

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ
高速化の作戦

実践編

事前にやっておくこと
性能解析入門
ジョブ投入方法
チューニング

目次

はじめに

基礎編

ハードウェアの復習と計算の
流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

計算を早く終わらせたい!

- ▶ 研究活動にプログラムは欠かせない (理論系に限らず)
 - ▶ 自分でプログラムを組むことも多い
- ▶ 計算に長時間かかる場合も (1 週間, 1 ヶ月, ...)
 - ▶ 高解像度, 長時間
 - ▶ 山ほどパラメータ振る
- ▶ 速いマシン (スパコン, 加速器) はそうそう使えない
- ▶ 手元で頑張る
 - ▶ ジョブ投入の仕方
 - ▶ プログラムチューニング
 - ▶ 並列化 (今回は話さない)

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

本発表のスタンス

- ▶ テクニクの羅列は少なく，原理 + 理詰めを志向
 - ▶ あとあと応用が効くように
 - ▶ 現在の CPU がずっと主流かは分からない
 - ▶ 研究では特殊な計算機を使うこともある
- ▶ 少ない手間でそれなりに速くなればそれでいい
 - ▶ チューニングは，やらないにこしたことはない

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

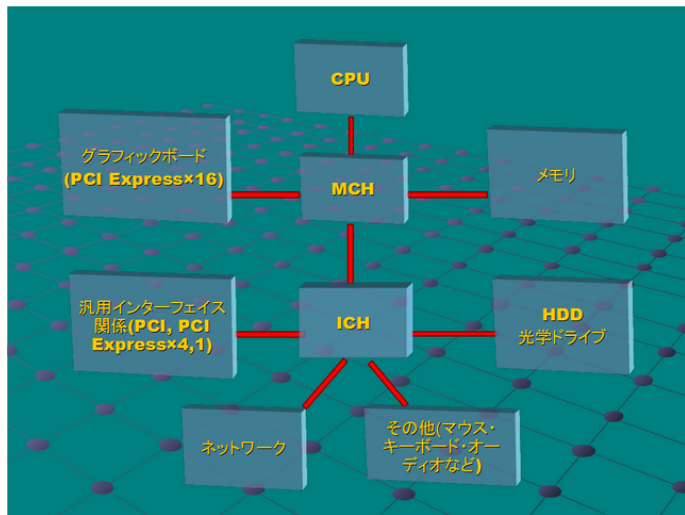
チューニング

基礎編

PC のハードウェアの例

高速化を考える

納多 哲史



目次

はじめに

基礎編

ハードウェアの復習と計算の流れ
高速化の作戦

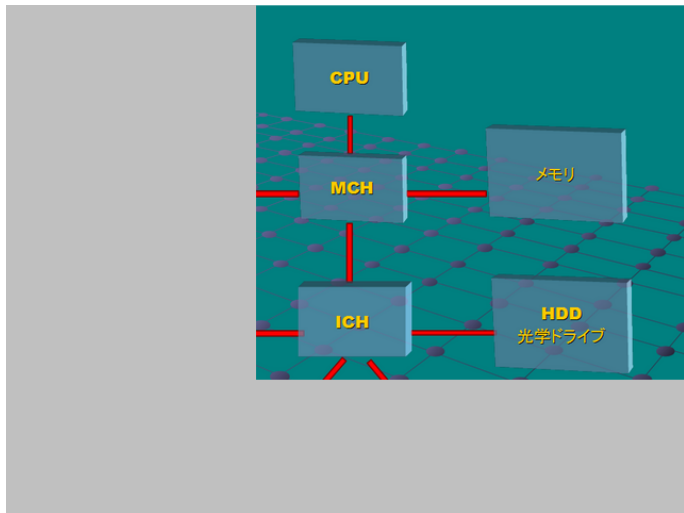
実践編

事前にやっておくこと
性能解析入門
ジョブ投入方法
チューニング

PC のハードウェアの例

高速化を考える

納多 哲史



目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

高速化の作戦

選択と集中

一番時間がかかる場所に手を入れたほうがコストパフォーマンスが良い

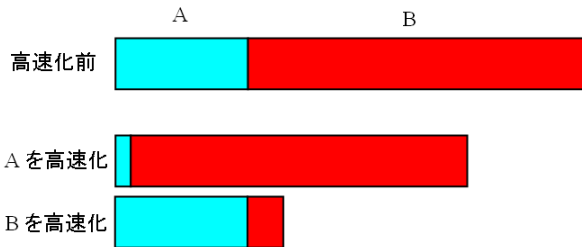


Figure: 手を入れた場所が 10 倍速くなると仮定したときの速度変化. 棒の長さが時間に対応する.

ごく一部のコードが大部分の計算時間を占めている場合もある.

高速化の手順

- ▶ 遅い場所の特定
- ▶ 遅い理由の特定
- ▶ 対処
 - ▶ 各過程の時間を減らす
 - ▶ 演算
 - ▶ 転送待ち
 - ▶ 各過程の時間配分を調整する
 - ▶ 肉を切らせて骨を断つ

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

高速化の手順

- ▶ 遅い場所の特定
- ▶ 遅い理由の特定
- ▶ 対処
 - ▶ 各過程の時間を減らす
 - ▶ 演算
 - ▶ 転送待ち
 - ▶ 各過程の時間配分を調整する
 - ▶ 肉を切らせて骨を断つ

遅い場所・理由をどうやって知るか？

→ プロファイラによる性能解析 (後述)

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

実践編

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

その前に...

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

本編の趣旨から外れるが 高速化には必須

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

ほとんど手間なく、 かなりの改善が 得られる方法

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

それは...

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

他人へ丸投げ

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

他人が配布しているものを
ありがたく利用しましょう。

コンパイラ

目次

はじめに

基礎編

ハードウェアの復習と計算の

流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

- ▶ **無料より有料のコンパイラを使った方が速い**
 - ▶ お金をかける場合, CPU だけでなくコンパイラにも投資推奨
 - ▶ Intel コンパイラ (Linux) の非商用利用は無料
 - ▶ 神戸の情報実験機には Fujitsu コンパイラがある (Squeeze からは使えない)
- ▶ インストールやオプションを調べるのに手間はかかるが最初の一回やればよい

コンパイルオプション

よく授業で習う例:

```
$ g95 program.f
```

最適化オプションをつける:

```
$ g95 -O program.f
```

- ▶ 最適化の内容はコンパイラ依存. 詳細は man を参照.
- ▶ やりすぎるとエラーや精度低下などの副作用があることも

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

ライブラリ・組込関数の利用

高速化を考える

納多 哲史

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

- ▶ チューニング済のルーチンが提供されていることがある。
 - ▶ ライブラリ (例: LAPACK, FFTW, ISPACK)
 - ▶ 組込関数 (例: Fortran 90 における行列の転置, 行列積)
- ▶ コードが短くなるメリットも

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

本編に戻る

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

性能解析入門

GNU Profiler の使い方

コンパイルオプションをつける

```
$ g95 -pg program.f
```

これによって出来たバイナリファイルを実行すると `gmon.out` というファイルができる。

```
$ gprof ./a.out
```

で性能情報が標準出力に現れる。

- ▶ コンパイラによってコマンド名などが異なるが、基本は同じ。詳細は `man` を参照。
- ▶ 性能解析込みのプログラムは遅いので本番では使わないこと

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

性能情報の例

Flat profile:

Each sample counts as 0.01 seconds.

%	cumulative	self		self	total	
time	seconds	seconds	calls	s/call	s/call	name
70.81	4.90	4.90	500	0.01	0.01	sub2_
27.89	6.83	1.93	200	0.01	0.01	sub1_
1.16	6.91	0.08	1	0.08	6.91	MAIN_

左から順に

- ▶ 全体に対する割合
- ▶ 上からの累積時間 (sec)
- ▶ 時間 (sec)
- ▶ 呼ばれた回数
- ▶ 1 回あたりの時間 (sec)
- ▶ 同上 (下位ルーチン含む) (sec)
- ▶ 名前

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

より詳細に計りたい場合

高速化を考える

納多 哲史

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

時間を計測する関数を使う.

- ▶ Fortran
 - ▶ system_clock (f90)
 - ▶ cpu_time (f95)
- ▶ C 言語
 - ▶ clock

実行プロセス数

- ▶ 大量のプロセスが同時に走ると、余計な処理が生じる
 - ▶ レジスタやキャッシュへのデータのロード、アンロードが繰り返し替えされる
- ▶ CPU コア数 (ハイパースレッディング可能ならその 2 倍) までが無難

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

スワップ落ち抑止

- ▶ 全プロセスの使用するメモリ合計が物理メモリの合計を上回るとスワップ領域がメモリとして使われう。
 - ▶ HDD なので非常に遅い
- ▶ メモリを大量に使用するジョブの投入は少なめに
 - ▶ 経験的に convert はメモリを大量消費するので注意

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ
高速化の作戦

実践編

事前にやっておくこと
性能解析入門
ジョブ投入方法
チューニング

ディスクアクセス

- ▶ HDD は磁気ヘッドからデータを読み取る
- ▶ 複数箇所と同時にデータの読み書きの命令があると各所で少しずつ読み書きしていくのでヘッドが移動する分の待ち時間が増える
- ▶ ディスクアクセスの多いジョブは同時に1本までにしておくのが無難
 - ▶ gzip など
 - ▶ 複数のジョブを投入したいときはシェルスクリプトなどで逐次実行させるとよい

目次

はじめに

基礎編

ハードウェアの復習と計算の

流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

目次

はじめに

基礎編

ハードウェアの復習と計算の
流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

計算量の削減

重複している計算をひとつにまとめる

before

$$a = (\text{重い計算}) + a0$$
$$b = (\text{重い計算}) * b0$$

after

$$\text{tmp} = (\text{重い計算})$$
$$a = \text{tmp} + a0$$
$$b = \text{tmp} * b0$$

注意: 「(重い計算)」はすべて同一の計算

後で使いまわせる計算結果は変数に格納しておくと計算時間を節約できる

転送の削減

先ほどと矛盾しているように見えるが、計算量と転送の待ち時間のトレードオフ

before

```
n=100000000
```

```
do i=1,n
```

```
  tmp(i) = x(i)*y(i)
```

```
end do
```

```
do i=1,n
```

```
  a(i) = tmp(i) + a0
```

```
  b(i) = tmp(i) * b0
```

```
end do
```

after

```
n=100000000
```

```
do i=1,n
```

```
  a(i) = x(i)*y(i) + a0
```

```
  b(i) = x(i)*y(i) * b0
```

```
end do
```


おまけ: メモリアクセス最適化

メモリアクセスは連続させる. アクセス時に連続した場所から複数個取ってくるため.

before

```
do i=1,ni
  do j=1,nj
    b(i,j) = a(i,j)
  end do
end do
```

after

```
do j=1,nj
  do i=1,ni
    b(i,j) = a(i,j)
  end do
end do
```

注意: C 言語は配列の添字の並び順が上記と逆.

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング



Amazon.co.jp より引用

Hisa Ando, 2011: プロセッサを支える技術, 技術評論社, ISBN978-4774145211.

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ
高速化の作戦

実践編

事前にやっておくこと
性能解析入門
ジョブ投入方法
チューニング

まとめ

- ▶ アーキテクチャ(どうやって計算を行っているか)を知ることが高速化の近道
- ▶ チューニングの前にやれることはやっておく
- ▶ プロファイラによって時間がかかる場所を絞り込む

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

参考文献

- ▶ Hisa Ando, 2011: プロセッサを支える技術, 技術評論社, ISBN978-4774145211.
- ▶ Kevin Dowd, 1994: ハイ・パフォーマンス・コンピューティング- RISCワークステーションで最高のパフォーマンスを引き出すための方法, インターナショナルトムソンパブリッシング, ISBN978-4900718036.
- ▶ 北海道大学 EPnetFaN 座学編 2006 年度「チューニングのすすめ」(光田千紘)
 - ▶ <http://www.ep.sci.hokudai.ac.jp/~epnetfan/zagaku/2006/0714/pub/>
- ▶ Fortran 入門: Fortran Tip 集
 - ▶ http://www.nag-j.co.jp/fortran/FI_18.html#AUTOTOC_18_7

目次

はじめに

基礎編

ハードウェアの復習と計算の流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング

参考文献

- ▶ 神戸大学 ITPASS 実習 2010 年度「ハードウェア解説」(佐伯拓郎)
 - ▶ <http://itpass.scitec.kobe-u.ac.jp/exp/fy2010/100521/lecture/pub/>
- ▶ 理化学研究所 情報基盤センター「チューニング技法入門」
 - ▶ <http://acc.riken.jp/HPC/training.html>
- ▶ 東京大学 情報基盤センター「チューニング連載講座」
 - ▶ <http://www.cc.u-tokyo.ac.jp/publication/news/>
- ▶ アムダールの法則 - Wikipedia
 - ▶ <http://ja.wikipedia.org/wiki/アムダールの法則>

目次

はじめに

基礎編

ハードウェアの復習と計算の

流れ

高速化の作戦

実践編

事前にやっておくこと

性能解析入門

ジョブ投入方法

チューニング