

次世代スーパーコンピュータを 活用する人材を育てる

工学院大学
小柳義夫

2007/11/29

SS研究会

1

何が問題か

- IT分野の人材養成
 - 20万人不足、インドに取られる、……
 - どんな人材が必要か
 - 諸分野の人材をITに活用できるか
- 計算科学技術の人材養成
 - 第3の科学としての計算科学
 - 「計算」は下等なことか？
 - なぜ諸分野だけに任せておけないか

2007/11/29

SS研究会

2

「一流の物理学者はコンピュータを使わない」?

- 1964年頃、東大にOKITAC7090が設置される
 - アセンブリ言語: OKISIP
 - アルゴル風言語: ALGOLIP
- 1966年頃、大型計算機センターにHITAC5020設置
 - FORTRAN: HARP5020
- 高橋・後藤研究室に、FACOM270/20
 - 一晩借り切ることができた

「一流の物理学者はコンピュータを使わない」?

- 素粒子論研究室: 木下、山本、村井、小柳などが、コンピュータを使って素粒子理論の研究 (Regge pole, N/D model, SU(3)など)
- 指導教授のM先生曰く
「一流の物理学者はコンピュータを使わない」
- 歴史的には間違い: von Neumann, Fermi, Alder, Feynman,

「一流の物理学者はコンピュータを使わない」?

- 物理だけではない
- 化学のスペクトロスコピーの某大家は、弟子にコンピュータの購入を許さなかった。そこで、分析機器に内蔵されたコンピュータをこっそり利用していた、とか?
- コンピュータ科学の大家であるM先生、「計算は土農工商の末端」?
- その影響は今でも

コンピュータの歴史に見る人材の育ち方

1. 神代時代
2. 大型計算機センターの時代
3. スーパーコンピュータ前夜
4. スーパーコンピュータの登場
5. 並列計算機での立ち後れ

神代時代

- 1950年代:コンピュータの萌芽時代
 - TAC, PC-1, PC-2,
 - 友隣電気:リレー計算機の計算サービス
- プログラミング:おそらく機械語(アセンブラはあったかどうか?)
- 利用者はそれぞれexpertiseを保有。
利用者=プログラマ

大型計算機センターの時代

- 1960年代半ばから7大学に大型計算機センターが設置された。
 - 東大HITAC5020、京大FACOM230-60など
- 森口繁一著「JIS FORTRAN入門」
 - ドリル方式でFORTRAN文法を独習
 - 多くのユーザはこれによってコンピュータを利用
 - 「文法を覚えれば、プログラムできるのか？」

大型計算機センターの時代

- 当時の計算機の特徴
 - 演算器が遅かった
 - メモリ(もちろんコアメモリ)とのバランスがよかった
 - 現在のように多段のキャッシュだの仮想記憶だのメモリレイテンシだのを考える必要が少なかった
 - それで済んだ
- 当時のチューニング
 - むしろ、メモリがすくなかった。例えば、東大のHITAC 5020Eの主記憶は64KW (256KB)
 - 大きな問題をコンピュータで取り扱うには、メモリを節約するためにプログラム上の工夫が必要であった。
 - 見やすいプログラムよりは、アクロバティックに圧縮したプログラム

2007/11/29

SS研究会

9

スーパーコンピュータ前夜

- ベクトル計算機の時代
 - 1976 Cray-1
 - 1977 FACOM 230-75 APU
 - 1978 HITAC M180 IAP
- IAP研究会(故唐木幸比古氏)
 - IAPを使いこなす手法の講義
 - IAPによる使用事例、
 - ベクトル計算機の技術動向
- これにより、多くのユーザは、ちょっとしたプログラミングの工夫によって、速度が大きく変わることを学び、自分の利用技術を向上させた。
 - 東大周辺でのベクトル計算機の利用技術という点では、このIAP研究会の果たした役割はきわめて大きかった。

2007/11/29

SS研究会

10

スーパーコンピュータの登場

- 1983 HITAC S810/20 (東大)
 - このような新しいアーキテクチャの計算機にしてはユーザの対応が速かった
 - IAP研究会などで、ユーザがベクトル計算機の利用の技術をかなり身につけていた
- ベクトル化コンパイラ
 - 当時のベクトル化コンパイラの性能は十分ではなかった
 - 初期のCrayのコンパイラよりはるかにまし
 - プログラムを高速化するためにはいろいろなノウハウが必要(それにより2~30倍も高速化!!)
 - スーパーコンピュータの時代になっても、IAP研究会はしばらく続けられ、今度はスーパーコンピュータの利用技術について情報交換を行った。

スーパーコンピュータの登場

- 日本のベクトル化コンパイラの飛躍的發展
 - その後、日立、富士通、NECのベクトル化コンパイラは驚くほど改良され、使いやすくなった。
 - Cray, Alliant, Titan, Convexなどアメリカのコンパイラよりも優れているという報告 (SC'89)。
 - 反面、日本のユーザは、ちょっとだけプログラムに手を入れれば、スーパーコンピュータは高速に動いてくれるという環境に慣れてしまった。
- スーパーコンピュータの戦略的設置
 - 7つの大型計算機センターやいろいろな分野の研究所に設置され、日本の科学技術の重要なインフラとなった。
 - 1980年代後半において、アメリカにはもちろん多数のスーパーコンピュータが設置されていたが、主として原子力、軍事、宇宙などの研究所にあり、一般の大学の研究者はあまり利用することができなかった。これに危機感を抱いて始まったのが、HPCC計画である。

並列計算機での立ち後れ

- 並列ベンチャーの台頭
 - 1980年代中頃から、アメリカでは多くのベンチャービジネスが並列計算機を作り始めた。
 - アメリカではスーパーコンピュータを利用するチャンスが乏しく、高性能な計算をするには、このような新しい機械を使う必要があったのである。
- 日本の立ち後れ
 - 日本でもこのころから、並列アーキテクチャや相互接続網について盛んに研究されてはいた
 - 日本の企業は、並列計算機を画像処理、論理シミュレーション、ハードウェア設計などの専用システムとしてしか認識していなかった(たとえばスパコン大プロ1981-9)。
 - 応用諸分野のユーザも並列処理を難しい技術としてあえて取り組もうとするものは多くなかった。

並列計算機での立ち後れ

- 本格的並列計算機の登場
 - 1990年代になると、CrayやIBMが本格的に並列計算機に参入し、スーパーコンピュータの業界を席卷し始めた。
 - その背景には、汎用プロセッサチップの急速な性能向上("Attack of Killer Micros"と呼ばれる)があった。
 - これにより10年ほど前に雨後の筍のように登場した多くのベンチャーは、消滅するか吸収されてしまった(3)。
 - 日本でも、Cenju-2 (1993), AP1000 (1994), Cenju-3 (1994), AP3000 (1996), Cenju-4 (1997) などの商用高並列計算機も登場したが、商品としての位置づけは、実用的な計算機というより並列処理研究用のテストベッドであった。

筑波大学での私の経験

- 1980年代前半：格子ゲージシミュレーションへの取り組み
 - まず汎用機
 - Metropolis法の理解と実装
 - ILUCR法の発見
 - ベクトル計算機の利用
 - ベクトル化Metropolis法
 - ベクトル化乱数
 - ILUCR法の超平面法によるベクトル化
 - チューニング技術、とくにunrolling

2007/11/29

SS研究会

15

筑波大学での私の経験

- 星野・川合のPAX計算機との出会い
 - 並列アルゴリズムの基本の勉強
 - 2次元torus NNM接続。まさに格子計算に最適と直感
 - 4次元格子の2次元メッシュへのマッピング技術
 - 並列化Metropolis法の開発
 - 並列化乱数生成法
- 格子ゲージ専用機へむけて、qcd-pax, cp-pacsへ

2007/11/29

SS研究会

16

何が欠けているか？

- 計算が研究の重要な柱であることの認識
 - シミュレーション:ミクロの法則からマクロの現象へ
 - 予期しない結果が出てくる
- **計算の意識化・明示化**
 - モデルの定式化
 - アルゴリズムの開発
 - プログラム開発技法(「動くだけではダメ」 文法から会話へ。ソフトウェア工学的手法の必要性)
 - アーキテクチャに則したプログラム最適化

数年後のスーパーコンピュータ

- 新しいMooreの法則:「チップのコア数は18ヶ月で倍増する」
- 数十コアのチップが登場
 - コアの制御、チップ内並列化
 - メモリ・ウォール問題(転送速度が追いつかない)
 - 内部メモリ(どう制御するか)
- 巨大な相互接続網
- **「アーキテクチャを意識しなくては、プログラムを最高速で走らせられない。」**

期待される人間像

1. 計算機オタク(これでは**ダメ**)
 - なんでもいいから速い計算機を作りたい。
 - 作ったから、みなさん使ってください。
 2. 応用オタク(これでは**ダメ**)
 - 論外オタク:このプログラムをそのまま速く
 - 速い計算機を作ってくださいれば、いかようにもプログラムをチューンして使いこなして見せます。
 3. 計算機と応用の融合
 - 応用(複数)のために計算機を設計し、計算機を見て応用を開発する。
- **Architecture-Algorithm-Application co-design!!**

2007/11/29

SS研究会

19

計算科学の確立

- 「計算科学」は独立の分野か？
 - 「理論」「実験」と並ぶ第3の科学だというのが、ただの「理論」とか「実験」という分野はない。
 - では、「物理」や「化学」などの中に埋没？
- 計算科学は横型の科学
 - しかし、共通の手法があり、**共有**すべき知識は多い。
 - モデル化、アルゴリズム、計算手法、データ解析
- **「共有を意識化すべきである」**

2007/11/29

SS研究会

20