次世代スーパーを中核とする拠点形成

工学院大学
小柳義夫

何が問題か

- 計算科学技術をどう推進するか
  - 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？
- 拠点の役割
  - スーパーコンの高度利用
  - それによる諸分野の新しい展開
  - 諸分野の人材を計算科学に活用

「一流の物理学者はコンピュータを使わない」？

- 60年代後半の素粒子論研究室：小柳など数人が、当時設置された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、弟子に大型コンピュータの利用を許さなかった(?)。そこで、弟子は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？
- 数値計算の大家であるM先生、「計算（そろばん）は士農工商の末端」？
- その影響は今でも

「計算」は下等なこと？

- 多くの分野の研究者は、計算を多用しているが、それについて語ろうとしない。
  - 計算科学の研究発表でも、各分野の成果は語るが、モデル化、アルゴリズム、コーディング手法、アーキテクチャ活用などについては語らない。
  - 「おれのプログラムを誰か高速化してくれ！」ではだめ。
- 計算科学の特徴
  - 違う分野でも、数値アルゴリズムやコンピュータの利用手法では共通点が多い。
  - その共通点での研究交流が必須。計算科学の発展のためにも、コンピュータの発展のためにも。

拠点の基本的機能

- 「資源を切り売りするセンター」ではなく、「研究・開発を企画・推進するセンター」
  - 大学の情報基盤センターも同様
  - ユーザからの要求を順番に対応するだけではなく、研究内容にまで踏み込んで、戦略的に利用を進める。
  - センター内の研究者の研究もあるが、全国（あるいは国際的な）研究プロジェクトの推進が中心。
  - アプリもあるが、共通的なアルゴリズム、基盤ソフト、アーキテクチャなどの研究が特に重要。

次世代スーパーの特徴

- ベクトル部とスカラ部から成る。
  - ベクトル・ユーザ:「われわれのプログラムがそのまま走るんだ。」
  - スカラ・ユーザ:「われわれのPCクラスタのプログラムがそのまま走るんだ。」
- これらはいずれも大きな誤解である。
  - これまでのベクトルやスカラとは大きく違うマシンである。

次世代スーパーの特徴

- チップ中に超多数の演算器(multi/many core)
  - それらが休みなく動作しないと性能は出ない。
  - どう制御するか。
- 相対的に狭いバンド幅
  - チップ内のバッファーメモリ:どう制御するか
  - アルゴリズム、コーディング、モデル化の工夫
- 超高並列処理
  - コア間、ノード内、ノード間、S－V間
  - OpenMP, MPI, …..

次世代スーパーの特徴

- コンパイラやツールがどこまで面倒を見てくれるか？
  - コンパイラ屋は、「任してください」と言う。
  - 私見では、当面望み薄！
  - 性能を出したければ明示的に指示する必要：
    - （部分的）アセンブリー言語
    - 指示文(directive)
    - 技巧的コーディング、など
  - チューニング・ツールに期待

では誰がやるのか？

- ユーザの期待:「次世代センターに持って行けば、そこの支援者が高速化してくれる。」
  - 幾ばくかの助言は得られるであろう。
  - しかしプログラムを見ただけでは分からない。
  - プログラムを一番分かっているのはユーザであるから、ユーザの関与・努力が必要。
  - 分野の近い支援者は有効であろう。
- チューニングはユーザと支援者の共同作業。
  - ユーザ側にも一定の技術知識は必要。
  - 最高速をねらうには、ユーザの努力が必須。

次世代スーパーの利用者像

- 松ユーザ：ギンギンにチューニングされたプログラムを最高速でぶっ飛ばす。
  - 期待されるユーザであるが、それだけに限定しては次がない。
- 竹ユーザ：ほどほどにチューニングされたプログラムを走らせるユーザ。
- 梅ユーザ：初心者。まだ未熟だが、このマシンに触り、他のユーザと交流することにより、進化することを期待。どんどん使わせる。
- バランスが肝心。

センターの運用

- ジョブのスケジューリング
  - 超並列・ハイブリッドアーキテクチャ
  - スケジューリングは複雑で、ソフトだけで効率化は困難
  - 人間の和による調整も必要
- 「協議によるスケジューリング」: 松ユーザや竹ユーザの一部のような、大規模ジョブ。
  - Face-to-face で調整。戦略的スケジュール。
- 「自動スケジューリング」: 梅ユーザなどの小規模ジョブはソフトに任せる
- 地球シミュレータ、KEK，筑波大CCSなどの経験

人材育成

- 従来の計算科学の人材養成：FORTRANの文法を勉強すればプログラムが書ける。
  - 「英文法を勉強しても英会話はできない。」
  - モデル化、アルゴリズム、コーディング技法などは自己流のまま。
  - あるいは、研究室先祖伝来のプログラムを中も知らずに使い続ける。
- 計算速度が遅く、複雑なメモリ階層もなく、メモリバンド幅が相対的に大きかったので、どうにかなった面もある。

人材育成

- 日本のベクトル計算機（80年代）:
  - 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
  - でも初期のCrayのコンパイラよりははるかにまし
  - 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!!）
  - 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。

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巻し始めた。
  - 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。

並列処理の人材育成

- 「並列ソフトウェアコンテスト」
  - 1994年から2001年　情報処理学会JSPP併設
  - 並列計算機を何種類か用意し、与えた問題を何分で解くか。
  - 大学院生、大学生、専門学校生、高校生を対象
  - 高校生のグループが入賞したこともある。
  - アメリカのSCxy国際会議のK-12に刺激されて発足
  - 次世代スーパーの一部を使って開催する可能性も考えられる（若手への刺激）

まとめ：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

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」