
第4回
2018/12/11
「コンピュータの性能はどう考えるのか？」

計算機システムの性能とは？

以下のような議論で、「性能」とは何か？

- 計算機を、クロック周波数が2倍高いものに買い替えたが、私のプログラム速度は1.5倍しか上がらなかった・・・
- 某大学スパコンが新しくなって、隣の研究室のプログラムは3倍速くなったらしいが、私のプログラムは1.2倍しか速くならない・・・
- 速度が上がった計算機は発売されたが、私の解析プログラムはメモリ500GBないと動かないので、今の計算機から乗り換えられない
- スマホを、3Dグラフィック性能2倍という新しいものに買い替えたが、電池の持ちは半分になってしまった。そういえば私は3Dゲームしない・・・
- ある研究者は、世界一スパコンの10倍もの速度のスパコン設計を考えた。ただし電力は100倍消費し、原発1基分必要である。この計画は認められるか？

計算機システムの性能とは？(続き)

さまざまな指標：

処理速度、メモリ・ストレージ容量、グラフィック性能、消費電力
ほかにも、ネットワーク速度、互換性、サイズ、質量...

⇒どの指標が重要かは、ユーザ・用途による (カタログやセールストークが決めるのではない)

今回は、主に計算機の**速度に注目**。それでも様々な議論が

例：計算機AとBの処理速度を比較した。

プログラム1については、Aのほうが1.2倍速かったが、プログラム2についてはBのほうが1.5倍速かった → なぜこのようなことが起こる？

プログラムの実行時間

Linux/Macで、

```
% time your-command opt...
```

```
real      0m0.048s
```

```
user      0m0.031s
```

```
sys       0m0.006s
```

- “real” → Elapsed Time, Real Time (経過時間)
 - 実際のプログラムの実行の経過時間
= 全ての時間のトータル (CPUの実行時間+ディスクのアクセス待ち/読み書き時間、OS内部の時間、その他入出力、他のユーザの実行、etc.)
- CPU time
 - I/Oや他のプログラムの実行時間は数えない
 - “sys” → system time (OSの内部の実行時間)と、
 - “user” → user time (OSの外部でのユーザプログラム自身の実行時間)
 - マルチコアを用いた場合はその合算となるので、Elapsedを超えることも
- どの指標を使うのかがよいかは場合による。以下では原則userに注目

一般的な性能の定義

- マシン X で動作しているプログラムに対し、

$$\text{性能}_X = 1 / \text{実行時間}_X$$

- 「X は Y より n 倍性能が良い」

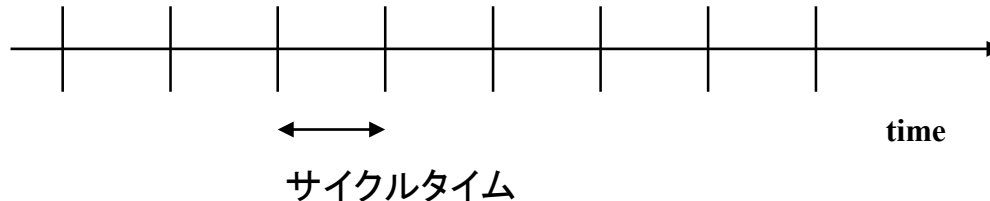
$$\text{性能}_X / \text{性能}_Y = \text{実行時間}_Y / \text{実行時間}_X = n$$

- 実行時間は、小さいほうがよい。性能の数値は、大きいほうがよい
- 問題点:
 - あるマシン A はあるプログラムを20秒で実行する
 - 別なマシンBは同じプログラムを25秒で実行する
 - Q: 果たして、全てのプログラムで、マシンAはマシンBより1.25倍性能が良いと言えるであろうか？

クロックサイクル

- 実行時間を、秒数だけでなく、より精密にはクロックサイクルを用いることもできる

プログラム実行時間(秒)
= プログラムの総サイクル数 / クロック周波数(Hz)



- クロック周波数 = 毎秒あたりのサイクル数 (Hz = 1 cycle/sec)
- サイクルタイム = クロックの刻みの間隔 = $1 / \text{クロック周波数}$
-

A 200 MHz のクロックは $\frac{1}{200 \times 10^6} \times 10^9 = 5 \text{ ナノ秒}$ のサイクルタイム

どのように性能を向上させれば良いか？

プログラム実行時間(秒)

= プログラムの総サイクル数 / クロック周波数(Hz)

= プログラムの総サイクル数 × クロックサイクル(秒)

Q: したがって、性能を向上させるには(他の部分が等しかったとすると)

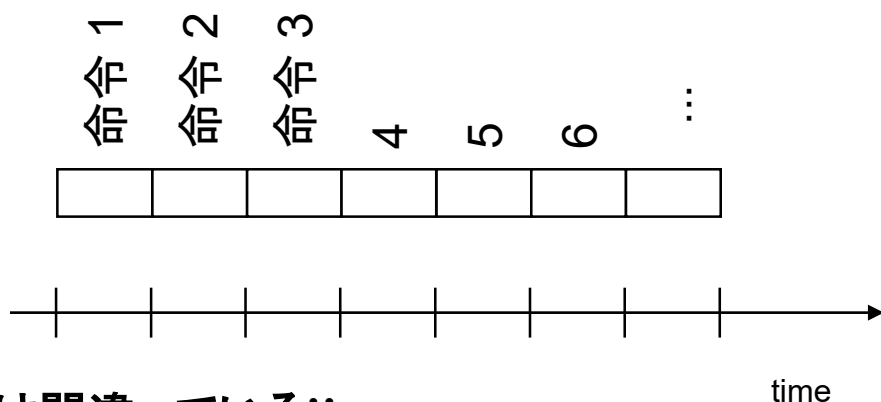
プログラムが要求するサイクル数を_____、または

クロックのサイクルタイムを_____、つまり、

クロックレートを_____。

プログラムの実行には何サイクルかかるのか？

- 最初の仮定?? プログラムの総サイクル数
= プログラムで実行される命令数



この仮定は間違っている!!

異なる命令は、異なるサイクル数がかかる

ある命令のサイクル数は、CPU AとCPU Bで(同じISAでも)異なりうる
同じ命令、同じCPUですら(例えば lw)、場合によりサイクル数が違う

何故か？

メモリアクセスはレジスタ間操作より時間がかかる

キャッシュミス/ヒットで時間が違う

ジャンプがあると、時間がかかる

浮動小数点命令は、時間がかかる、などなど

サイクルタイムによる性能比較の例

Q:

- 我々がよく実行するプログラムPは200MHzのクロックを持つあるマシンA上で10秒で実行される。新型のマシンで、6秒以内で実行したい。
- クロックの可変のマシンBが使えるようになったが、CPU内部構成が変わっている。プログラムPをコンパイルしたところ(ISAが同じなので命令数は同じだが)、マシンAよりも1.2倍のクロックサイクルが必要と分かった。
- 上記の目的のために、マシンBを何Hzで動作させればよいか。

回答

- 仮にマシンA上でプログラムP実行に必要な総サイクル数を k とすると、
 - $\text{CPU時間}_A = k / \text{周波数}_A$
 - $k = 10 \times 2 \times 10^8 = 2 \times 10^9$ サイクル
- マシンBでのCPU時間は、
 - $\text{CPU時間}_B = k \times 1.2 / \text{周波数}_B$
 - $\text{周波数}_B = 1.2 \times 2 \times 10^9 / 6 = 4 \times 10^8 = 400\text{Mhz}$

クロックとサイクル、他の指標

- プログラムの実行の尺度
 - ある数の命令(機械命令)
 - ある数のクロックサイクル
 - ある時間(秒)
- これらの定量的尺度を表現する用語:
 - サイクルタイム cycle time (1サイクルあたりの時間(秒))
 - クロックレート/クロック周波数 clock rate (1秒あたりのサイクル数(Hz))
 - CPI (cycles per instruction、1命令あたりの平均クロックサイクル)
浮動小数点演算を多用するアプリはCPIが高くなるかもしれない
 - MIPS (毎秒何百万命令実行数 millions of instructions per second)
※MIPS ISAとは別の用語
 - CPI, MIPSは単純な命令を用いるプログラムでは良い数値→良い尺度ではない

速度性能を示すさまざまな尺度・指標

- あるプログラムに対しては性能は実行時間で決定されるのは先に延べた
- ほかのパラメータは性能を決定する有効な尺度になるであろうか？
 - プログラムを実行するのに必要なサイクル？
 - プログラムで実行される命令数？
 - クロックレート？
 - CPI？
 - MIPS？
- 良くある誤信：性能をある一つのパラメータのみで評価してしまうこと
← 本当はさまざまなパラメータがある

CPI の例

- Q: 同じ命令セットアーキテクチャ(ISA)の二つの異なるマシンがあったとしよう
(例: Core-i3マシンと Xeonマシン)

あるプログラムに対して,

マシン A のCPUはクロック周波数が2GHzで、CPI が 2.0

マシン B のCPUはクロック周波数が1GHzで、CPI が 1.2

このプログラムに対して、どちらのマシンがどの程度速い?

- ヒント: もし二つのマシンが同じISAであったとすると、我々の定量的尺度のどれが同一になる? (e.g., クロックレート, CPI, 実行時間, 総命令実行数, MIPS)
- 回答: 総命令実行数を k とすると、それぞれの実行時間 t_A , t_B は、
 - $t_A = k \times 2.0 / (2 \times 10^9)$
 - $t_B = k \times 1.2 / (1 \times 10^9)$
 - 性能_A/性能_B = $t_B / t_A = 1.2$

命令数の例

- Q: コンパイラのデザイナーがある特定の言語機能を機械語に変換するのに、あるマシンに対して二つの機械語のコードの列の選択肢があるとする。実装の性質により、命令は三種類に分けらる。それぞれのサイクル数は以下の通り:
 - クラスA: 1サイクル、クラスB: 2 サイクル、クラスC: 3 サイクル
- また、コード列は、以下の命令のクラスを含んでいるとする
 - 一番目のコード列は5命令: Aが2、Bが1、Cが2
 - 二番目のコード列は6命令: Aが4、Bが1、Cが1
- この時、
 - それぞれのコード列のCPIを求めよ
 - どちらのコード列がどれだけ速いか?

MIPS の例

- Q: 二つのコンパイラが100Mhzのマシン上でテストされており、大規模アプリケーション用のコードを出力する
 - 命令のクロックサイクルは前問同様に、クラスA: 1サイクル、クラスB: 2 サイクル、クラスC: 3 サイクル
 - 第一コンパイラのコードでは、クラスA 500万命令、クラスB 100万命令、クラスC 100万命令、が実行される
 - 第二コンパイラのコードでは、クラスA 1000万命令、クラスB 100万命令、クラスC 100万命令、が実行される
- この時、
 - MIPS値の高い方はどちらのコンパイラ?
 - 実行時間が短い(性能が高い)のはどちらのコンパイラ?

アムダールの法則 (Amdahl's Law)

改良後の実行時間 =

影響されない部分の実行時間 +(影響される部分の実行時間 / 改良の度合)

Q:

- あるプログラムのマシンA上の実行時間が100秒で、そのうち数値演算が80%の時間を占めていたとする。
- 新しいマシンBで、数値演算性能を大幅に向上させる技術が使えるようになった。数値演算以外はAと同じとする
- 実行性能を4倍にするには、マシンBの数値演算性能を何倍にしなければならぬか？
 - $T = 1/5T + 4/5T \Rightarrow 1/4T = 1/5T + 4/5T * 1/P \Rightarrow P = 16$
- また、実行性能を5倍にするのには??
 - $P \rightarrow \infty$
- 並列処理などにも良くAmdahlの法則が用いられる

処理速度の2つの考え方

- Response Time または Latency ← これまでは、この値かその逆数に注目
 - 私のプログラムが終わるのにどの程度時間がかかる？
 - 「ガチャを引く」ボタンをタップしてから、結果がわかるまで何ミリ秒？
 - 単位は秒
 - 数値は小さいほどよい
- Throughput
 - 時間あたり、あるマシンが一度に走らせられるジョブの数は？
 - 時間あたり、あるゲームサーバが最大で処理可能なリクエスト数は？
 - 単位は「何らかの数量」/秒
 - ジョブ/秒、リクエスト/秒、bit per second...
 - 数値は大きいほどよい

Throughputが、Latencyの逆数にならない例

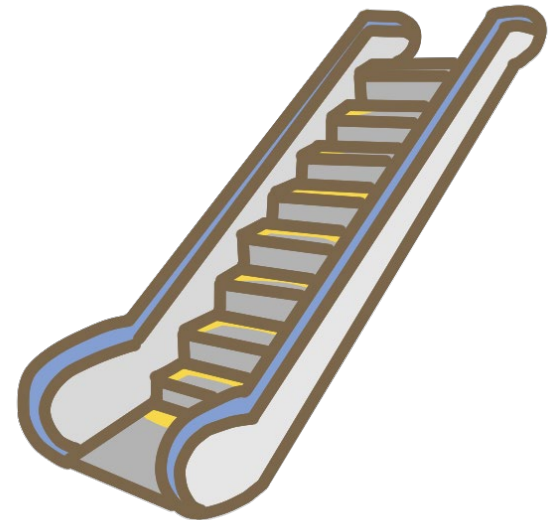
- あるエスカレータに乗ると、上につくまで20秒かかる。
- 上から下まで20段、一段に2人まで乗れる

Latencyは？ → 20秒

Throughputは？ → このエスカレータは満員の場合、1秒に2人を継続的運ぶことができる → 2人/秒

Latencyの逆数=0.05人/秒ではないのはなぜ？

これまでの議論との違い：一度にエスカレータに複数の人が乗ることができる



このエスカレータの処理速度を上げることを考えよう。

(A) 動く速度を2倍にする → Latencyは1/2に、Throughputは2倍に。しかしこの場合は危険

(B) 横幅を広げる、隣に増設する → 安全にThroughputを2倍にできる。しかしLatency変わらず

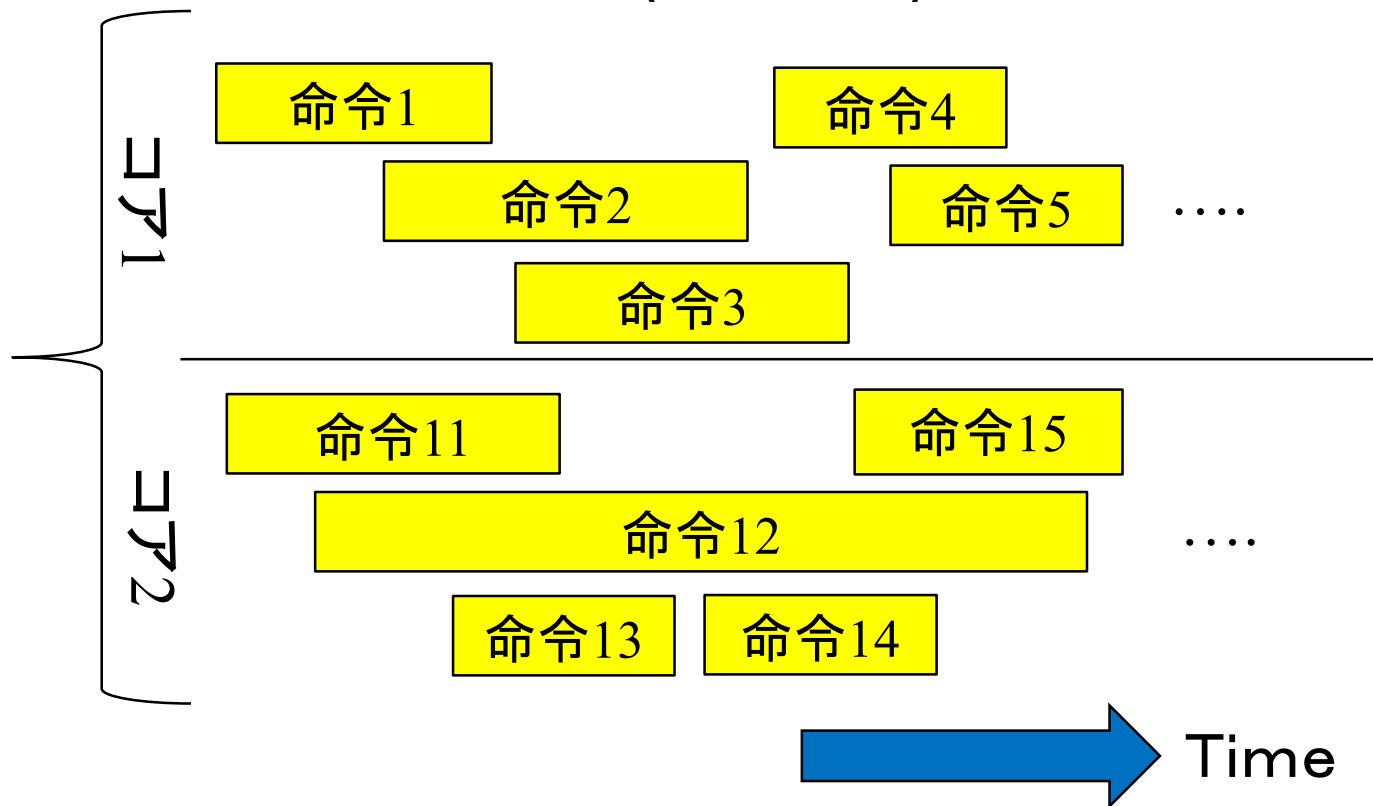
現在のプロセッサの動作は、先ほどの説明より複雑

理由1: 複数コアが同時に動いている

理由2: 1コアの中でも、各命令はパイプライン動作している

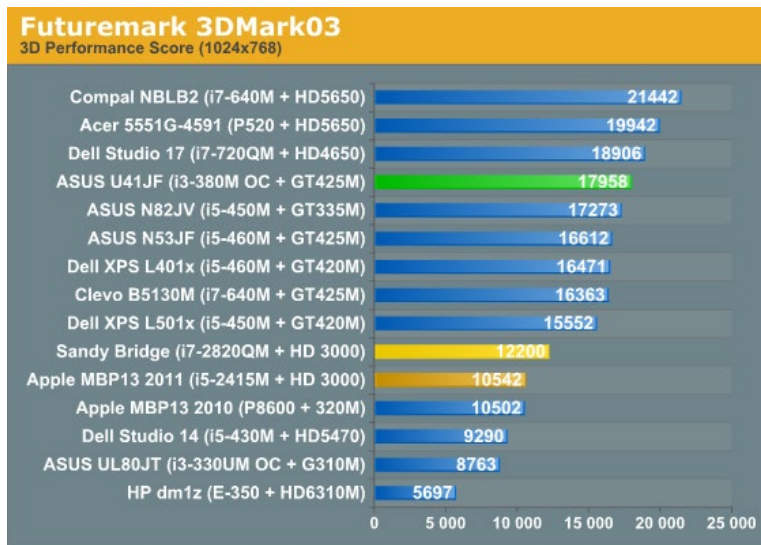
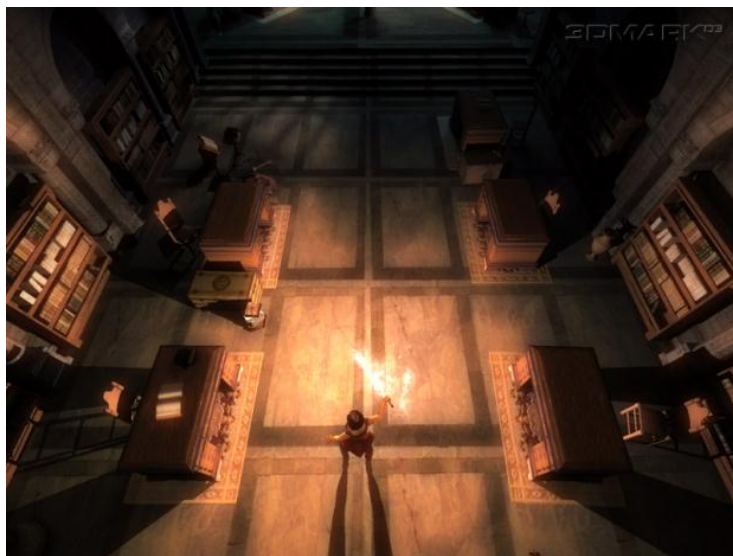
→ 命令1の時間 + 命令2の時間 + 命令3の時間... ではない

→ パイプライン技術により、CPIは大幅向上 (小さい数値)



ベンチマーク: 計算機の実行速度を示す道具

- ベンチマーク: 性能測定用の決められた処理の実行
- 3Dグラフィックス用のベンチマークも
- クロック周波数やコア数だけで、計算機性能を判断するよりはかなり良い方法
- だが、あるベンチマークと、自分が求める特性が一致するかは一般的に難しい



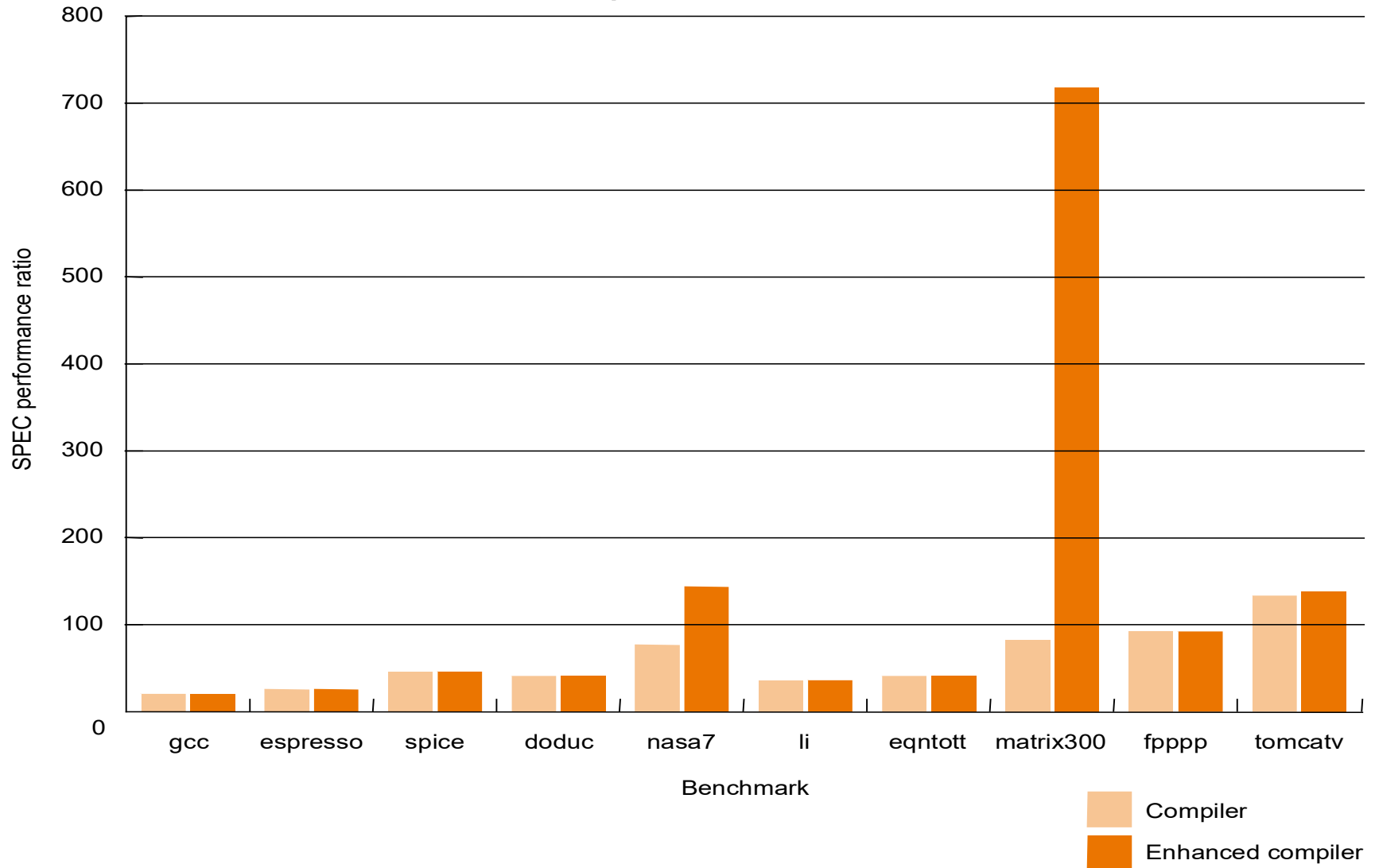
ベンチマークに関して

問: 計算機を買いたい、私のプログラムを速く実行するのはどれか？

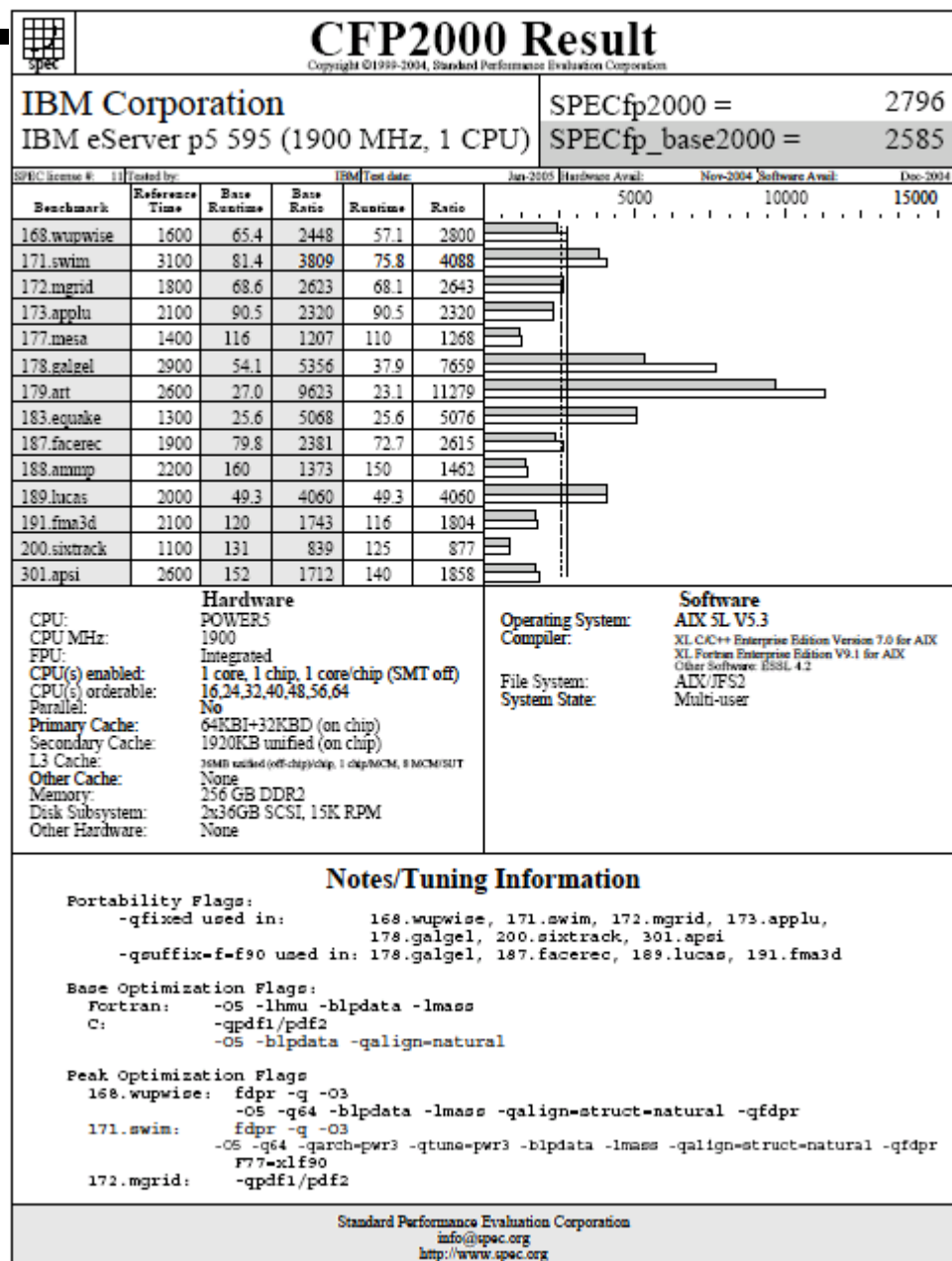
- 本当のプログラムをそれぞれの計算機で動かせばベストな判断ができるが、不可能なことが多い → ベンチマークを参考にする
- 小さいベンチマーク (Small benchmarks)
 - ハードウェアデザイナーにとっては扱いやすい
 - 標準化は容易: 例: Dhrystone, Whetstone, Sieve, stream
 - 乱用される恐れ: 針小棒大な結果、チューニングが可能 (benchmark tuning)
- SPEC (System Performance Evaluation Cooperative), www.spec.org
 - 多数の企業間で、本物のプログラムと入力を標準化 (数千~数万行)
 - 性能の非常に重要な尺度。アーキテクチャだけでなく、コンパイラ技術の評価にも用いられる
 - SPECSpeed2017 Int, SPECSpeed2017 FP, SPECrate2017 INT, SPECrate2017 FP
 - SPECviewperf, SPEC MPI2007, SPECEnterprise2018...
 - それでもやはり乱用される可能性がある(次のスライド)
- 他にもいろいろ

SPEC '89

- ベンチマークチューニングの結果 →matrix300は以後不採用に



Specの報告の例



スーパーコンピュータのベンチマーク: Top500

- スパコンでは最も有名なベンチマーク
- “The Top500 Supercomputing Sites” <http://www.top500.org/>
- 1993年より、世界のスーパーコンピュータのベンチマークを記録し、500位までリスト掲載
- 年二回更新、6月はドイツInternational Supercomputing Conferenceにて、11月は米国IEEE Supercomputingにて発表



Top500 とは? (その2、技術内容)



- ランキング指標: Linpack (Partial pivotingを行うLU分解)実行時のFLOPS
 - 行列サイズ n に対し $2/3 n^3 + O(n^2)$ 演算
 - プロセッサの演算性能、メモリ、ネットワークを測る(という題目)
 - プログラムの改造はokだが、標準的にはHigh Performance Linpack (HPL)というプログラムがある
 - ※ 2008年TSUBAME1.2のころは、GPUバージョンがなかったので遠藤がプログラム作成
- 用語
 - FLOPS: 秒あたりの浮動小数点演算回数の単位
 - 全命令を数えるMIPSと違う
 - Rmax – 計測されたLinpack実行のFLOPS値
 - Rpeak – 理論最高性能FLOPS値(カタログからわかる)
 - Efficiency (実行効率) – Rmax / Rpeak のパーセンテージ
 - Power – Linpack実行時の消費電力

Top500とは?(その3、競争)



- 25年で200万倍の速度向上
 - 59GFlops → 143PFlops
- 現在の世界一は米国Summitスパコン
- かつては日米が世界一を争う
- 今は米国と中国

2002年、日本の地球シミュレータが世界一に(2位の4倍の大差)

アメリカは”Computenik”ショック

1957年のSputenikになぞらえる

2009年、計画中の京コンピュータに対して事業仕分け「二位ではいけないんですか」→ 最終的に予算復活

→ 2011年に京は世界一獲得

Top500とは？(その4, 批判)

- ベンチマークが始まったころは、演算・メモリ・ネットワークのすべてを測る意図
 - 現在は、ほぼ演算性能で決まる。実際のスパコン上のプログラムはよりメモリ・ネットワークが大事
 - 計算機を多数ならべるだけでよいのか？電力は？
- 近年種々のベンチマークが提案されてきた
 - Green500, Graph500, HPCG, IO500...

The Green 500

- 最もTop500の性能に対して電力性能が良いスパコンのランキング
 - Top500に入らないと出場権がない
- Linpack実行の性能値をその際の電力で割った値。
- <http://www.green500.org>



TSUBAMEシリーズは 2マシンがGreen500 世界一

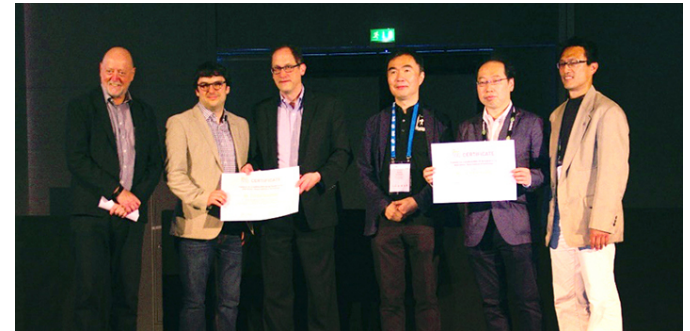
TSUBAME-KFC

- 油浸小型スパコン
- 2013/11, 2014/6世界一
- 4.50GFlops/Watt



TSUBAME3.0

- 2017/6世界一
- 14.11GFlops/Watt



性能評価・まとめ

- 性能は本来はある特定のプログラムに固有のものである
 - 全体の実行時間は性能の一貫性のある尺度となりうる
- あるアーキテクチャに関しては、性能向上は以下によってもたらされる:
 - クロックレートの増加 (CPIに対する著しい悪影響なしに)
 - CPIを低下させるためのアーキテクチャ構成の新技術
 - 例:パイプライン、スーパースカラなど
 - 命令数とCPIを減少させるようなコンパイラのコード生成
- 落とし穴: あるマシンの側面の改良が、必ずしもそれに比例した性能向上をもたらすとは限らない
- 性能のメトリックは、近年では実行時間以外にも、電力効率などいくつか重要な指標が出てきている。
- 性能に関して、カタログなどで読むことを全て信じてはいけない!!