HPC と省電力設計

牧野淳一郎
理化学研究所
計算科学研究機構

話の概要

- 何故、スパコンで省電力？
- ENIAC から「京」まで
- The Intel Way
- 地球シミュレータと GRAPE-6
- HPC における省電力
- ポスト「京」
- まとめ

何故、スパコンで省電力？

もちろん、

スパコンがどんどん省電力でなくなってきたから

どれくらい省電力でないか？

2011年某社某氏との会話

「 天文台の次期システムはどうですか」

私「 電力がですねえ、、、 今 140 で、、、 今のレンタル料でそちらのシステム入れると 600kW くらいにはなりますよね？」

「 良い線ですね、 空調もいれると×××」

国立天文台三鷹キャンパスの総電力量は 1.5MW しかない、、、

2011年某社某氏との会話

「 天文台の次期システムはどうですか」

私「 電力がですねえ、、、 今 140 で、、、 今のレンタル料でそちらのシステム入れると 600kW くらいにはなりますよね？」

「 良い線ですね、 空調もいれると×××」

国立天文台三鷹キャンパスの総電力量は 1.5MW しかない、、、

で、 どうなったかというと、、、

2011年某社某氏との会話

「 天文台の次期システムはどうですか」

私「 電力がですねえ、、、 今 140 で、、、 今のレンタル料でそちらのシステム入れると 600kW くらいにはなりますよね？」

「 良い線ですね、 空調もいれると×××」

国立天文台三鷹キャンパスの総電力量は 1.5MW しかない、、、

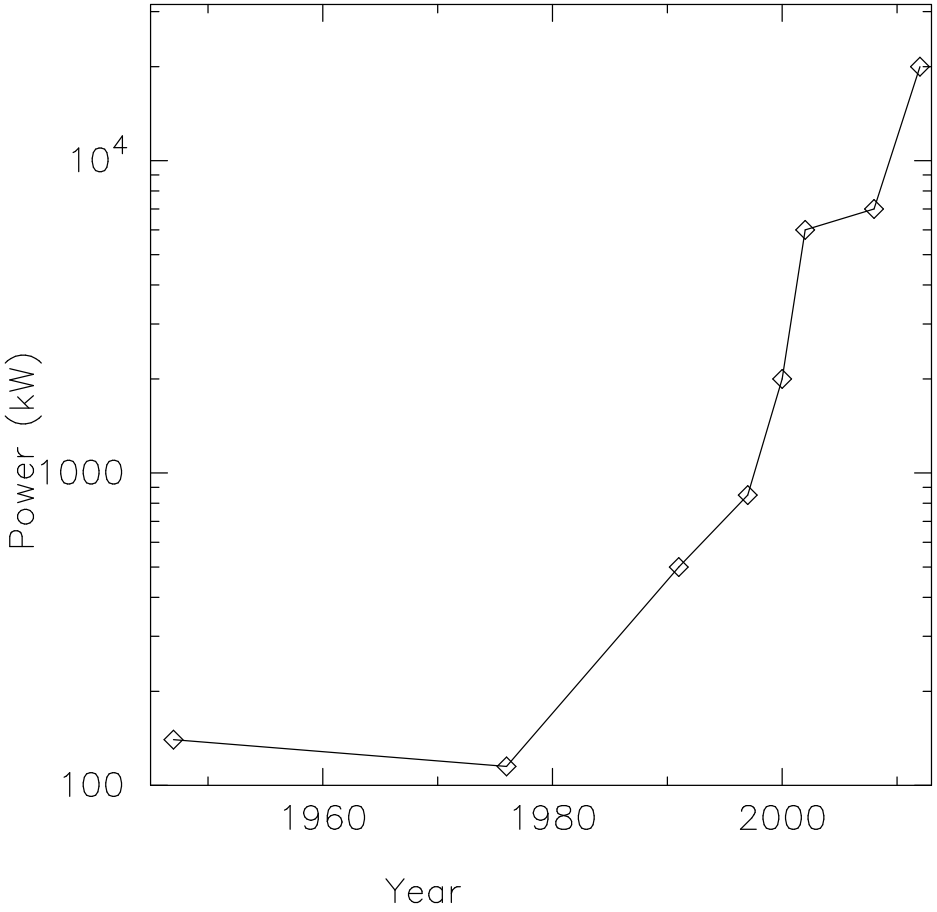
で、 どうなったかというと、、、

スパコンは三鷹キャンパスではなく水沢キャンパスに

ENIAC から「京」まで

ENIAC	1947	140kW
Cray-1	1976	115kW
Cray C90	1991	500kW
ASCI Red	1997	850kW
ASCI White	2000	2MW
ES	2002	6MW
ORNL XT5	2008	7MW
「京」	2012	20MW

グラフにしてみると、、、



ENIAC から Cray-1 まであまり変わらない

そのあと20年間で10倍

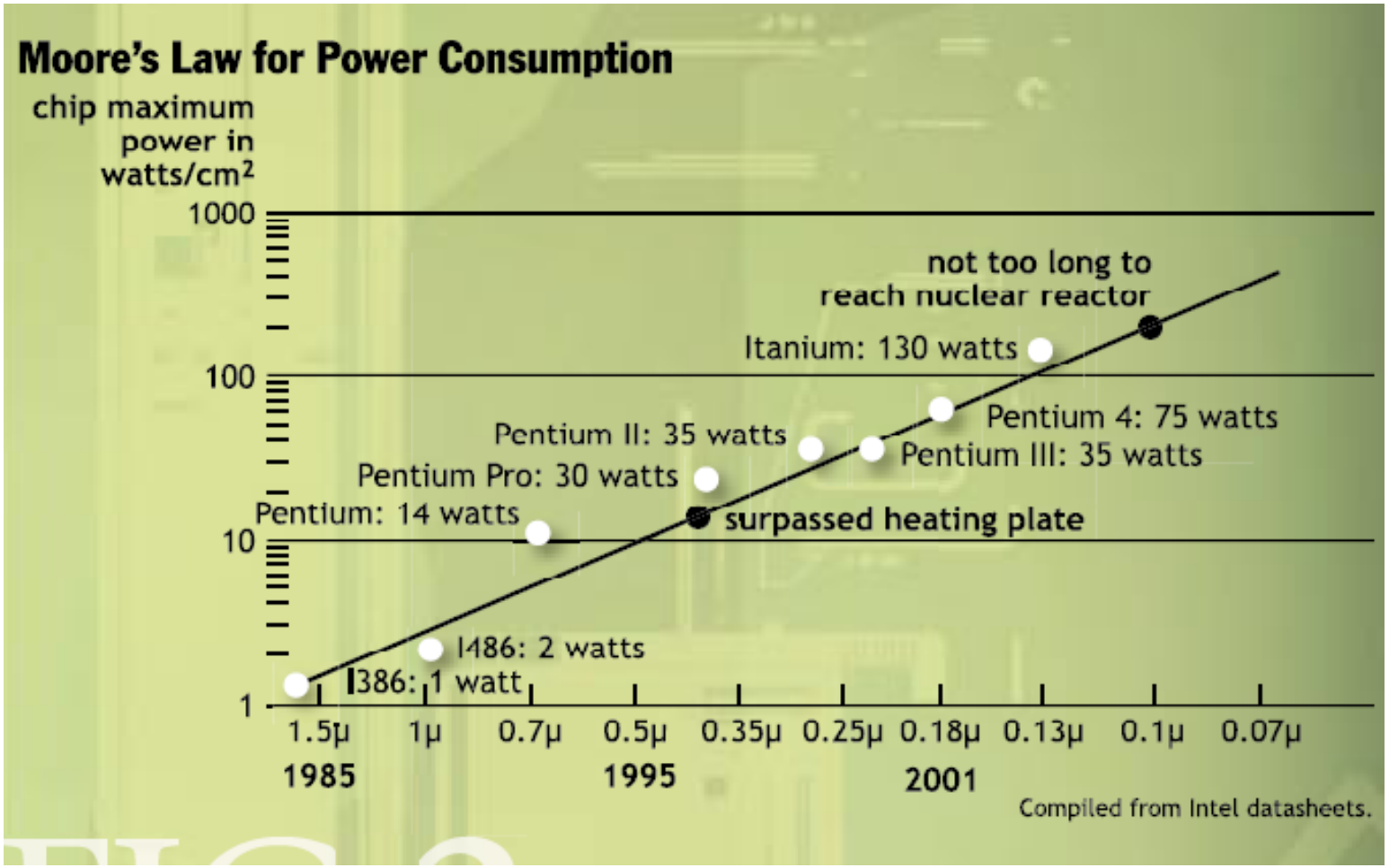
そのあと10年間でさらに10倍

何故こんなことに？

理由

- 計算機に使うお金が増えている。ASCI Red は $ 50M、「京」は、、、、
- プロセッサの面積当り消費電力が増えた
- プロセッサの面積当りの値段が下がった

面積当りの消費電力



Feng 2003 から、　実は 2003 年以降は増えてないけど、、、

$100\mathrm{W/cm^2}$ の意味

- 普通にパッケージにいれて強制空冷で (まあ水冷でも同じ) 冷やせる限界
- クロックの上限を決める。この10年間 CPU のクロックは殆ど上がってない

古典的 CMOS スケーリングとの関係

古典的スケーリングの時代 (130nm くらいまで)

フィーチャーサイズ 1/2 → 面積当りキャパシタンス 2 倍、

電圧 1/2、クロック 2 倍 → 消費電力不変

- サイズの 3 乗に反比例して電力性能上がるはず (実際にはそこまであがってなかった=非効率な高クロック設計)

ポスト古典スケーリングの時代 (130nm 以降)

- 電圧下げるのは限界、 リーク分も増大
- 消費電力一定 → クロック下げる必要発生
- 設計による電力効率の向上が必要になった (130nm まで無駄してきたのを回収すればいいという話もある)

で、これは問題か？

- 「そういうものだ」と思えば別に問題ではない！？
- 10年で100倍とかのペースで計算速度をあげたいと思うと問題
  2020年くらいにエクサフロップスになっても、200MW必要では困る(電気代のほうが高い！)

性能を落とさずに電力を減らす？どうやって？

The Intel Way

半導体世代が変わった時に

- コア数はちょっとづつ増やす
- コア内演算器数 (SIMD 幅) 結構どんどん増やす
- 全コアが動く時のクロックは抑える
- 1 コアだけの時にはクロック上げる

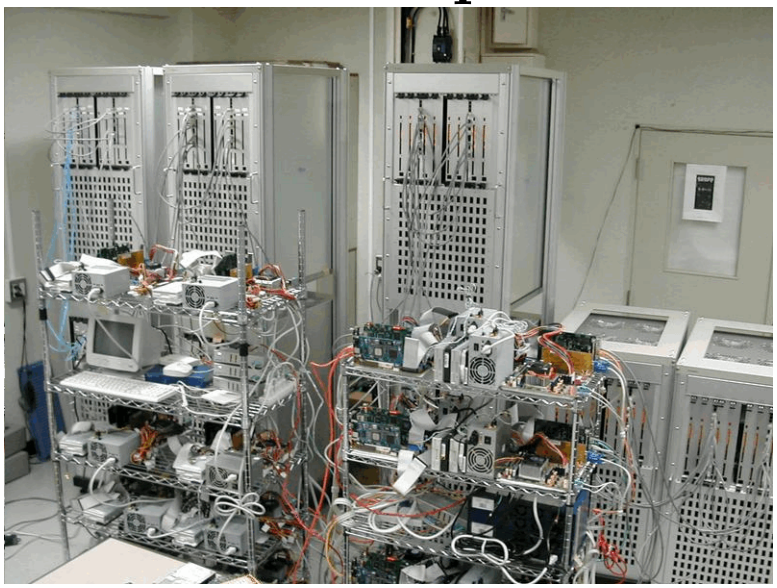
これで名目ピーク性能をあげつつ、並列化できていないプログラムでもある程度の性能向上

Intel を追いかけてては勝負にならない。もうちょっと違うやり方はないか？

地球シミュレータと GRAPE-6

GRAPE-6 2002 250nm

50kW 64 Tflops



ES 2002 150nm

6MW 40 Tflops

電力当り性能は 100 倍以上違う

何が違うか?

プロセッサチップの比較

	地球シミュレータ	GRAPE-6
テクノロジ	150nm	250nm
面積	400mmsq	200mmsq
トランジスタ数	6000万	800万
クロック	500MHz	90MHz
消費電力	140W	15W
演算性能	8Gflops	30Gflops
演算器の数	16	～300
Gflops/W	0.06	2

GRAPE-6は重力相互作用計算専用パイプラインプロセッサ。

ESは「汎用」

要するに

演算器当りのトランジスタ数が違う

GRAPE-3	4千	専用パイプライン
GRAPE-6	3万	専用パイプライン
GRAPE-DR	40万	SIMD 超並列
Cray-1	40万	古典的ベクトルプロセッサ
Intel 80860	60万	初期マイクロプロセッサ
地球シミュレータ	400万	最終期ベクトルプロセッサ
Fermi	300万	GPGPU
Sandy Bridge	4000万	現行マイクロプロセッサ

Cray-1 の40万はベクトルレジスタ含んでないかも

SB は大半のトランジスタがキャッシュメモリなのでちょっと不当な数えかた？でもキャッシュも電気は食っている。

というわけで、、、HPC における省電力とは何か？というと

- まず第一に、「演算器あたりのトランジスタ数を減らす」＝高効率設計
- その次: 使ってないところは動かないようにする
- もちろん、他にも無限に色々なことがある
  - 動作速度と電力消費のトレードオフ: 回路構成、トランジスタ種類、動作電圧、パイプライン段数等

どうやってトランジスタ数を減らすか？

という以前に、現代のプロセッサでなにがトランジスタ数と電力を喰っているのか？

- 演算器そのもの
- レジスタファイル
- 命令フェッチ、デコード等のロジック
- キャッシュ
- 外部メモリとメモリインターフェース

演算器そのもの

- 現代の典型的なプロセッサでは電力消費の主要な部分ではない
- 通常の設計では今後みえてくる
- 極端な省電力設計ではもちろん最後にはこれが見える
- 低クロック化、パイプライン段数を減らす
- 演算器自体の構成(特に乗算器)も重要

レジスタファイル

- データ幅、ポート数、ワード数でサイズと消費電力が決まる
- スーパースカラー、VLIW でポート数が増えるのは低消費電力化に逆行

命令フェッチ、デコード等のロジック

- 賢くすれば必ず電力増える
- あまりに賢くないと、演算器の効率低下、結果的に電力性能低下に
- SIMD の幅を広げるのは極めて有効。命令デコーダの数自体が減る
- 但し、あまり広げると実行効率の低下や演算器、レジスタファイル、キャッシュの間のデータ移動の電力が増える(各ブロックが大きくなり、配線が長くなる)

キャッシュ

- 現状ではおそらく電力消費の主要な部分
- L1, L2, (あればL3) がそれぞれ大量の電力消費
- L1 はしょうがないとして、L2, L3 をなるべく使わないようにできれば大きく電力性能あげられる

外部メモリとメモリインターフェース

- キャッシュと並んで電気喰うところ
- なるべくバンド幅下げたい

まとめると

- データを動かすと電気を喰う
- チップ内部でも遠くに動かすと沢山電気喰う (レジスタ、L1, L2, L3, 外部メモリとどんどん増える)

つまり

- 階層キャッシュは悪である
- 外部メモリはバンド幅きりつめるべきである
- チップ内部でもメモリと演算器は物理的に近くに置くべき
- 当然、物理的に遠くのメモリへはバンド幅下がる

ハードウェアの要求はそうだとして、使いものになるの？

データ局所性

HPC アプリケーションの極めて多く:偏微分方程式を扱う

- 物理法則は偏微分方程式でかかれている
- 陰的解法だと連立一次方程式を解くが、最近陽的解法の発展が目覚ましい
- 粒子系でも、遠くはまとめられる
- 遠くでもまとめられないような系だと、ものすごく計算量が多いのでデータ移動コストは相対的に小さい

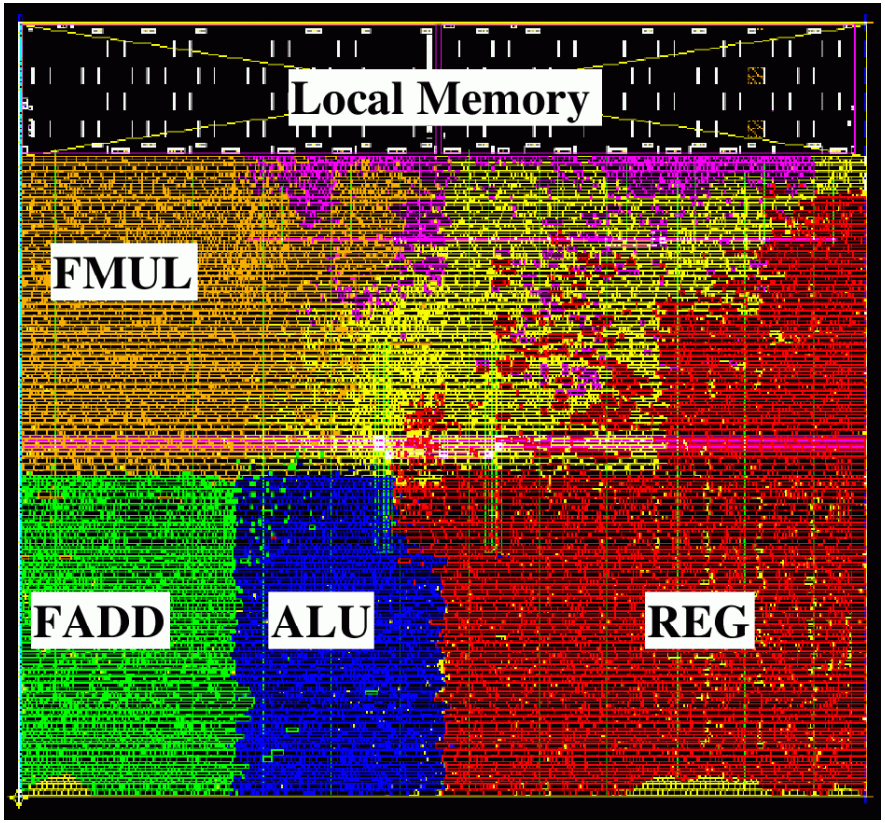
演算器は近くのメモリは速くアクセス、遠くは遅く、でもなんとかなる

実際のハードウェアの概念

- 小さなメモリ+演算器で基本ユニットを構成
- これをチップに沢山いれて、SIMD 動作させる
- 基本ユニット間、チップ間のネットワークは、想定するアプリケーションの要求から決める。

そういう考え方で作ったもの: GRAPE-DR (2004-2008 のプロジェクト)

GRAPE-DR PE のフロアプラン



0.7mm by 0.7mm

浮動小数点演算器の部分は
チップ面積の 1/5 以下
(1/3 以上くらいにはした
かったぃぃ)

比較からわかること

- 専用パイプラインで計算精度まで切り詰めれば演算器当り
  1万トランジスタ以下にできる
- 汎用プロセッサでは 30 万トランジスタくらいが限界
- Cray-1, Intel 80860, GRAPE-DR はその辺
- GPU はその10倍
- x86 はさらにその10倍

実際の数字は？

Little Green 500, June 2010

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	815.43	National Astronomical Observatory of Japan	GRAPE-DR accelerator Cluster, Infiniband	28.67
2	773.38	Forschungszentrum Juelich (FZJ)	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
2	773.38	Universitaet Regensburg	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
2	773.38	Universitaet Wuppertal	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
5	536.24	Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw	BladeCenter QS22 Cluster, PowerXCell 8i 4.0 Ghz, Infiniband	34.63

#1: GRAPE-DR, #2: QPACE: German QCD machine

#9: NVIDIA Fermi

Green 500, Nov(Dec) 2010

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	1684.20	IBM Thomas J. Watson Research Center	NNSA/SC Blue Gene/Q Prototype	38.80
2+	1448.03	National Astronomical Observatory of Japan	GRAPE-DR accelerator Cluster, Infiniband	24.59
2	958.35	GSIC Center, Tokyo Institute of Technology	HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows	1243.80
3	933.06	NCSA	Hybrid Cluster Core i3 2.93Ghz Dual Core, NVIDIA C2050, Infiniband	36.00
4	828.67	RIKEN Advanced Institute for Computational Science	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect	57.96
5	773.38	Universitaet Wuppertal	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54

#1 BG/Q #2+: GRAPE-DR,

#2,3: NVIDIA Fermi, #4: K-computer

トランジスタ数を切り詰めることの効果

- 半導体技術では 3 世代遅れの GRAPE-DR でも Little Green500 でトップになれたりする。
- チップ単体の性能はもっと高い。 4Gflops/W くらい。

トランジスタ数を切り詰めることの効果2

- 少ないマンパワーで設計できる。
- そもそも設計しないといけないものが少ないため
- 開発コストも小さくなる。まあ、それでも 10 億くらい。

トランジスタ数を切り詰めることの問題点

- GRAPE-DR の場合、 メモリバンド幅を犠牲にしている。
- GRAPE-DR の場合、各 PE が外付メモリをランダムアクセスとかはできない。
- 但し、これは対象にしたアプリケーションがあんまりメモリバンド幅いらないものだったから。１桁くらいなら増やせなくもない。

エクサスケールにむけて？

- 2012-2013の２年間、「演算加速機構を持つ将来のHPCIシステムに関する調査研究」ということでフィージビリティスタディ
- 文部科学省的結論「技術的可能性は実証された」
- といういいかたの意味するところは、、、

まとめ

- HPC において省電力はもっとも優先度の高い課題になりつつある
- 省電力の定義＝電力あたり性能。つまり、実際に計算している時にどれだけ電気を喰わないようにできるか。
- 色々なアプローチがあるが、究極的には、演算器とメモリを物理的に近くするのが重要になろう。
- 次の国家プロジェクトは、、、