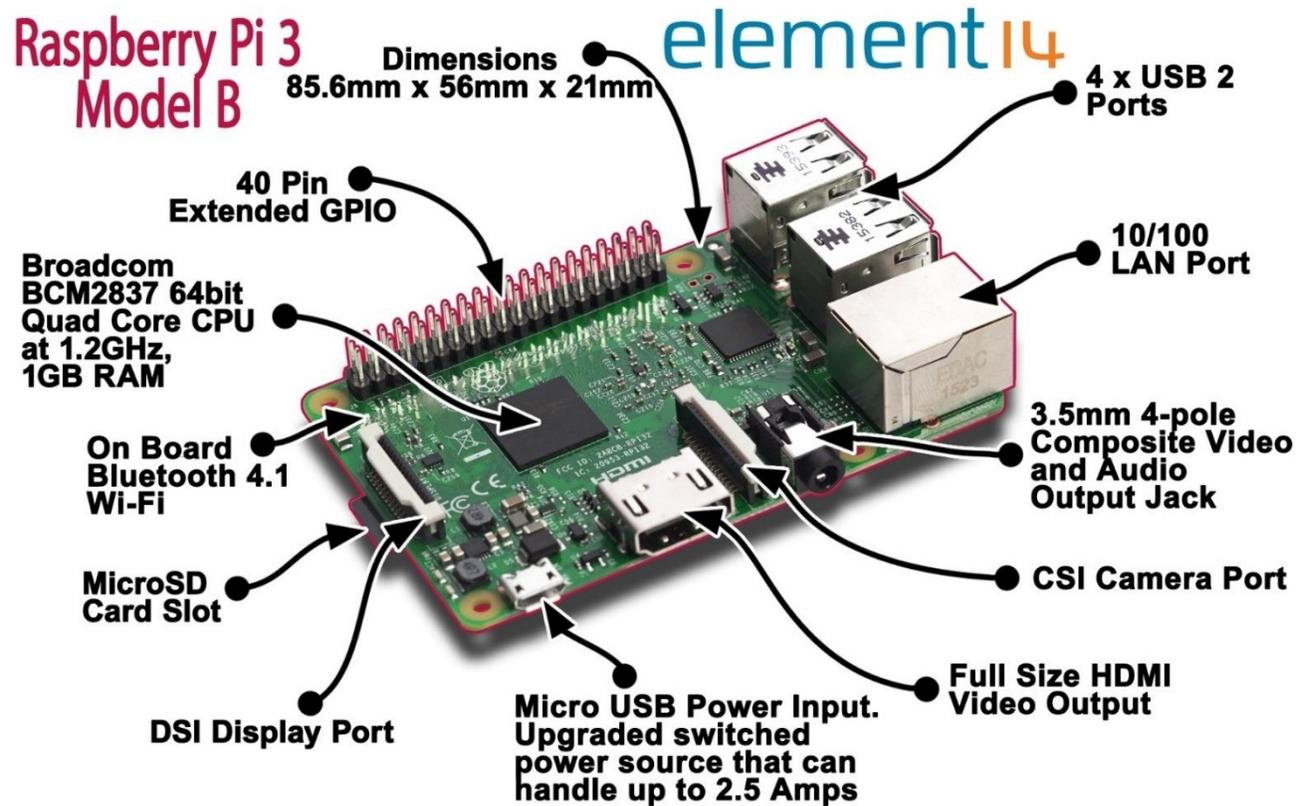


Raspberry PiでのEGS5の実行



川崎医科大学
辻 修平

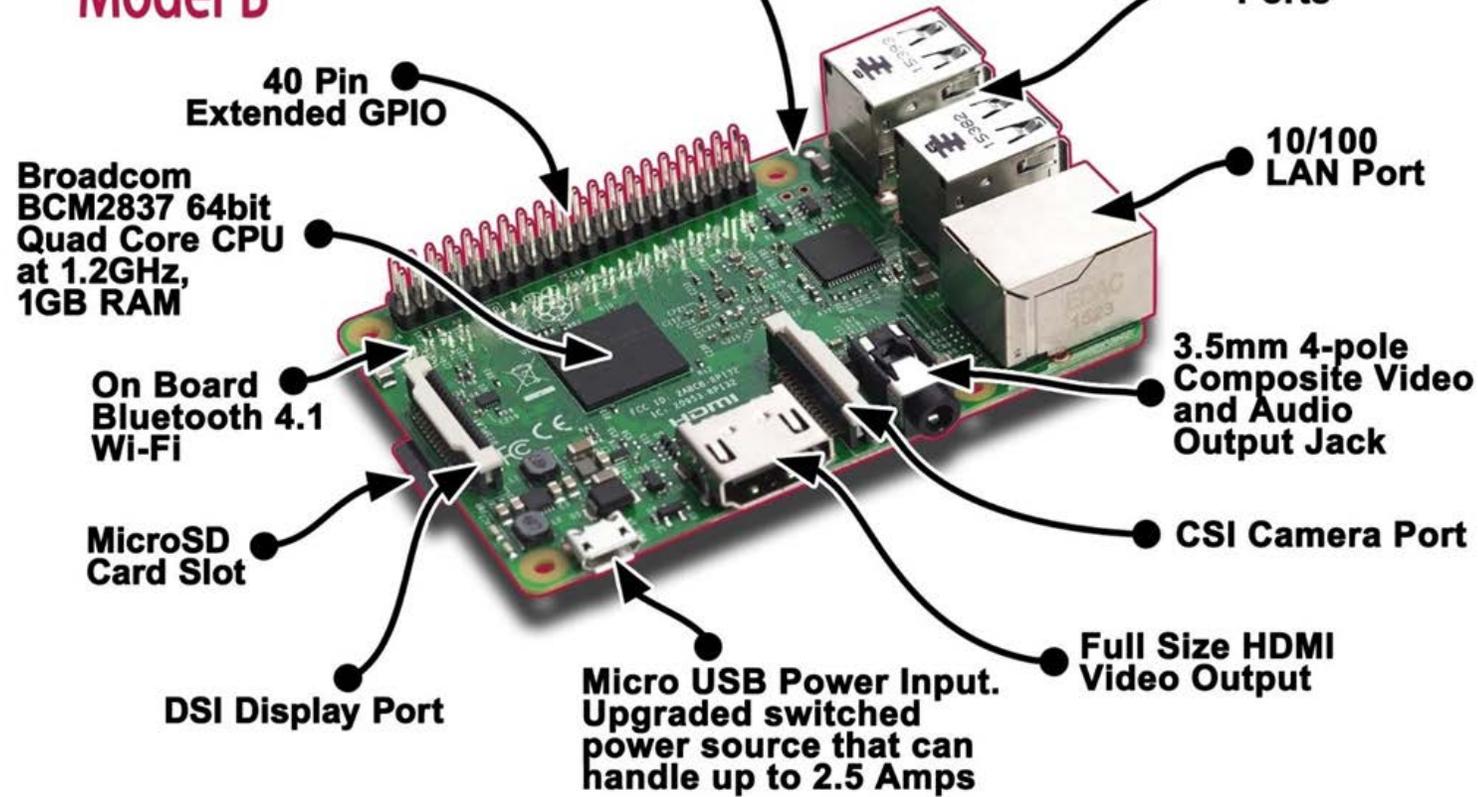
Raspberry Piとは。

Raspberry Piは、かつてイギリスで**教育用**コンピュータとして普及したエイコーン社「BBC Micro(1981年)」の再来として、学校で基本的な**コンピュータ科学の教育**を促進することを意図している。(Wikipediaより。)

Raspberry Pi 3 Model B

Dimensions
85.6mm x 56mm x 21mm

element14



ハードウェア

64bit Quad Core 1.2GHz
ARM Cortex-A53

1GB RAM

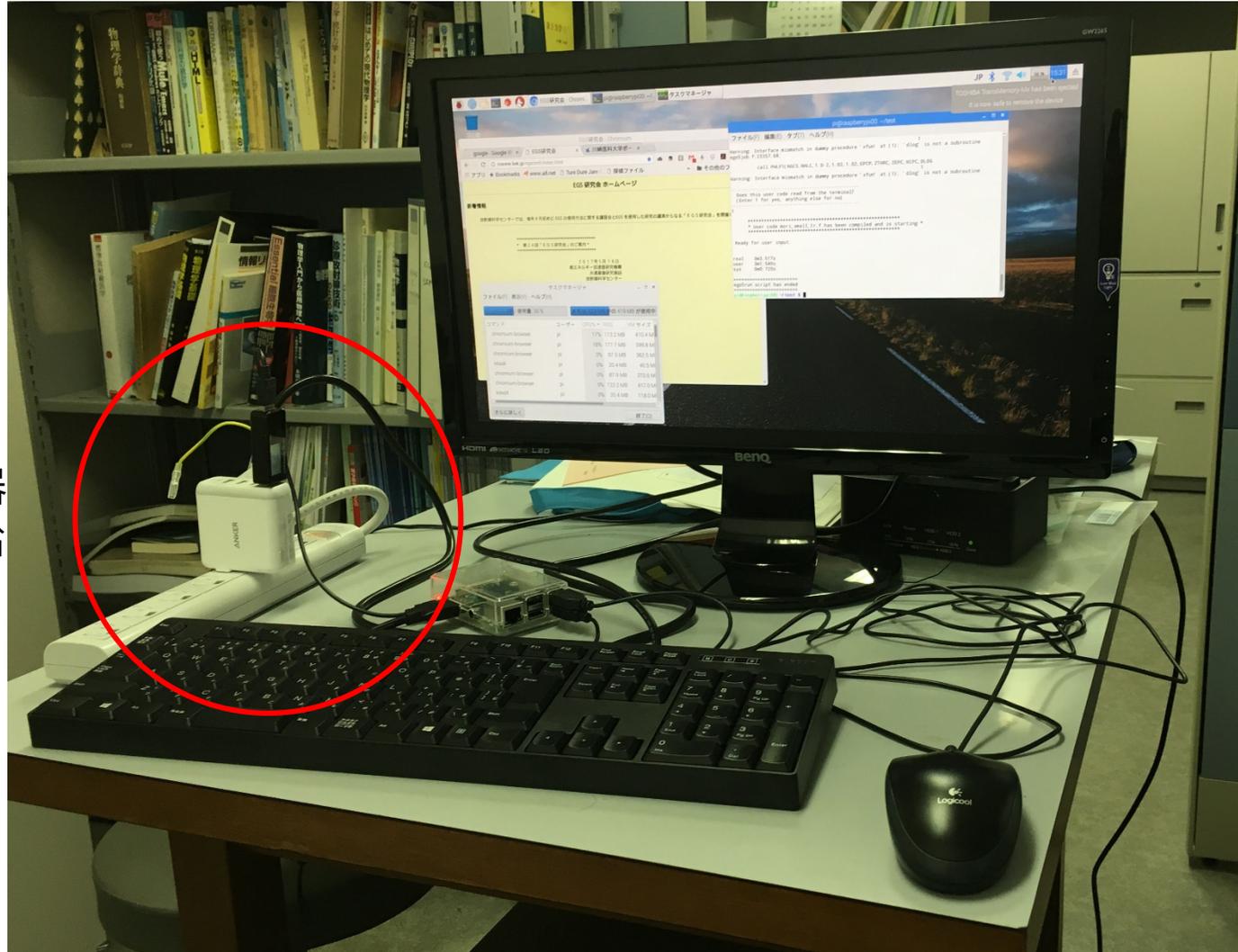
ストレージ
MicroSD(32GB以下が望ましい。)

ソフトウェア

Raspbian, Ubuntu MATE,
Windows10 IOT CORE, ...etc.

Linuxが動く

モニター、キーボード、マウスをつなげると・・・・・・・・



USBの充電器
から電源供給

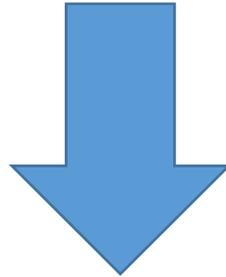
ネットはwifiが
使えます。

LANポートも
付いています。

普通に使えます。

はたして本当に「**教育用**」なのだろうか。

実用できないだろうか？



ある「**展望（野望？）**」に気づいた。 **（後述）**

とりあえず、EGS5を動かしてみる。

具体的方法

初期設定

<https://www.raspberrypi.org/downloads/raspbian/>から
〇〇〇〇-raspbian-Jessie.zipをダウンロード (現時点 : Release date:2017-07-05、Kernel
version:4.9)

Windows上で展開

DD for Windowsをダウンロードして展開。

microSDカードへカードを入れ、DDWin.exeを「管理者として実行」する。
書き込んだmicroSDカードをRaspberry Piに入れて電源を入れると起動する。

ターミナルで

```
sudo apt-get update  
sudo apt-get upgrade  
sudo rpi-update
```

参考URL

<http://blog.shibayan.jp/entry/20150228/1425121187>

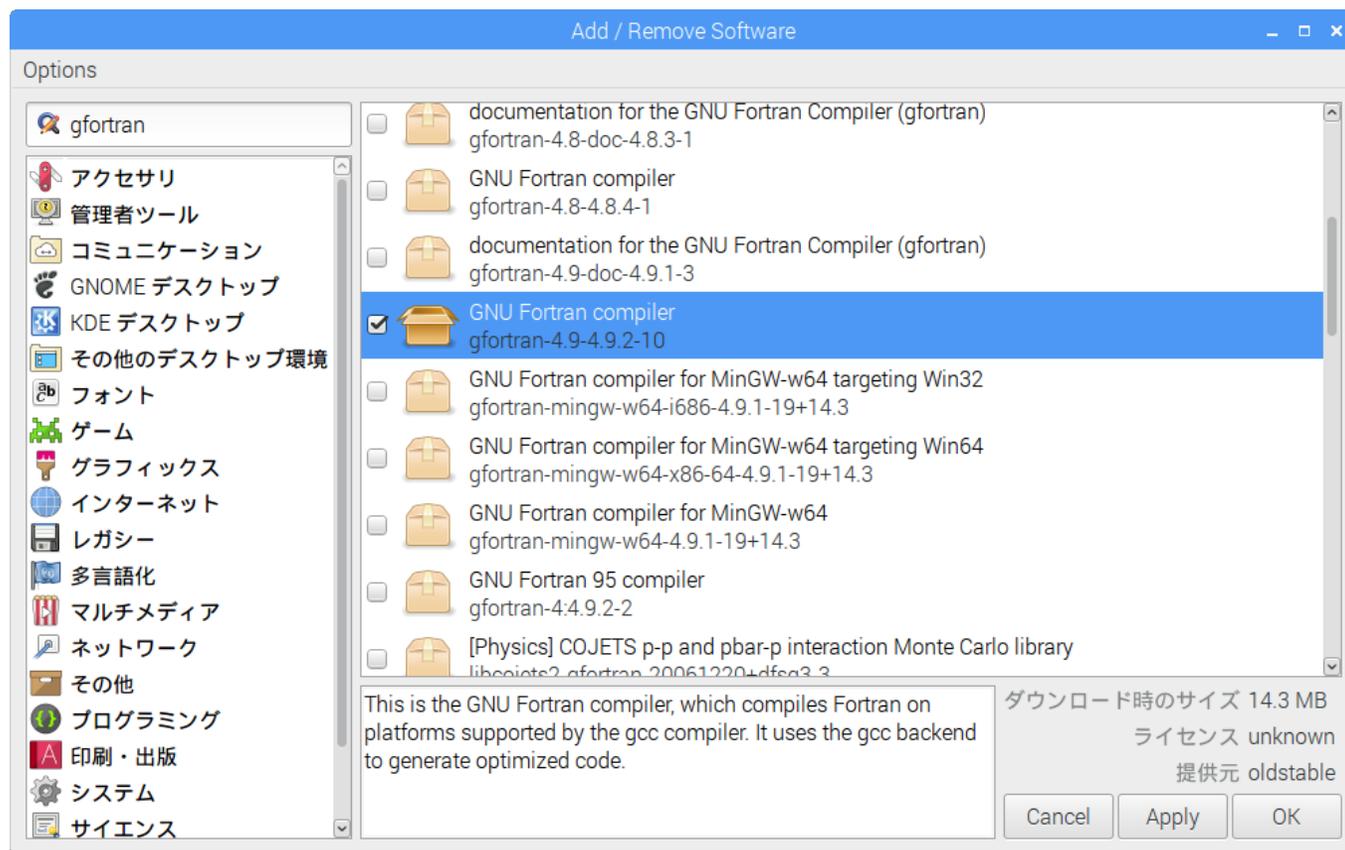
<https://jyn.jp/raspbian-setup/>

開発環境の設定

この時点で、gcc 4.9.2が入っている。（2017年7月現在）

gfortranの導入

gfortranはGUI環境のAdd/Remove Softwareから入れる。



/usr/binの中で
gfortran-4.9という名前で入っ
ているので

```
In -s gfortran-4.9 gfortran
```

とシンボリックリンクを張ったほ
うがよい。

EGS5の設定

EGSのサイトから、ダウンロードして展開。

egs5runの編集

一行目の#!/bin/shを#!/bin/bashに変える。

変更しなかった場合、コンパイルまではできるが、このシェルスクリプトで実行できない。

```
/home/pi/egs5/egs5run: 437: /home/pi/egs5/egs5run: time: not found  
と表示され止まる。
```

(あとは適宜自分の環境を設定する。コンパイラはgfortranを使う。)

egs5/include/egs5_h.fの編集

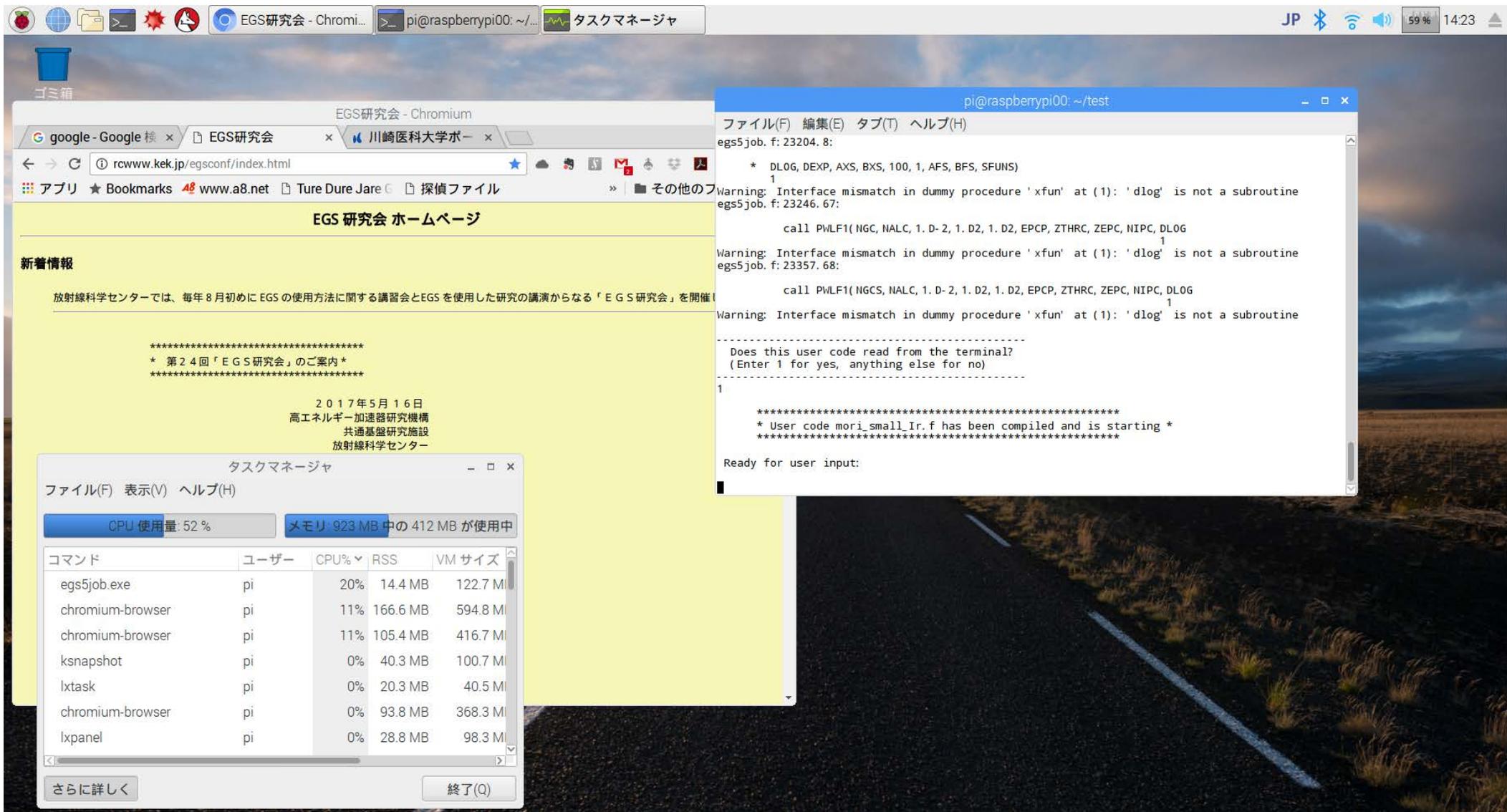
メモリが1GBしかないため、

parameter (NBFIT = 2)

parameter (NEXFIT = 2)

に変更。(よってGS多重散乱は使えない。)

これで動く(はず)。



EGS5の実行画面

EGS5をRaspberry Pi上で実行させるためにやっではいけないこと、できないこと。

g77を走らせるために古いバージョンのgccをソースからコンパイルする。

```
gcc-3.4.6.tar.gz  
configure --prefix=/usr/local/gcc-3.4.6 --disable-bootstrap  
--with-fpu=vfp --with-float=hard
```

```
Makefile:1273: recipe for target 'crtbegin.o' failed
```

makeで必ず失敗する。

fort77を入れて実行する。

fort77とは・・・f2cを介し、Fortran77プログラムをCに変換する。
そしてそのままコンパイルし、実行ファイルを作る。

EGS5プログラムのコンパイラの段階で失敗する。

ここまでで、EGS5は、Raspberry Pi上で実行できた。ここから先の展望（野望）は・・・

これを複数台使った卓上**クラスターパソコン**の製作



西永俊文作



Simon J. Cox作
Raspberry Pi and Lego Supercomputer



Josh Kiepert作
Pri Cluster

そもそもの「並列計算」の発端は・・・・・

ネット上で「cuda aware open mpi」というものに出くわしたことが発端です。
(GPGPUをMPIで制御できるといったもの)

- Geforceでも使えるが、倍精度の計算ならば、TESLAがよい。ただし、EGS5を使うならば、元の計算コードをCUDA用に変更する必要がある。
- 現在、私はXeon Phi (knights landing)の計算機を使用しています。
(産総研 清水様とのプライベートコミュニケーション)



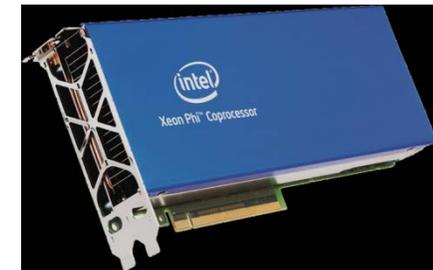
Rakuten キーワードから探す 商品価格ナビで 🔍 買い物かご お知らせ myクーポン

商品価格ナビ ランキング 新製品市場 ご利用ガイド

サービスを使えば使うほどポイントUP! いつでも毎日 ポイン

商品価格ナビトップ > パソコン・周辺機器 > PCパーツ > グラフィックボード > NVIDIA Tesla K80 NVTK80-

NVIDIA Tesla K80 NVTK80-P Passive
総合評価 ★★★★★ 0.00 (0件)
最安値 **753,936円** 送料無料
店: ナノズ 楽天市場店 ★★★★★ 0.00 (0件)
発送日: 1~2日以内に発送予定 (店舗休業日を除く)
その他のショップ: 新品(8) 753,936円~
メーカー: アスク 発売日: - JAN: 4537694226523



スーパーコンピュータなどのHPC向けプロセッサ、システム価格は税込864,000円から

MPIの実行 . . . 実行できることを確認。

具体的方法 **MPICHをソースコードで取ってくる。**

```
mpich-3.2.tar.gzを展開。（2017年7月現在）  
configure --prefix=/usr/local/mpch-3.2  
Sudo make  
Sudo make install
```

環境変数の設定

```
export PATH=...../usr/local/mpich-3.2/bin:$PATH
```

EGS5mpiの設定

egs5mpirunの編集

```
#!/bin/bashに変更
```

mpif77はmpifortにシンボリックリンクがはられているのでそのままよい。

```
OPT_LEVEL=O2 3以上にするとエラー
```

```
CFLAGS="-fno-automatic -finit-local-zero"を
```

```
CFLAGS="-fno-automatic -finit-local-zero -fno-range-check"にした。
```

（イベント数100億でエラーが起こるため）

EGS5をRaspberry Pi上で実行させるためにやっではいけないこと、できないこと。

```
Open MPI はソースからインストールできない  
openmpi-2.1.1.tar.gz  
configure --prefix=/usr/local/openmpi-2.1.1  
Makefile:1743: recipe for target 'atomic-asm.lo' failed
```

GUI環境のAdd/Remove Softwareからのmpich-3.1-5のインストールはお勧めしない。

mpirunやmpiexecは入るが、mpicc、mpifortなどのコンパイラが入らないため。

PC /AT互換機 (PC) との比較

PC/AT互換機スペック

CPU : Core i7 6700K (オーバークロック
4.5GHz)

Memory : 32GB

HDD : 2TB (mirror raid構築)

OS: Windows 10 Pro

VMware(R) Workstation 12 Player Ver. 12.5.7

Memory: 4GB

HDD: 20GB

OS : Scientific linux Ver.6.3

Compiler: g77 Ver. 3.4.6 (最適化オプション -O)

MPI: Open MPI Ver. 1.6.2 (最適化オプションなし)



自作機

OS: Windows 10 Pro (MS-DOS)
Compiler gfortran Ver. 4.9.2 (最適化オプション -O)

Intel® Math Kernel Library Benchmarks(FLOPS測定)

Intel(R) Optimized LINPACK Benchmark data

Current date/time: Wed Jul 12 10:30:14 2017

CPU frequency: 4.529 GHz
Number of CPUs: 1
Number of cores: 4
Number of threads: 4

Performance Summary (GFlops)

Size	LDA	Align.	Average	Maximal
1000	1000	4	164.8195	176.6230
2000	2000	4	167.4113	173.0733
3000	3000	4	192.0482	193.3342
4000	4000	4	202.8230	204.6895
5000	5000	4	201.7475	212.2730
10000	10000	4	218.4498	221.5161
15000	15000	4	220.2758	222.4337
20000	20000	4	219.5442	219.8963
<u>25000</u>	<u>25000</u>	<u>4</u>	<u>220.5971</u>	<u>220.8444</u>
30000	30000	4	219.3668	219.3668
35000	35000	4	220.4812	220.4812
40000	40000	4	205.4967	205.4967

本体 :
220GFLOPSと判定

理論値 : (Wikipediaより)
16 FLOPS/Clock × 4.5GHz × 4コア
= 288GFLOPS

Intel(R) Optimized LINPACK Benchmark data

Current date/time: Wed Jul 12 16:19:17 2017

CPU frequency: 4.521 GHz
Number of CPUs: 4
Number of cores: 4
Number of threads: 4

Performance Summary (GFlops)

Size	LDA	Align.	Average	Maximal
1000	1000	4	97.3288	100.5634
2000	2000	4	98.1424	100.2392
5000	5008	4	106.1645	107.3601
10000	10000	4	106.0124	106.4934
<u>15000</u>	<u>15000</u>	<u>4</u>	<u>110.3614</u>	<u>110.3849</u>
18000	18008	4	110.1393	110.3667
20000	20016	4	103.7467	104.1593

VMwarePlayer+Scientific Linux :
110GFLOPSと判定 (? ? ? ?)

※このMacBook air (Core i5
(2core) 1.3GHz, memory 4GB)

Intel(R) Optimized LINPACK Benchmark data

Current date/time: Wed Jul 19 12:16:21 2017

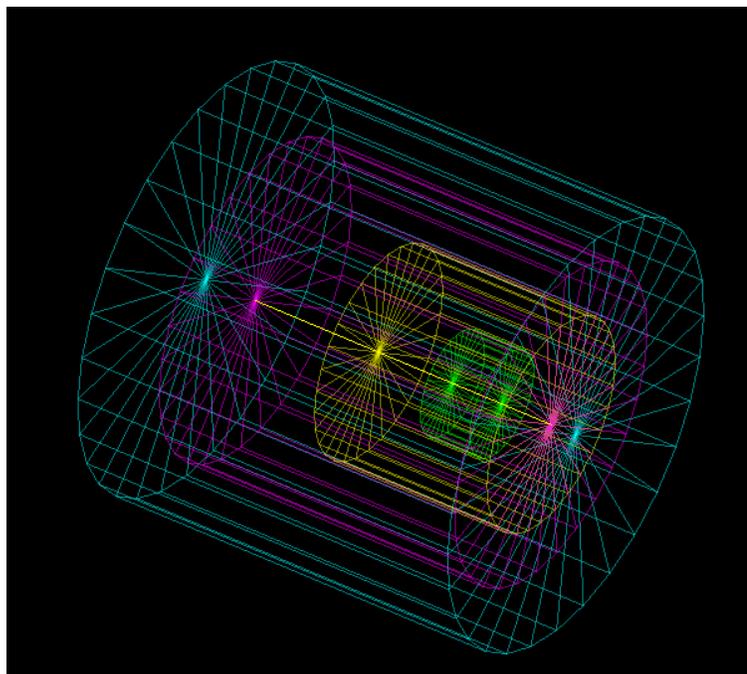
CPU frequency: 2.599 GHz
Number of CPUs: 1
Number of cores: 2
Number of threads: 2

Performance Summary (GFlops)

Size	LDA	Align.	Average	Maximal
15000	15000	4	43.4464	43.4464
14000	14008	4	45.5212	45.5352
13000	13000	4	48.4429	49.3847
<u>12000</u>	<u>12008</u>	<u>4</u>	<u>48.5461</u>	<u>49.8232</u>
11000	11000	4	44.7740	48.9055
10000	10008	4	48.2093	48.4916
8000	8008	4	49.3595	49.5665
6000	6008	4	46.2490	48.5232
1000	1000	4	27.4195	33.1489

50GFLOPSと判定

テストしたプログラム



ucsource.f

単に単一エネルギーのビームを
発射するのみ。集計はしない。

※PCのWindowsとVMware+linux
は、ucnaicgv.fとucphantomcgv.fも比較

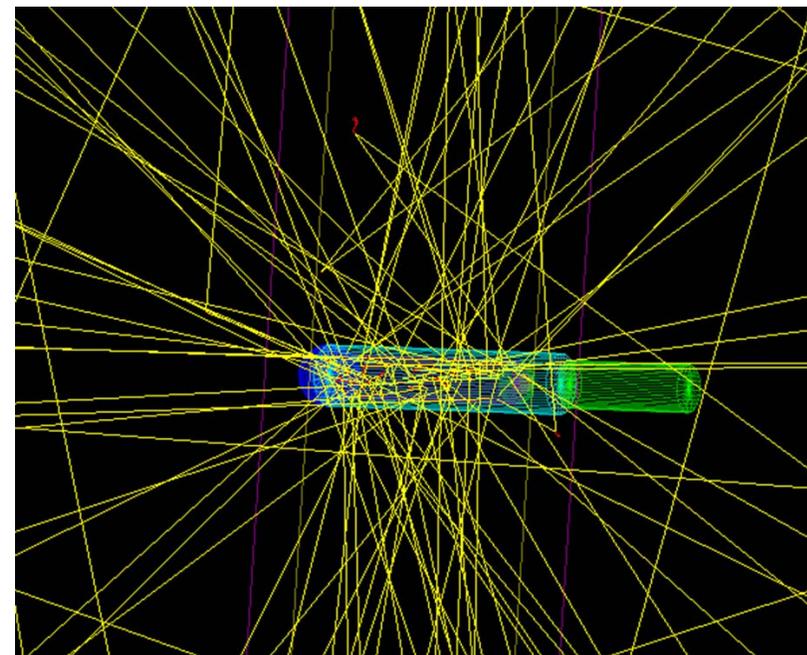
Raspberry Pi 3 B

1GB RAM

OS: Raspbian (Kernel version 4.9)

コンパイラ: gfortran ver.4.9.2.10 (最適化オプション -O)

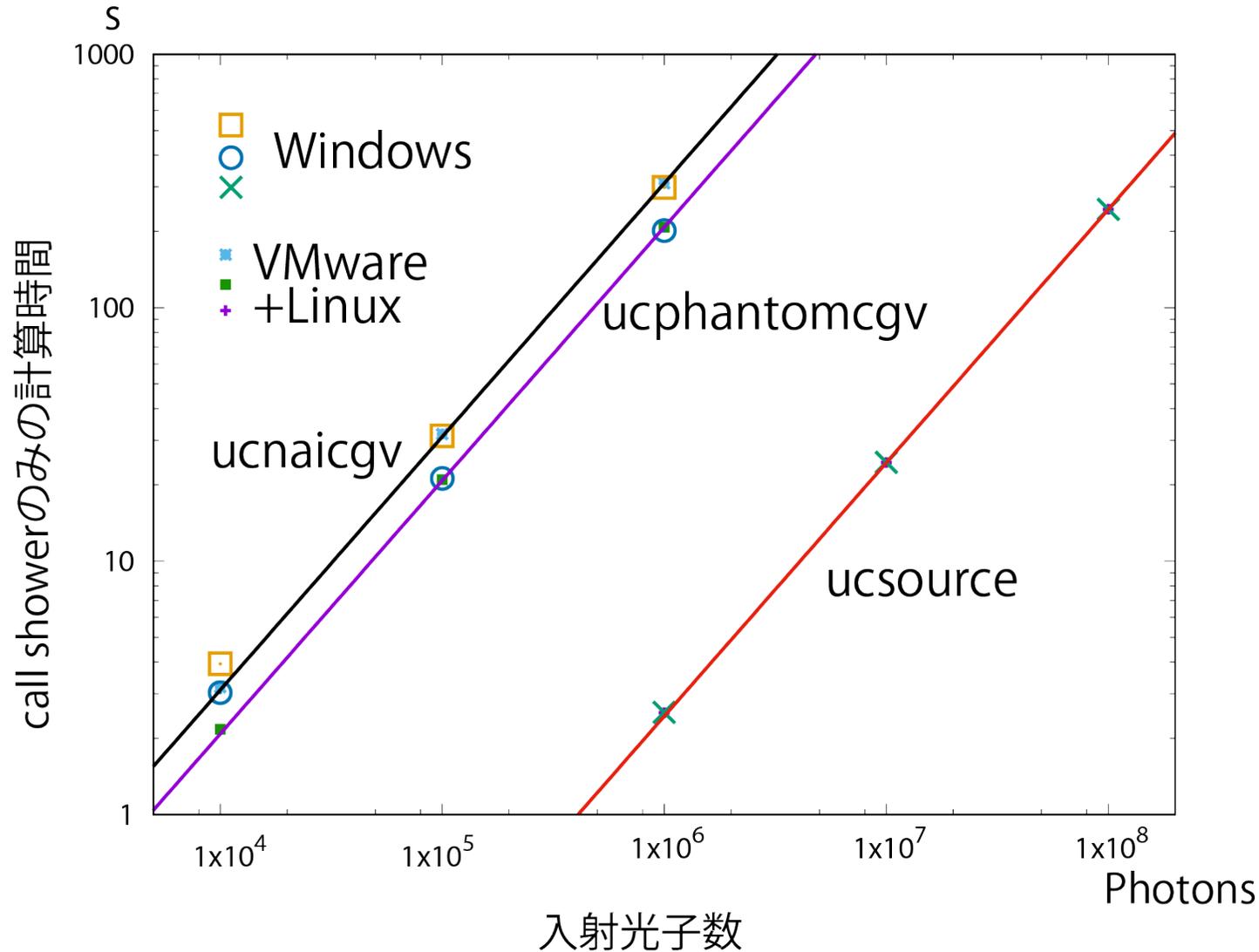
MPI: MPICH Ver.3.2 (最適化オプション -O2)



^{192}Ir 小線源

Irのスペクトルでエネルギーを
ランダムに振らす。
距離ごとに吸収線量を集計。

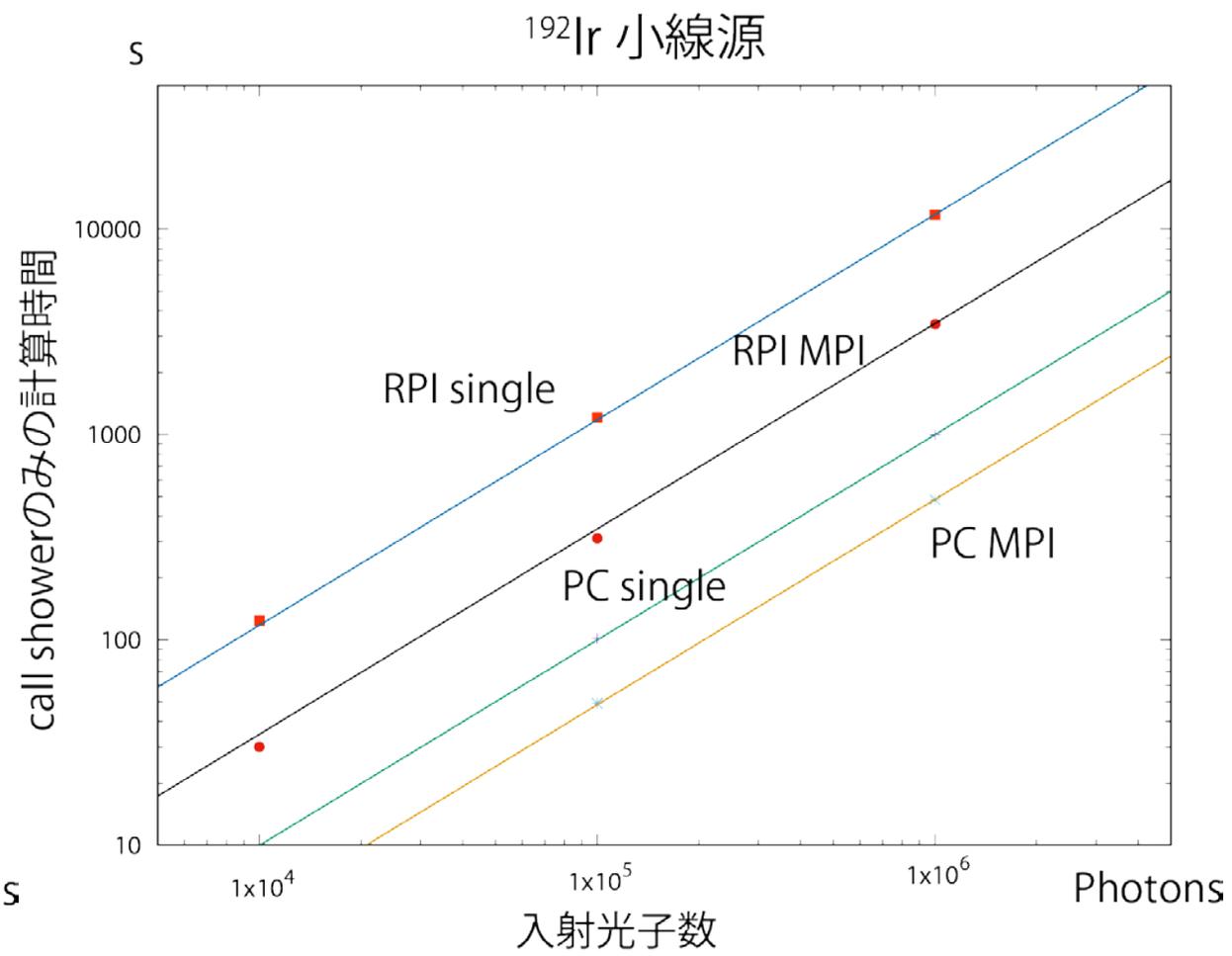
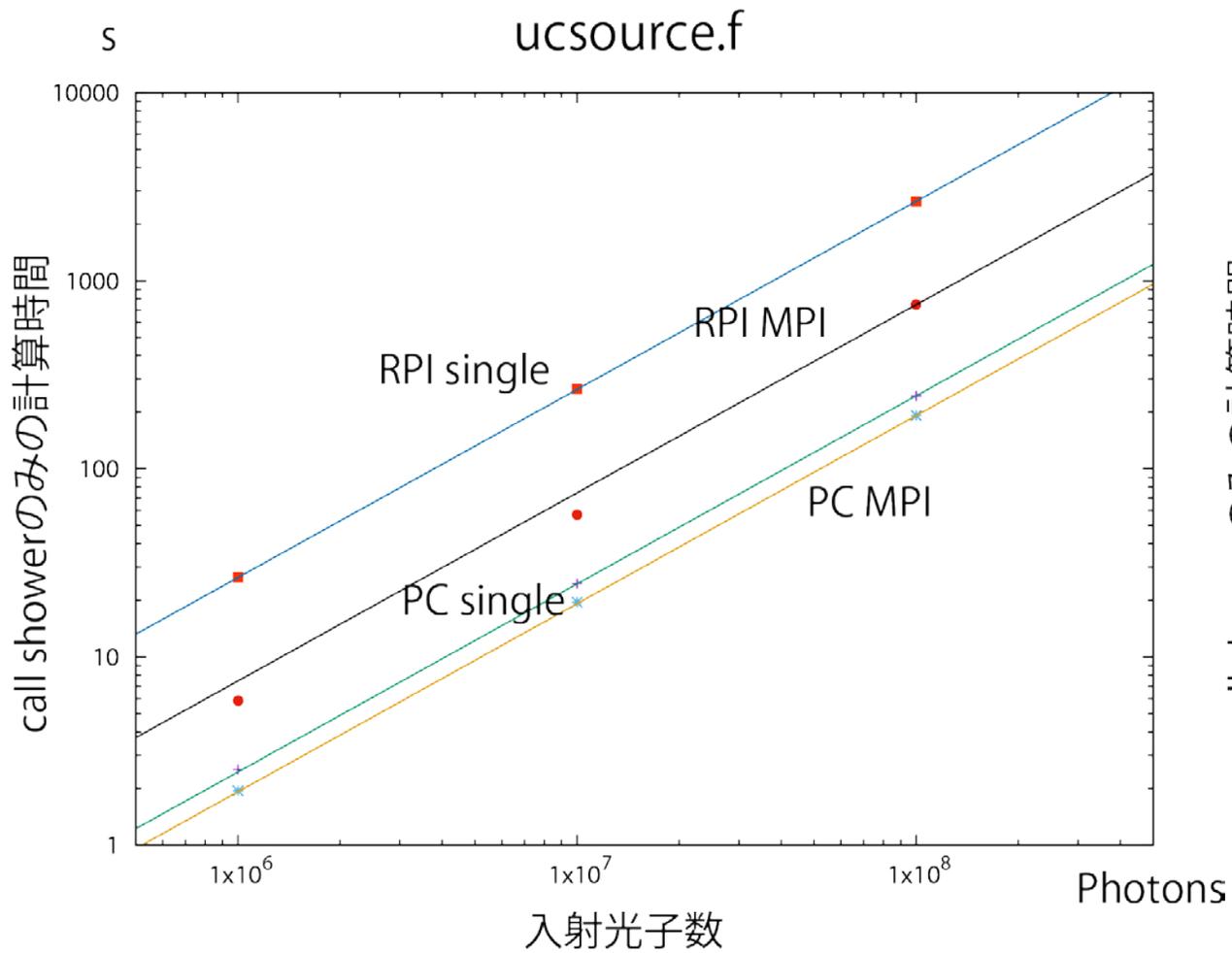
Vmware上でプログラムを走らせると計算能力は半分になるのか。



Windowsからコマンドプロンプトで実行するのも
Vmware+LinuxもEGS5の
計算速度はほぼ同じ

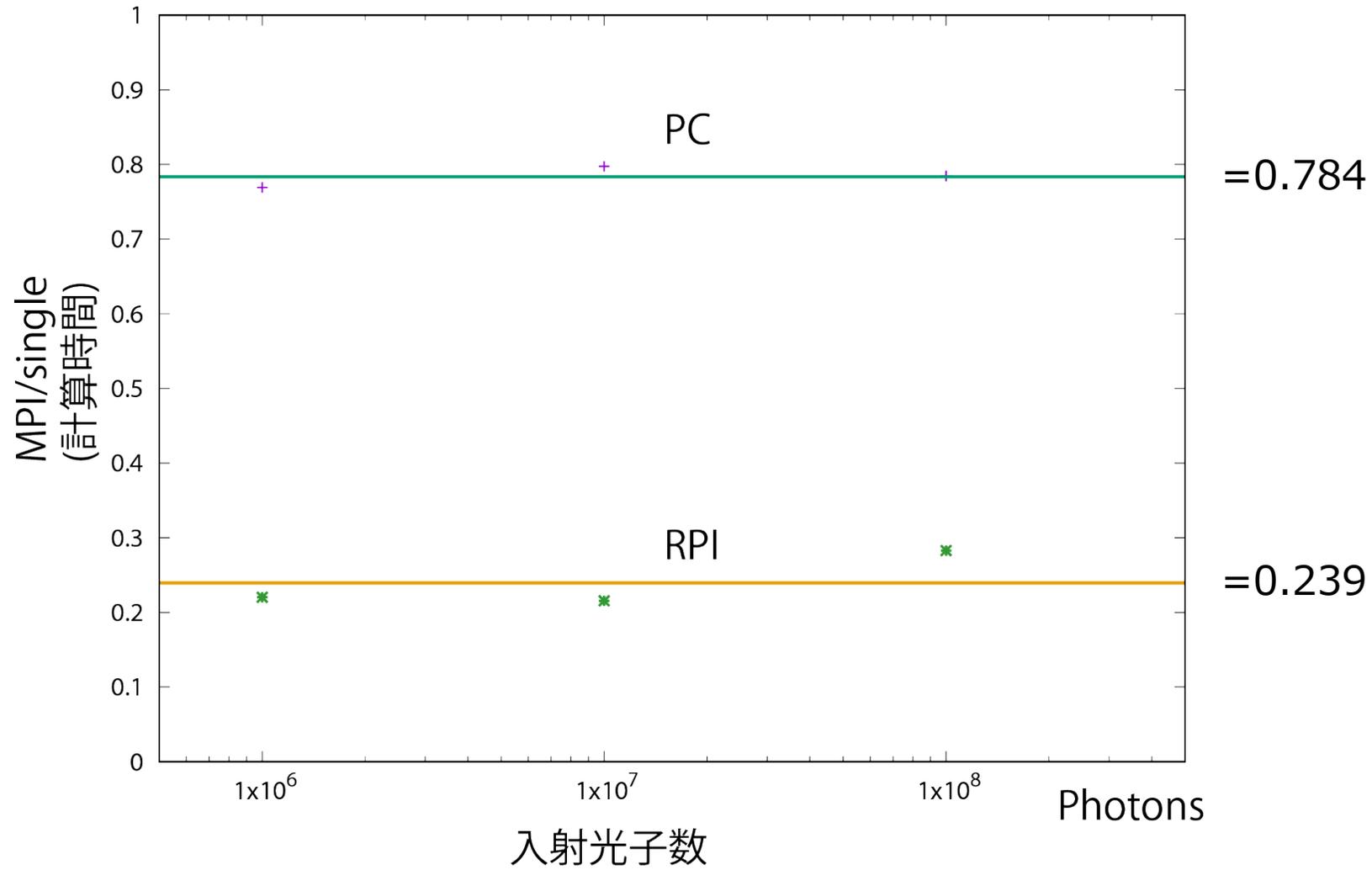
EGS5ベンチマークテストより
Vmware+Linuxも**220GFLOPS**

PCはVMware+Scientific linux

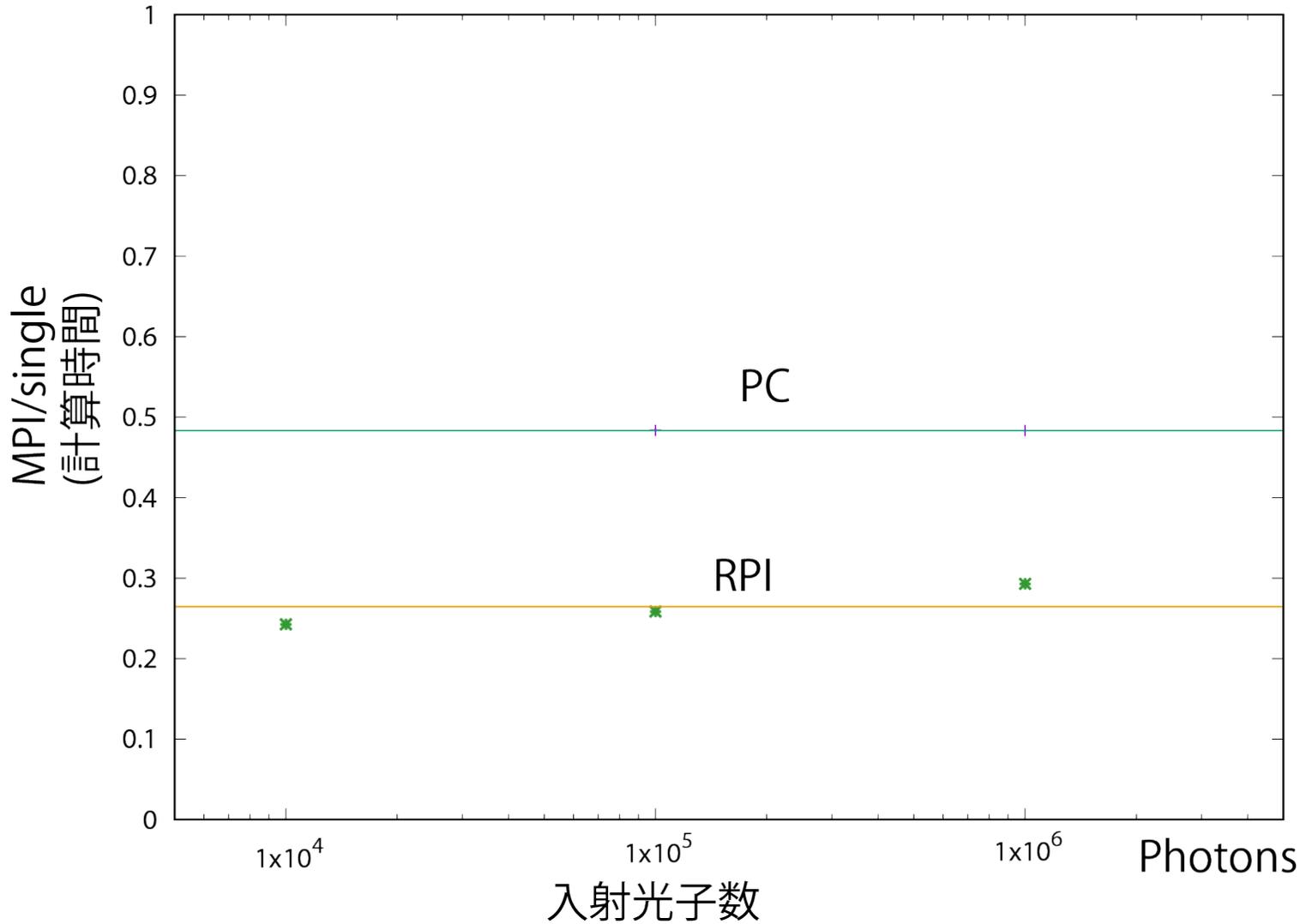


MPIの効果

ucsourceでのsingle coreとMPI(4 core)の比



^{192}Ir でのsingle coreとMPI(4 core)の比

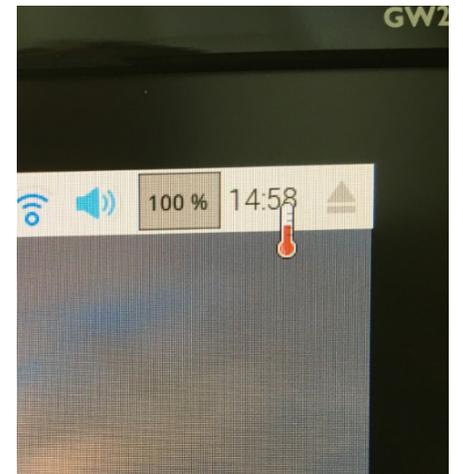


=0.483

=0.264

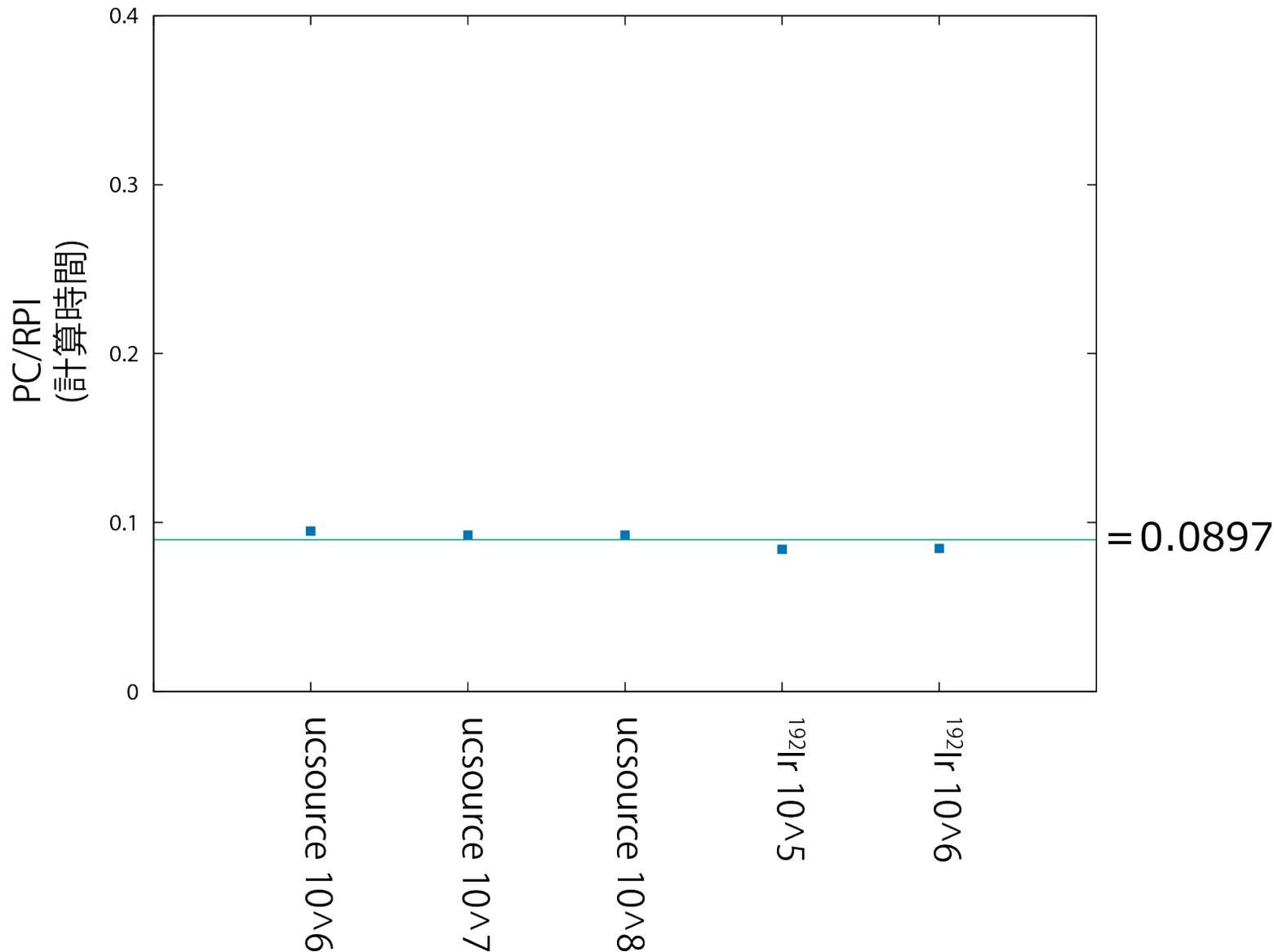
PCが1/4にならない要因

- Windows 10
- Vmware
- Linux
- Open MPI (ver. 1.6.2)
- Hyper Threading
- ? ? ? ? ? ? ? ? ? ?



Single coreでの比較

single coreでのPCとRPIの比



Raspberry Piの計算能力の推定

$$220 \times 0.0897 = 19.7 \text{GFLOPS}$$

約 20GFLOPS

理論値 (wikipediaより)

ARM Coretex-A57 44.8GFLOPS
(4FLOPS/CLOCK×2.8GHz×4コア)

この値を参考に

$$\underline{4 \text{ FLOPS/CLOCK}} \times 1.2 \text{GHz} \times 4 \text{コア} = 19.2 \text{GFLOPS}$$

消費電力



3.982W \doteq 4W

構築費用

Raspberry Pi 3 MODEL B	¥4896
TOSHIBA microSDHC 16GB Class10	¥998
エレコム エコ USBケーブル U2C-JAMB03BK	¥513

	¥6407
ELECOM 電源タップ 雷ガード 一括スイッチ マグネット 付き 抜け止めコンセント 3P 7個口 2m T-Y3A-3720WH	¥1208
Anker PowerPort 4 (40W 4ポートUSB急速充電器) A2142122	¥2399

	¥3607
合計	¥10014

クラスタパソコンを構築すると考えると.....



24ポートが最もお手頃価格

クラスタ同士もつなげることを想定すると22ノード

Raspberry Pi 3 MODEL B	22× ¥4896
TOSHIBA microSDHC 16GB Class10	22× ¥998
エレコム エコ USBケーブル U2C-JAMB03BK	22× ¥513

22× ¥6407
¥140954

ELECOM 電源タップ 雷ガード 一括スイッチ マグネット 付き 抜け止めコンセント 3P 7個口 2m T-Y3A-3720WH	¥1208
Anker PowerPort 4 (40W 4ポートUSB急速充電器) A2142122	6× ¥2399

¥15602

ネットワーク関係

NETGEAR ギガ24ポート GS324-100JPS(ネットワークハブ)	¥9980
iBUFFALO 0.5m BSLS6SU05WH2(LANケーブル)	22×¥304

¥16668

¥140954

¥15602

¥16668

合計 ¥173224

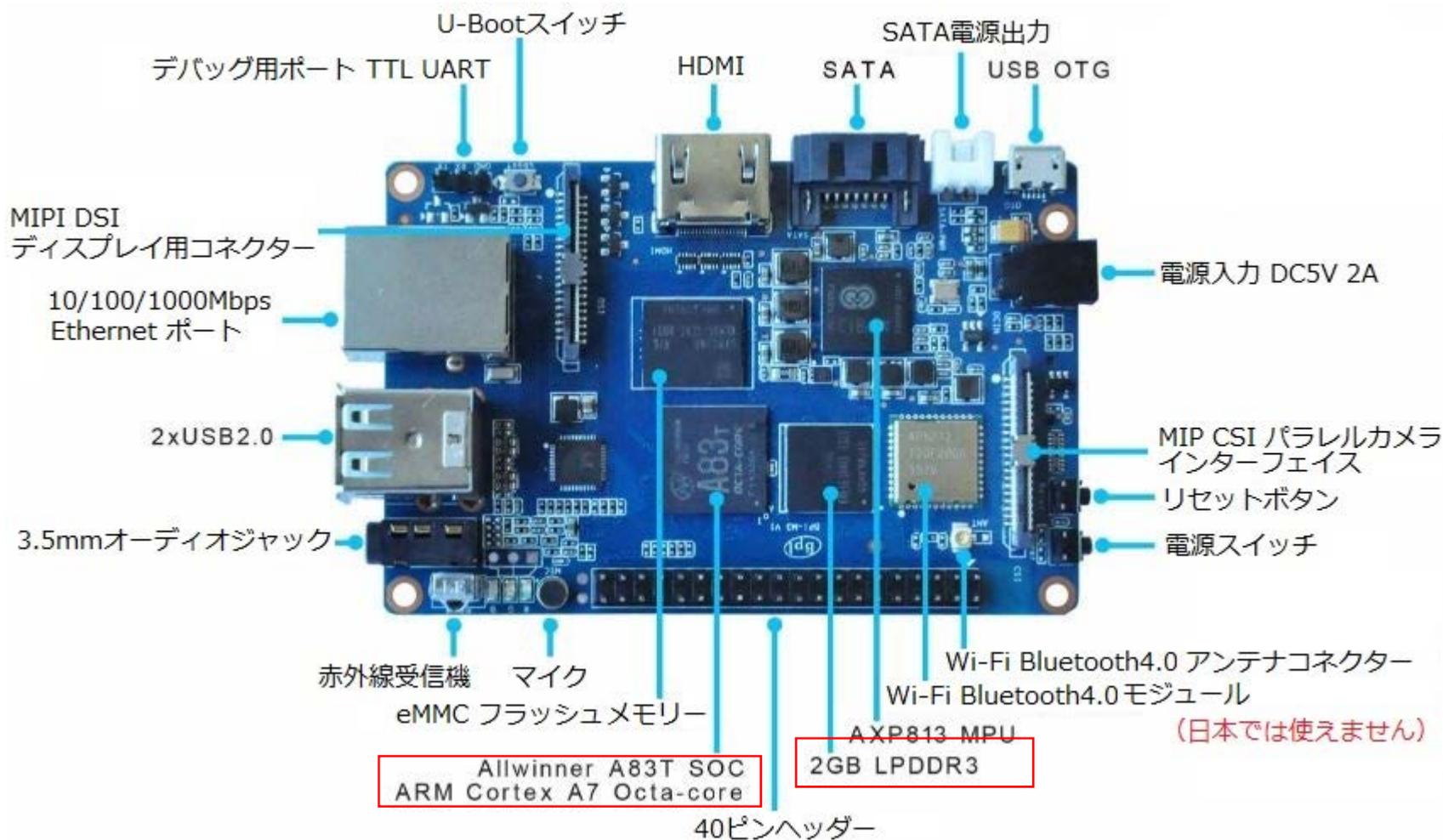
スペック

440 GFLOPS

消費電力 100W以下(22×4W)

今現在、様々なシングルボードコンピュータが存在する。

BPI-M3 Banana PI M3 オクタコア・シングルボードコンピュータ



CPU:
A83T ARM Cortex-A7 Octa-core
1.8GHz

$$8/4 \times 1.8 / 1.2 = 3$$

計算上では3倍能力が
上がる。

値段: ¥13824

UDOO X86



価格： \$165 (Advanced plus)

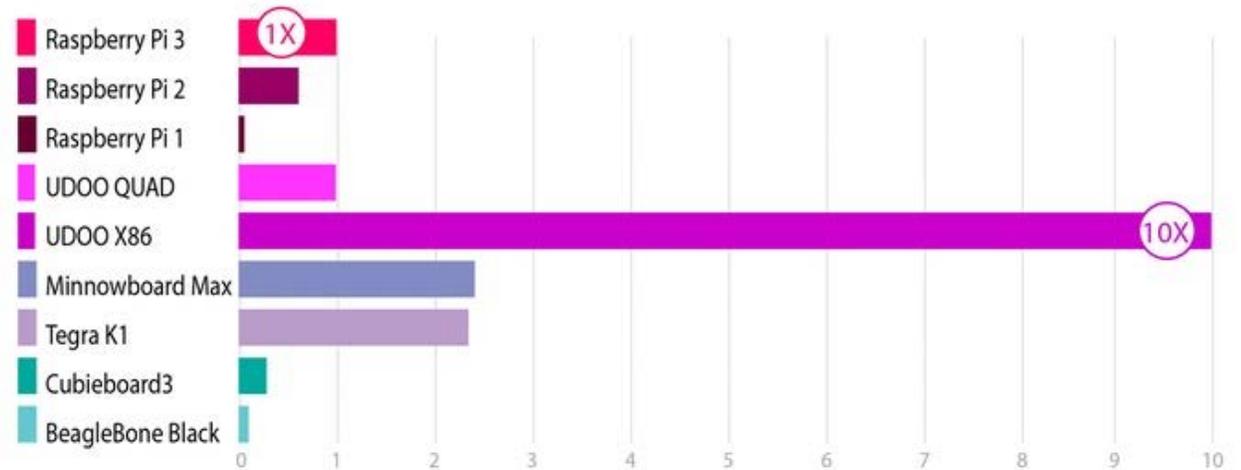
= ¥18500

CPU Intel Celeron N3160 2.24 GHz(4core)
Memory 4GB

10 TIMES MORE POWERFUL THAN RPI3

SYSBENCH TEST MULTI-THREAD CPU

Speed-up over Raspberry Pi3 execution time (higher is better)



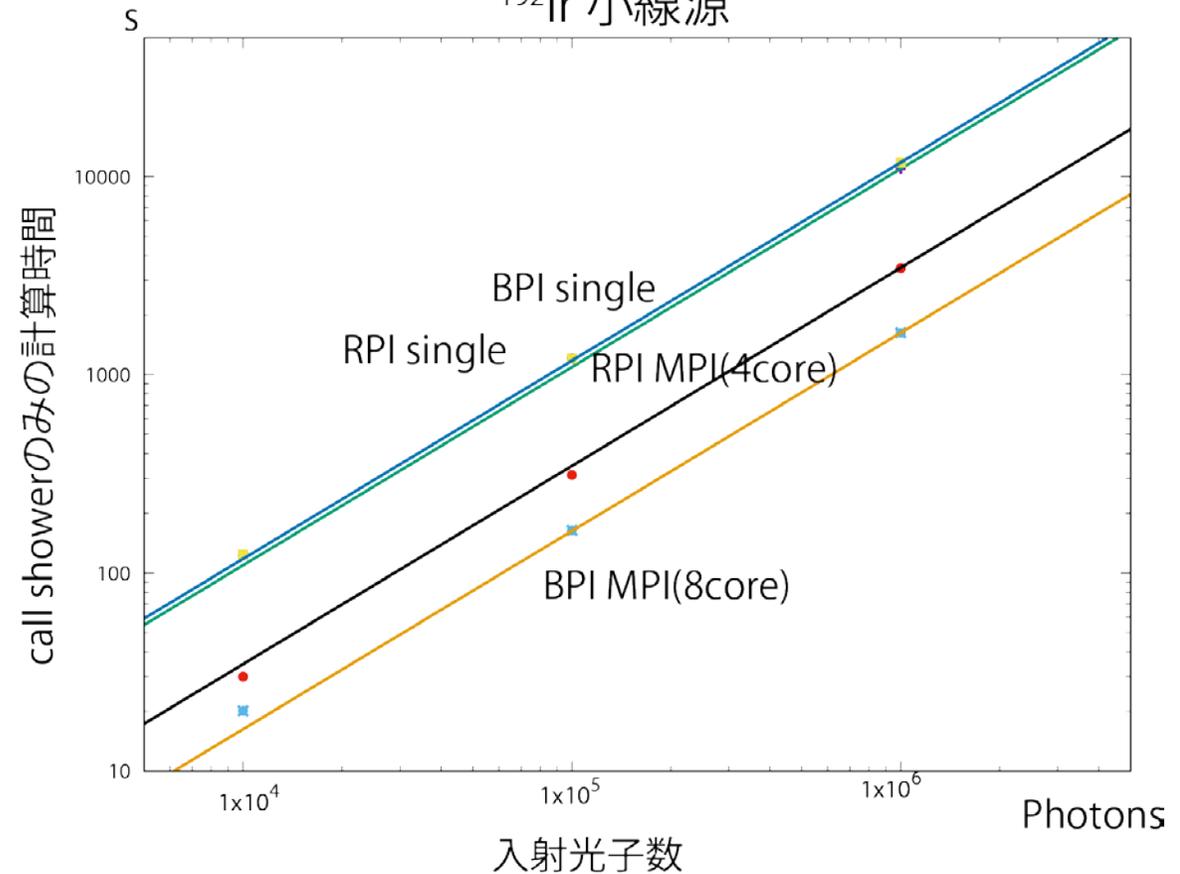
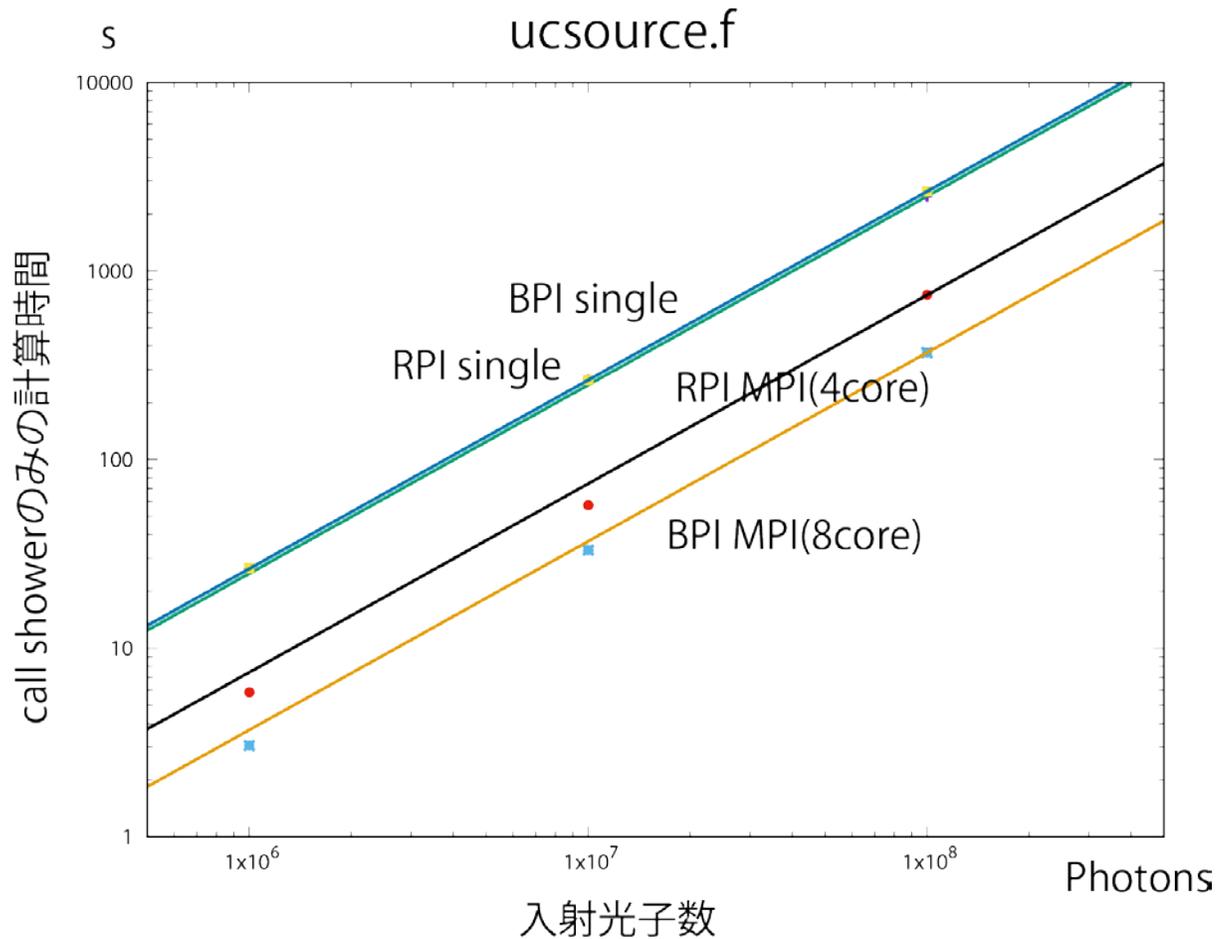
Raspberry Pi 3の10倍????

BPI-M3 Banana PI M3

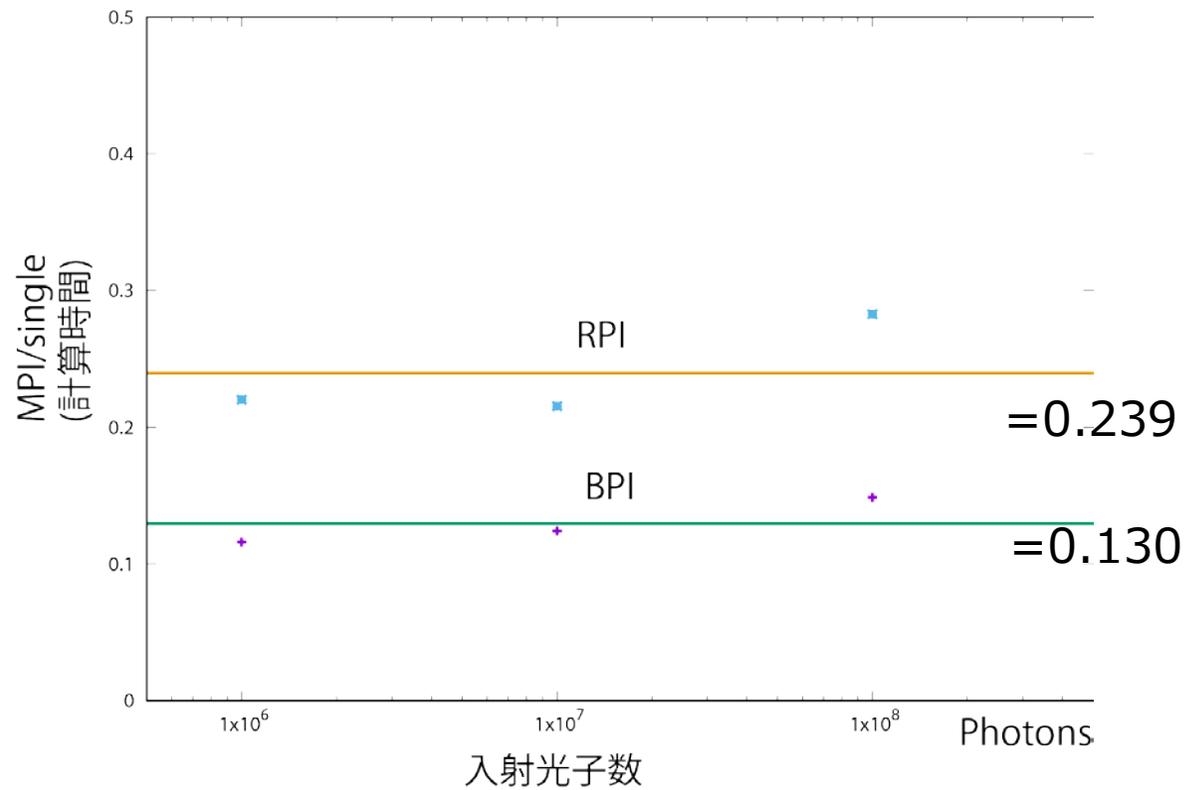
egs5/include/egs5_h.fは編集する必要なし
parameter (NBFIT = 128)
parameter (NEXFIT = 32)
でよい。



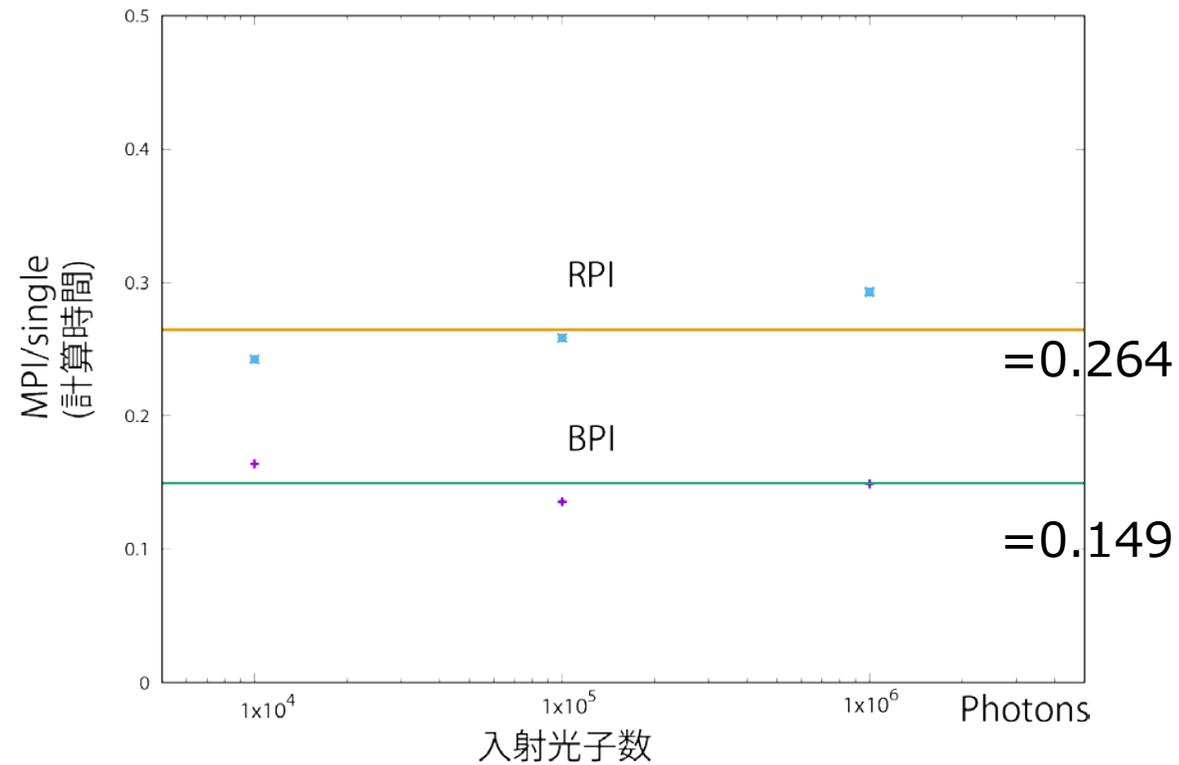
^{192}Ir 小線源



ucsourceでのsingle coreとMPIの比



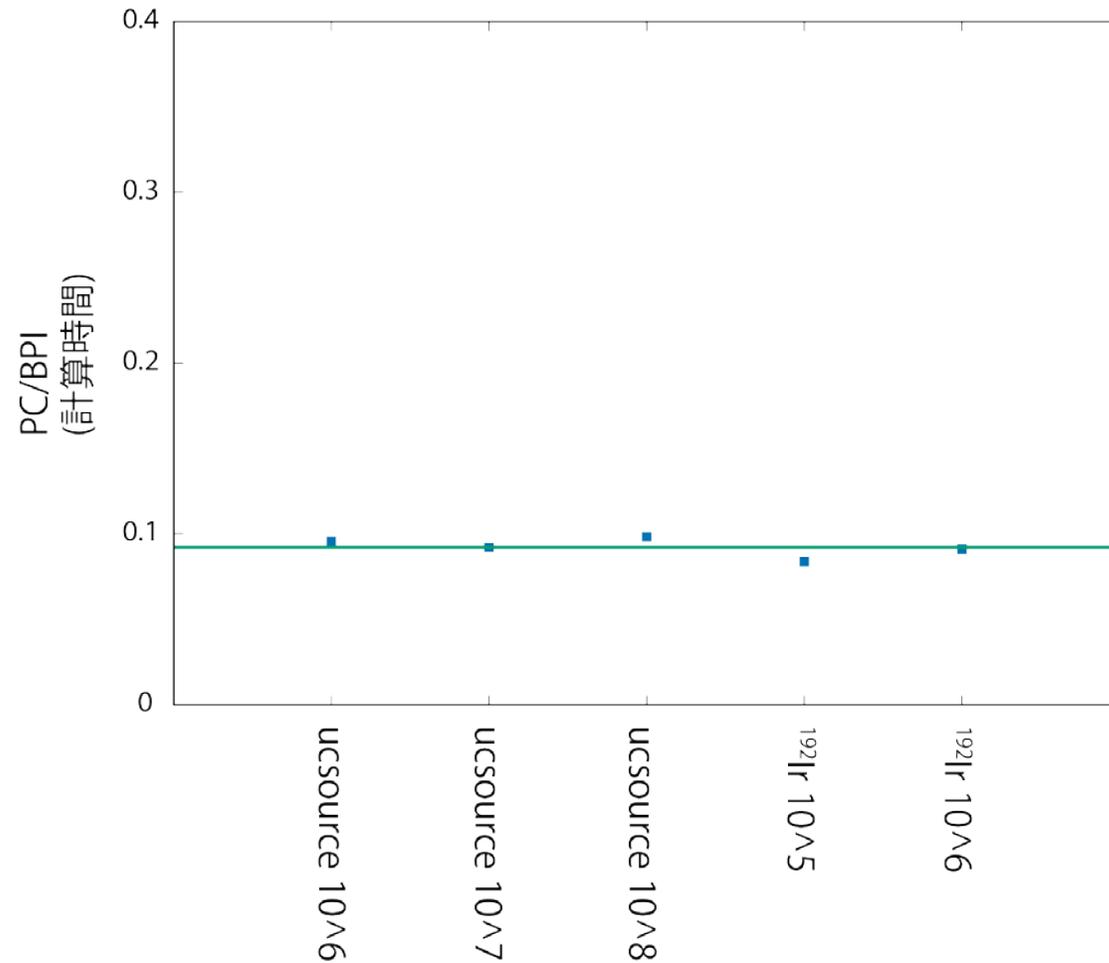
^{192}Ir でのsingle coreとMPIの比



消費電力 5.2W



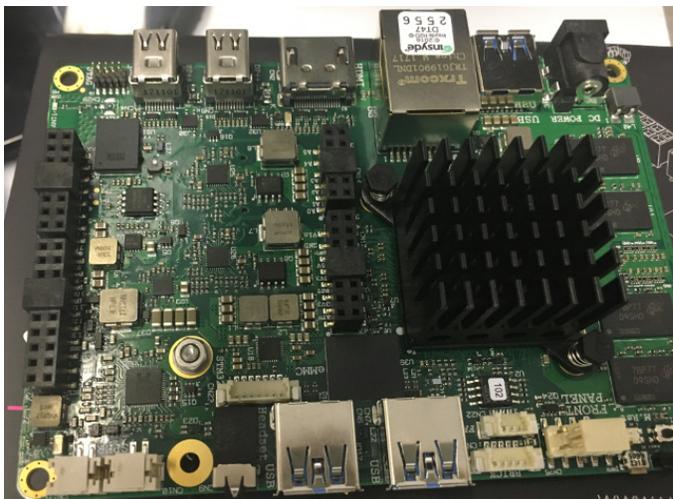
single coreでのPCとBPIの比



= 0.0923

$0.0923 \times 220 \times 8/4 =$
40.6 GFLOPS

UDOO X86



注意事項

電源が12V 単に12V 出ているだけではダメ



このような5V→12V に変換する
ようなものを使っても
まず、OSのインストールがで
きない。

be sure it's a **12V 3A** and
it's properly working.

1コアあたりのEGS5の計算能力はRPIの約2倍
(10倍ではない)

デメリット

- 1ボード当たりの値段は3万円を超える。
- 電力を食う。
- デカい。(RPIやBPIの面積が約2倍)

まとめ

Raspberry Pi 3 Model Bで、EGS5が動くことを確認した。

- 64bit Quad Core 1.2GHz ARM Cortex-A53
- 1GB RAM
- OS: Raspbian (Kernel version 4.9)
- コンパイラ: gfortran ver.4.9.2.10

MPIを組み込みことができ、EGS5の並列計算ができることを確認した。

- MPI : MPICH-3.2

Raspberry Pi 3 Model Bの計算能力 20 GFLOPS
消費電力 約 4W

Banana Pi M3でも動作を確認

計算能力 40 GFLOPS
消費電力 約 5W