

RaspberryPi の浮動小数点演算性能からみる CPU アーキテクチャの特徴

2017/02/25

佐藤周行

1. この計測の目的

もともとの目的である教育を別にして、RaspberryPi がセンサー+アルファとして IoT デバイスとしての役割を期待されるようになった。IoT デバイスとしては、計算能力に大きなものを期待しないのはもちろんである。実際、RaspberryPi のシステム構成として特に CPU (ARM) については通常のシステムとはかなり構成が異なる。

本稿では、RaspberryPi の計算能力を、浮動小数点演算性能から計測し、システム全体としての性能の特徴を明らかにすることを目的とする。

2. 計画

表 1 の環境に対して

- A. 行列積による性能計測 (Compute Intensive)
- B. 行列和による性能計測 (Bandwidth Intensive)

を行う。

表 1 計測環境

システム	RaspberryPi 3 Model B (1.2GHz×4)
コンパイラ	gcc 4.9.2 (Raspbian 4.9.2-10)

2. 1 コンパイラについての注意

Raspbian 提供の GCC コンパイラは最適化オプションの有無により、性能が大きく変化することがわかった。

表 2 に、行列積 (アンローリングなし。N=400) を用いたコンパイラオプションによる性能の差を示す。

表 2

指定なし	39.0 MFlops
-O1	97.4 MFlops
-O2	164.6 MFlops
-O3	165.1 MFlops
-O4	166.1 MFlops

Intel 系システムの Linux 上の gcc では、指定なしが-O2扱いだったはずだが、RaspberryPi では構成が異なっている。以下、最適化オプションとして-O4を指定してコンパイルすることにする。

2. 2 計測計画とその結果

A. 行列積

行列積は、従来型の CPU アーキテクチャに従えば、キャッシュを効率的に利用すれば **Compute Intensive** な計算ができることで有名である。

ここでは、いわゆる **IKJ** 順にループを構成する。

1. キャッシュの効果をみるために、ナイーブな形(アンローリングなし)で行列のサイズを小から大にスライドさせていって性能を計測する。表 3 に、N=200から1000まで変化させた場合の性能データを載せる。以下、計測は1回のみ。(良い子は真似してはいけない)

表 3 行列積性能 (アンローリングなし)

N	200	400	600	800	1000	1600
MFlops	111.2	159.8	161.0	161.1	159.8	152.4

2. CPU の **Instruction Parallelism** の効果をみるために、ループアンローリングの段数を変えて性能を計測する。表 4 に、N=400として、アンローリング段数を2から16まで変化させた場合の性能データを載せる。

表 4 行列積性能 (アンローリングの効果)

Unrolling factor	2	4	8	16
MFlops	251.8	279.0	401.5	194.7

400MFlops 出たのは、gcc としては上出来ではないだろうか。(悪口)

B. 行列和

行列和で計測できるのは、メモリの **bandwidth** と浮動小数点演算のバランスである。表 5 に、性能を載せる。

表 5 行列和性能

N	200	400	600	800	1000	1600
MFlops	34.0	39.4	44.8	45.5	45.6	46.1

アンローリングをしても効果は見られないはずである。N=400の時の結果を表 6 に載せる。

表 6 行列和性能 (アンローリングの効果)

Unrolling factor	2	4	8	16
MFlops	29.5	40.7	27.6	36.7

3. 観察

A と B より、以下が観察される。アンローリング段数を 2 から 16 まで変化させた場合の性能データを載せる。

1. キャッシュの効果は見られない。行列のサイズが大きくなることでの性能の劣化が観察されない。キャッシュとして小さいものを搭載しているのがわかる。
2. アンローリングの効果は、行列積に対して大きい。これは、**instruction parallelism** が一定の効果を持っていることを示している。ただし、段数を大きくしての性能劣化が行列積で8を境にして見られる。これは **instruction cache** あふれを主とするものが原因として考えられる。

これより、**RaspberryPi** については、現状主流の CPU アーキテクチャとは別のものとして戦略を立てることが求められることが結論付けられた。

4. 結論

本稿では、**RaspberryPi** のシステムの性能を浮動小数点演算から計測した。