLOPS が必要です。技術の進歩の速さを感じま

す。

しかし、2048PU 化以降、インターミットに、原因不明のネットワークエラーが発生し、悩まされました。「再現性なし。発生位置も、発生プログラムも一定しない。」ということで、約半年、原因究明に苦しみました。結局、ケーブルの製造不良（接触不良）が原因でした。この不良が解決して、CP-PACS は安定稼働に入りました。

この CP-PACS で培った技術（失敗も含めて）が、日立のその後のスーパーコンピュータの開発に生きています。

私は、最後の WG で、中澤先生から「いろいろあったけれど、全体としてみれば、よくやった。」というお言葉を聞き、胸が熱くなりました。

CP-PACS との出会いと感動

荻山 得哉

(株) 日立製作所 公共情報営業本部筑波営業所(当時)

(株) 日立製作所 東北支社(現在)

平成7年2月、私と CP-PACS との出会いはそれまで設計段階で現場を率いた筑波営業所長 能沢健 を引き継いだ時から始まりました。

この頃の日立は、次期スーパーコンピュータの開発をそれまでのベクトルスーパーコンアーキテクチャで踏襲するのではなく、別の解を模索中であり、大きな転換を必要とする時期に来ていました。更に言及すれば、何らかの新技术でブレイクスルー出来なければ日立のスーパーコンピュータ事業はこれで終わりか、との危機感を(個人的にですが)抱く程の状況でした。そんな時、能沢から『このプロジェクトはノーベル賞に値する研究のために必要な、新しいスーパーコンを作るんだぞ』と言われて引き継いだものですから、それこそ「日立にとって良いチャンス」と発想したのは当然ですが、この「ノーベル賞に値する」の言葉に、一会社人を超えた深い社会的使命感と意義を鮮烈に感じた印象が今も残っています。

正直なところスーパーコンピュータのビジネス自体は利益を生み出す事業にはなかなか成りえません。従って「日立」という会社で事業の成長・拡大をミッションとする企業マンとしては立場上忸怩たる思いを隠し得ないのですが、この分野に携わり『何のために仕事をやっているのか』と問われたら、次のことを堂々と言いたいと常々思っています。『科学技術計算の基盤であるスーパーコンをしっかりと開発し、研究者の先生方にご提供することで、先生方が大きな研究成果を生み出し、その成果が人間社会に役立つことを通して我々は社会に貢献しているんだ』と(真に勝手な三段論法的解釈ですが)。この意味で、微力ながらも CP-PACS プロジェクトに参画させて戴いた年月は、私にとって社会貢献の意義を大きく見出せた体験であり、また大変貴重で幸せなものとして今現在も心の中に生きています。

開発途上のワーキングディスカッションは大変辛いものでした。先生方は未来に視点を置き、大きな目標に向けての隘路を何とか打開していこう、との理想的なご要求が続く。我々日立は現実解を求めて(かつ開発経費とのバランスも視野に入れて)解決策を見出そうとする。従ってその目標に向かうプロセスの違いや認識の違いの温度差が大きければ大きいほど会議は紛糾し、結果、先生方の妥協を許さぬ意気込みの強靱さに圧倒され、日立の設計者達は痛く落ち込んでセンターを去る、の連続でした。

そこで、毎夜毎晩その設計者達と営業・SEで酒を飲み、先の社会的意義や目標達成の際の喜びを夢に描きながら、お互いに「頑張れ！頑張れ！」と励ましあったものです。

しかしながら、時には、『世界一速いコンピュータを造るのが俺のライフワーク』と言っていた設計者の一人が、その(予算面も含めた)プロジェクトマネジメントの厳しさから『やってられない！』とさじを投げ、誰からとも無く『若い頃の情熱はどこに置いてきたんだ！』と檄が飛び、酒の勢いで昼間と同様に夜も紛糾してしまう、といった場面もありました。

身内の恥を晒すようでお恥ずかしい限りですが、先生方の強い信念と高い目標からくるご要求に、皆良く耐え(?)抜いたものだと思います。結局、そんな彼らを支えていたのは、日立の技術力を信じる気持ちであり、しっかり動くまでは絶対に逃げ出さない、という日立伝統の開拓者精神にあったかと思います。

さて当時、科学技術計算に用いられる汎用のスーパーコンピュータは日立の持つ S-3000 シリーズで 1GFLOPS 当たり数億円しました。その数字から計算すると、CP-PACS を開発するには少なくとも 1 千億円以上必要となります。これに対して、開発予算は十数億円だった訳ですから、実質 1 千億円以上の差があったこととなります。

このギャップをどう埋めていくのが開発当初の大きな課題だった訳ですが、この点は低価格、省電力の RISC チップを採用することでかなり解消されました。

ところが、RISC チップでは性能が出ない。

そこで最初にぶつかったのはメモリバンド幅の問題です。この部分はコストさえ掛ければどうにでもなる代わりに、ハード工場での開発費の捻出は並大抵の努力ではなし得ないものでした。

次はパイプライン処理の問題です。RISC チップにはこれまでのベクトルマシンと違いこの機能が無く、当然これを盛り込もうとするとチップ自体が独自開発同然となり、結局また膨大な開発費を必要とします。

そこで、先生方のご努力によりソフトウェアパイプラインニングというアイデアが生まれました。ところが、このアイデアに対して今度はソフト工場が消極的になってしまい、また実現に向かうには四苦八苦することになります。

しかし、新しい技術は将来必ず生きるはず、また活かさねば、と信じ、とにかく新技術を残したいその一心が社内の一体感を生みました。先生方のご指導の末、ついに「擬似ベクトル機構」が誕生し、RISC チップでも最小限の機能拡張でベクトル機並みの性能を出すことができる画期的な技術となりました。

この擬似ベクトル機構は、後に CP-PACS の思想をベースに製品化された SR2201 や、次の SR8000 というマシンにも引き継がれ大変なヒット商品となりました。

あるユーザの方は SR2201 を評してこう仰いました。

『SR2201 は超並列コンピュータ界の F1 マシンだ！』『名機！SR2201』

こうして、CP-PACS の設計図は目を追って確実に書きこまれていった訳です。

平成8年3月、ついに第一段階（目標の半数である 1024PE）の製作が完了し、筑波大学へ納入されました。

計算物理学研究センターのマシン室へ搬入される CP-PACS を見て、その後の苦労を予期するかのように「大変なのはこれからだ」という言葉を、心の中で呪文のように繰り返していたことを思い出します。

実際、納入されたばかりのマシンはハードウェア、ソフトウェアの両面で障害が多発しました。特にネットワークマージンの現地調整は困難を極め、ハード工場の技術者、QA(品質検査)、研究所のサポート部隊、ありとあらゆるメンバーが大学に集結し必死の対応が毎日、毎夜続けられました。

一方、ソフトウェアの面では超並列アーキテクチャを制御するための新しいOSであるHI-UX/MPPと、その上で動作するFORTRANコンパイラが頻繁に障害を起こしていました。ソフトウェアの問題はハードウェア以上に深刻で、とりわけスローダウンの障害では事の外、先生方にご迷惑をお掛けしたと、今もって大変恐縮しています。

確か宇川先生のジョブだったのでしょうか。流し始めてから計算が完了するまでに約8時間を要するもので、夜から朝に掛けて流されるのですが、あるところまで計算が進むと決まってシステムが停まってしまう。

こうなると、真っ先に佐久間、伊藤そして私の電話が鳴りました。

『システムがダウンしました。』と現地で対応している保守員からです。かかってくる時間は毎晩同じで、その度に眠い目をこすりながらも飛び起きて現場に急行しました。

現場では続々と関係者が集まり障害原因の究明が進められます。まるで工場出荷前に実施するヒートランテストのような状態でも、毎晩あきらめずにジョブを掛け続けた先生方の執念は頭の下がる思いで、本当に心苦しい毎日でした。ご迷惑ばかりお掛けし成果が出ない。しかしあの頃、間違いなく誰もが「必ず動かしてみせる」という共通の目標に向かって対策にあたっていたと信じています。

その典型的なエピソードの一つですが、夜中いつものように電話が掛かって来て現場に向かうと、土浦市内に住んでいる私が駆けつけるより早く、神奈川工場の設計者がいたことがありました。私が驚いていると『勘が働いた』と本人は笑っていましたが、後で聞いたら、何と大学の近所に自費でアパートまで借りて対応していたようです。

もう一つ印象的なのは、昼間は毎日といって良い程頻繁に対策・報告

会議を開くので、社内会議をそのまま先生方の前に持っていく状態だったことです。とにかく社内レビューすらできない状態で大学に出向くものですから、ハード工場とソフト工場が喧嘩を始めてしまうことも多々ありました。先生方からは“セクショナリズム”とお叱りを戴きましたが、双方とも少しでも良い物を作って残そうという情熱がそうさせていたと思います。お蔭様でこの議論は先生方も許して下さり、どんなに夜遅くなろうともとことん話し合い、見解を出したものです。

会議終了後は土浦の駅前にある中華料理屋で反省会をしながら食事をするので、この店は筑波営業所第二会議室と言われていました。本当に心苦しく辛い毎日でしたが、今となっては楽しい思い出でもあります。

その後次第に CP-PACS は何とか思うように動き出し、着々と成果が出始めました。

平成8年9月、残りの半分が納入され CP-PACS は総数 2048 個の CPU を搭載する超並列コンピュータとしてついに完成し、いよいよ最高の性能を世に知らしめるデビューの瞬間がやってきました。

LINPACK 試験を開始。理論ピーク性能 614GFLOPS に対して LINPACK 実行性能 368.2GFLOPS。実に実行効率 60% の CP-PACS プロジェクトが名実共に世界 1 位となった瞬間でした。

私は今でもこの時の感激を鮮明に覚えています。そしてこの一連の、関係者全員の努力が実を結び成し遂げたこの偉業を、若い人たちに伝えていく事が私の使命だと思っています。

技術は日進月歩です。一瞬でも留まればそれは即ち後退を意味します。日立が社会に、科学技術に対する使命を果たし続ける為にも、CP-PACS プロジェクトを通して学んだ技術へのこだわりや、お客様から喜ばれることの価値を次の世代へと繋げていき、また、お客様である先生方とこれだけ喧々諤々議論を交わしながらも、ひとつの目標に向け一体となり、信頼の絆で事を成し遂げさせて戴いたプロジェクトは過去にもこれからも無いので

はないかと、誇りを持って言い続けたいと思います。

『 感動を与えてくれた CP-PACS よ、ありがとう！ 』

最後になりましたが日立の実力を心底ご信頼戴き、また悪いところも十分ご理解戴いた上で、決してあきらめない強い信念と情熱で、様々なご教示、ご先導を賜りました先生方、CP-PACS をこよなく愛していらっしゃる先生方に心より敬意と御礼を申し上げ、拙文を閉じさせて戴きます。

平成22年4月14日

CP-PACS 稼働終了 ～その思い出と最後に立ち会って～

浅野 朋広

(株) 日立製作所 公共システム営業統括本部
筑波情報システム営業所(当時)
北海道支社(現在)

平成 17(2005)年 11 月 21 日.

後に平成 18 年豪雪と名付けられるほどの記録的大雪を経験することなど、誰もがまだ想像すらしていなかった冬の始まりに、CP-PACS は約 10 年を過ごした筑波大学計算科学研究センターをあとにしました。

一口に 10 年と言えばあっけない気がしますが、使用され続けたコンピュータ、しかも科学技術計算のコンピュータとしての寿命を考えると、途方も無い長寿のマシンであったと断言できると思います。

CP-PACS プロジェクトは平成 4 (1992) 年に始まったと伺っておりますので、設計段階も含めると実に 13 年以上に渡るものになりました。

当然、このプロジェクトに携わった日立の営業は何人もいますが、気が付けば私が一番長い付き合いになりました。平成 9(1997)年に担当になってから、10 年近くを共に過ごしました。

この 10 年来の盟友ともいえる CP-PACS が次々とトラックの荷台に積まれていくのを見ながら、私は初めて筑波に来た日のことを思い出していました。

私は平成 9 (1997) 年 6 月にここ筑波へ赴任して参りました。

赴任の翌日、いきなり CP-PACS の停電テストに引っ張り出され、それが CP-PACS との初対面になりました。右も左も分からぬまま走り回ってお手伝いしておりましたが、随分と活気のある現場だなあ、というのが最初の素直な印象です。

この停電テストは、納入されて約 1 年を経過した CP-PACS が停電時に計画されていたシャットダウンをしなかったため、その正しい動作を確認することと、停電による温度上昇などの影響を調査する目的で行なわれました。そのため、集まったメンバーは工場の設計や QA の人間だけでなく、日立の UPS 担当者や日立電子サービスの保守員まで、総勢 20 人を越えていたと記憶しています。

私にはセンター裏口のオートロックが正常に動くかどうかを確認する為の見張り役が与えられましたが、あとは 20 人分の弁当の調達と、当時の営業担当者だった宮脇の後をただついていだけでした。

さて、肝心のテストの方ですが、仮想的な停電テストとは言え、実際に停電が起きた時と全く同じ状況を作り出し試験をしました。ブレーカが落とされると、明かりは消え、コンピュータだけでなく空調機も全て停止します。普段は話し声もかき消すほどのうるさいマシン室が一瞬異様なほど静まり返り、秒刻みのカウントを繰り返す SE の伊藤の声だけが、マシン室の方からスピーカを通して淡々と響いてきました。そして数秒後には計画通りに動かない個所が次々と見付き、怒号や悲鳴にも似た声が飛び交いました。全員の表情には緊張感がみなぎっていました。さながら工場の設計現場をそのまま持ち込んだような喧騒と活気を目の当たりにし、昨日来たばかりの新参者の私は、圧倒されながらもどこかワクワクする感情を抑えられませんでした。

ここは面白い、と・・・

しかし、その数日後、今度は初めての運用連絡会議で最初のワクワク感はあるが吹っ飛びました。

それは半端な量ではない席上資料を見たときでした。今でもはっきりと思い出せます。あの頃の資料は間違いなく立ちました。それほど厚みがあったのですが、全部読むだけで一体どれほどの時間が掛かるのだろう、と本当に驚きました。実際、その頃の運用連絡会議は、資料を読み上げるだけでもかなりの時間を要しました。更に、まだネットワークマージンが安定していなかったため、それに関する議論だけでも 1 時間以上かかり、お

昼の1時頃から始まった会議は、大体夜の6時か7時くらいまで、ほとんど途中休憩も無しに続けられていました。

当時、私が赴任したばかりの頃にはワーキングは既に終了しており、筑波大と日立の間で行なわれる会議としてはこの運用連絡会議だけでしたので、それだけに内容が濃く、大変なものでした。

毎日がカルチャーショックの連続でした。しかし、そのお陰で色々なことを経験し、学ぶことが出来ました。

さて、話は変わりますが、今回のこの企画で、営業担当者としての寄稿をご依頼いただいた時、正直大変戸惑いました。何故なら、先述のとおり私は最も付き合いの長い営業担当者ではありますが、おそらく皆さんが一番ご苦労なさったであろう設計段階を知らないからです。きっと、私以上の適任者はたくさんいると思いました。しかし、CP-PACS プロジェクトの最後を取りまとめ、見送った者としての使命を果たさなければという思いで筆を執りました。ですから、CP-PACS の本当の最後を、これをお読みになる皆さんにお伝えできればと思います。

CP-PACS が撤去されていく日、現場に居合わせた私は9年という歳月を改めてかみしめました。

よく「十年一昔」と言いますが、確かにCP-PACS の性能だけ見てもそれは当たっています。CP-PACS は単体ピーク性能が300MFLOPS でした。今はパソコンでも数GFLOPS 出してしまうます。技術の進歩はすごいと思う典型的な事例です。

では、そんなにも時代遅れになってしまったCP-PACS は晩年誰にも使われなかったのかと言うと、全然そんなことはありませんでした。最後の最後まで稼働率は90%以上をマークしつづけ、驚くことに、シャットダウンの朝ぎりぎりまでジョブが積まれていたというのですから、このことからいかにこのマシンが多くの人に愛され、活躍したか想像に難くありません。

この事実は、CP-PACS プロジェクトに関わった全ての人が、どれほど素晴らしい仕事をしたのかという証明にもなったと言えるでしょう。

解体作業は手際のいいものでした。筐体が切り離され、間のネットワークケーブルは綺麗に切断されました。また、CP-PACS の心臓部であるボードのいくつかは記念品として何人かの人に分配されました。

こうしてバラバラにされた CP-PACS は、平成 17(2005)年 11 月 21 日のお昼頃には全てトラックに積み込まれ、計算科学研究センターをあとにしました。

さて、撤去された筐体の一部は、展示用に手を加えられ、照明やアクリルパネルのついた立派な展示品として生まれ変わりました。また、国立科学博物館にも寄贈され、設計仕様書と共につくば分館の所蔵庫に納められました。何年、何十年か後、誰かが CP-PACS に気付き、膨大なネットワークケーブルを見て驚きの声を上げるかもしれません。

マシン室は CP-PACS がなくなり一時は大変静かになりましたが、すぐあとに FIRST プロジェクトの新しいマシンが、さらにその数ヵ月後には PACS-CS プロジェクトのマシンが入り、あっという間に活気を取り戻しました。

時代がまた一つ入れ替わった瞬間でした。

これからも筑波大学計算科学研究センターでは、世界をリードする研究がなされ、発展していくことでしょう。いつか CP-PACS は、その長い歴史の中の小さな点に過ぎない存在になるかもしれません。しかし、CP-PACS は確かに存在し、大きな成果を残したことは決して消えることの無い事実として刻まれました。私はそんな瞬間に立ち会えたことを心から誇りに思っていますし、感謝しています。

末筆ながら、CP-PACS プロジェクトに関わった全ての方に敬意を表し、最後に立ち会う機会を与えて頂きましたことに心より御礼申し上げます。

超並列計算機 CP-PACS 年譜

1991年4月2日	メーカーに開発協力打診
1991年5月9日	学内検討第一回
1991年5月16日	学内検討第二回 CP-PACS の名称
1991年6月20日	Mult64 ⁴ 格子に対して実効性能 12.3%
1991年8月10日	暗黒の CP-PACS ミーティング
1991年8月29日	擬似ベクトル方式に関する中澤文書
1992年6月12日	日立と契約
1996年3月25日	1024PU+64IOU 設置
1996年4月8日	運用開始
1996年9月18日	2048PU+128IOU 設置
1996年9月27日	Linpack 368.2Gflops
1996年10月4日	運用開始
1997年7月28日～8月3日	クロックケーブル交換
1999年11月12日～14日	クロック位相変更作業 以後安定
2005年9月29日	稼動終了

稼動統計(1996年10月4日～2005年9月29日)

総稼動日数(パワーオン日数) 2813日

総稼働率(計算実行割合) 86%

CP-PACS 関係の主要な成果発表 1996年4月～2005年9月

素粒子物理学

学術雑誌論文 36 件

国際会議報告 94 件

Spires 引用総数 約 2500 100 回以上 2 件 50 回以上 16 件
(2005 年 9 月時点)

- クエンチ近似でのハドロン質量スペクトルの決定
- 初めての系統的なフレーバー数2の動的格子 QCD 計算
- 高温でのクォークグルオンプラズマの性質の研究
- domain-wall 法による K 中間子の弱崩壊の研究

宇宙物理学

学術雑誌論文 12 件

国際会議報告 29 件

- 宇宙輻射流体力学による宇宙再電離過程の計算
- HMCS による銀河形成の輻射流体力学計算

計算機工学

学術雑誌論文 32 件

国際会議報告 24 件

- スライドウィンドウ方式の高性能プロセッサアーキテクチャ及びコンパイラの開発
- ハイパクロスバ網の特性解析と各種並列処理への適用性の証明
- 異機種結合型高性能プラットフォームの開発と計算宇宙物理学への応用
- 入出力装置の多重性を生かした高性能入出力システムの開発と HMCS への応用

稼動終了後について

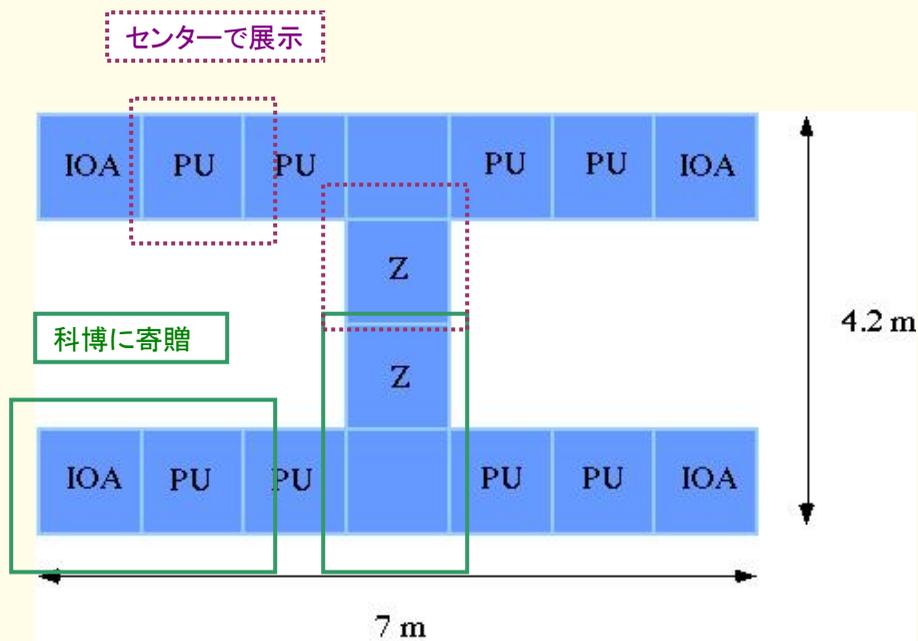
国立科学博物館に寄贈

- PU筐体 1
- IOU筐体 1
- Zクロスバ筐体 1
- 電源筐体 1
- 開発関係書類 一式

1. 1996年11月Top500リスト
2. 「筑波大学計算物理学研究センター10年の歩み」(平成14年7月)
3. CP-PACS仕様書等 約100冊
4. CP-PACS技術検討会資料等 約20冊
5. 情報処理学会誌 第37巻1号 特集:計算物理学と超並列計算機 - CP-PACS計画 -.
6. 「専用並列計算機による「場の物理」の研究」研究進捗状況報告書 (筑波大学計算物理学研究センター 平成6年8月)
7. 「専用並列計算機による「場の物理」の研究」研究成果報告書 (筑波大学計算物理学研究センター 平成9年8月)

計算科学研究センターにて展示

- PU筐体 1
- Zクロスバ筐体 1



CP-PACS フロアプラン

あとがき

CP-PACS は9年間の活躍の後に2005年9月29日に稼働を停止した。これに先立ち、2004年4月には、CP-PACS プロジェクトの舞台であった計算物理学研究センターは計算科学研究センターへと改組拡充され、その主力計算機システムは、2006年を境目にPACS-CS システムへ、さらにt2k-tsukuba システムへと移り変わった。さらに近年、計算科学とスーパーコンピュータを巡る日本と世界の趨勢は大きく動いている。

CP-PACS プロジェクトが構想されて、平成22年4月は丁度20年目を迎える。CP-PACS が歴史上の存在になりつつある今、関係者それぞれの視点から CP-PACS プロジェクトを回顧する文集を編纂することは意義深いことであろう。

実は、この編纂は、CP-PACS が稼働停止して半年後の2006年4月に既に企てられていたのであるが、様々の事情で滞っていた。今回、是非完成すべきだと後押しして下さった佐藤三久計算科学研究センター長に感謝する。また、短時日にも係わらず文章をお寄せ下さった多数のプロジェクトメンバーに謝意を表す。

最後に、表紙は金谷和至氏作成であることを記しご協力に感謝する。

宇川 彰

平成22年4月15日

ISSN 1344-3135

UTCCS Report 10-01



筑波大学 計算科学研究センター

www.ccs.tsukuba.ac.jp



〒305-8577 茨城県つくば市天王台1-1-1