

Ampere Altra の性能評価

大島 聡史^{1,a)} 永井 亨¹ 片桐 孝洋¹

概要: CPU の性能は計算機システムの性能を左右する重要な要素である。スーパーコンピュータやデータセンタ、クラウド計算サーバ向けの CPU としては Intel 社の Xeon が多くのシェアを占めてきたが、近年では AMD 社の EPYC や、スーパーコンピュータ「富岳」や「不老」に搭載された富士通社の A64FX など新たな CPU への注目も高まっている。本稿では、ARM アーキテクチャに基づく最新の CPU である Ampere Altra の性能について報告する。A64FX をはじめとするいくつかの CPU とベンチマーク性能を比較し、性能や特徴を評価する。

1. はじめに

大規模な問題や複雑な問題を高速に解くには高い計算性能が必要であり、CPU の性能は PC サーバやスーパーコンピュータの計算性能を大きく左右する重要な要素である。しかしムーアの法則の終焉が近づく今日、半導体の微細化が困難となるにともない CPU の高性能化も難しくなっている。そこで、より高い性能、より高い電力あたり性能を目指した CPU が開発されるようになってきた。特に近年では ARM アーキテクチャに基づく CPU を採用したスーパーコンピュータ「富岳」(以下、「富岳」)[1] が TOP500 などのスパコンランキングで世界 1 位を連続達成したことにより、ARM アーキテクチャに基づく CPU への注目が高まっている。筆者らは名古屋大学情報基盤センターにてスーパーコンピュータ「不老」(以下、「不老」)[2] を運用している。「不老」は特徴の異なる 4 つの計算サブシステムを備えた複合的なスーパーコンピュータであり、その中でも最も高い理論演算性能を有する Type I サブシステムは「富岳」と同じ CPU である A64FX[3] を備えている。そのため筆者らは A64FX の最適化技術について調査・研究を進めており、また A64FX に限らず、ARM アーキテクチャに基づく CPU やその性能にも注目している。

本稿では ARM アーキテクチャに基づく新しい CPU である Ampere Altra[4] の性能について報告する。Ampere Altra は ARM 社が開発したサーバ向けの CPU であり、スーパーコンピュータよりもデータセンタやクラウドサーバとしての利用が主な用途である。しかし非常に多くの計算リソースを必要とするユーザにとっては、スーパー

コンピュータとクラウドサーバはプログラムを実行する環境として同様に候補となったり、比較対象となりえるものである。また、最新の CPU の性能を比較した結果は様々な研究の参考となる。そこで、「不老」の主要な計算サブシステムである Type I および Type II サブシステムの CPU と性能を比較し、結果について論じるものとする。なお本稿では Oracle Cooperation の提供する Oracle Cloud Infrastructure を利用して性能を測定したが、通常サービス前の試用環境の提供を受けて短期間で測定したものであるため、本サービス時に同一の性能となるとは限らない点に留意されたい。

本稿の構成は以下の通りである。2 章では Ampere Altra の特徴と構成について述べる。3 章では高性能計算分野でよく用いられるベンチマークプログラムなどを用いて Ampere Altra の性能を評価し、現世代の他の CPU と比較する。4 章はまとめの章とする。

2. Ampere Altra CPU の特徴と構成

本章では Ampere Altra の特徴と構成を A64FX などの CPU と比較しながら述べる。Ampere Altra を含めたいくつかの CPU の主な性能値を表 1 に、各 CPU の構成の概要を図 1、図 3、図 4、図 2、図 5 に示す。Ampere Altra と EPYC (Rome, Milan) は、いずれも Oracle Cloud Infrastructure の提供するクラウド計算サーバに搭載されたモデルであり、次章の性能評価で実際に利用したものに対応している。スーパーコンピュータ「不老」および搭載された CPU の構成や性能の詳細については文献 [5] や Web ページ [2] を参照されたい。

Ampere Altra は ARM 社が開発したサーバ向けの CPU であり、主にデータセンタやクラウドサーバとしての利

¹ 名古屋大学 情報基盤センター
Chikusa-ku Furo-cho, Nagoya, Aichi 464-8601, Japan
^{a)} ohshima@cc.nagoya-u.ac.jp

表 1 計算機環境一覧

CPU	搭載システム	コア数 (スレッド数)	動作周波数	1CPUあたり理論演算性能	メモリ	ソケット数
Ampere Altra	Oracle Cloud Infrastructure	80 (80)	3.00 GHz	1.920 TFLOPS (80c*3.0GHz*8Flop/cycle)	DDR4 3200 MHz 500 GB/socket 8チャンネル 204.8 GB/s/socket	2
EPYC 7742 (EPYC Gen 2, Rome)	Oracle Cloud Infrastructure	64 (128)	2.25 - 3.4 GHz	2.304 TFLOPS (64c*2.25GHz*16Flop/cycle)	DDR4 3200 MHz 1 TB/socket 8チャンネル 204.8 GB/s/socket	2
EPYC 7J13 (EPYC Gen 3, Milan)	Oracle Cloud Infrastructure	64 (128)	2.55 - 3.5 GHz	2.6112 TFLOPS (64c*2.55GHz*16Flop/cycle)	DDR4 3200 MHz 1 TB/socket 8チャンネル 204.8 GB/s/socket	2
A64FX	スーパーコンピュータ「不老」 Type I サブシステム	48 (48) + アシスタントコア	2.20 GHz	3.3792 TFLOPS (48c*2.2GHz*32Flop/cycle)	HBM2 32 GB (8 GB/CMG) 1,024 GB/s	1
Xeon Gold 6230 (Cascade Lake, CLX)	スーパーコンピュータ「不老」 Type II サブシステム	20 (40) ただしHTTTは無効化設定	2.10 - 3.90 GHz	1.344 TFLOPS (20c*2.10GHz*32Flop/cycle)	DDR4 2933 MHz 192 GB/socket 6チャンネル 140.784GB/s/socket	2

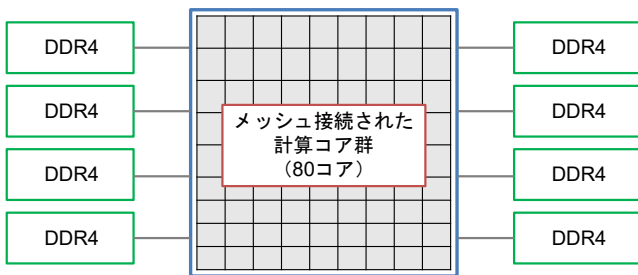


図 1 Ampere Altra の構成概要

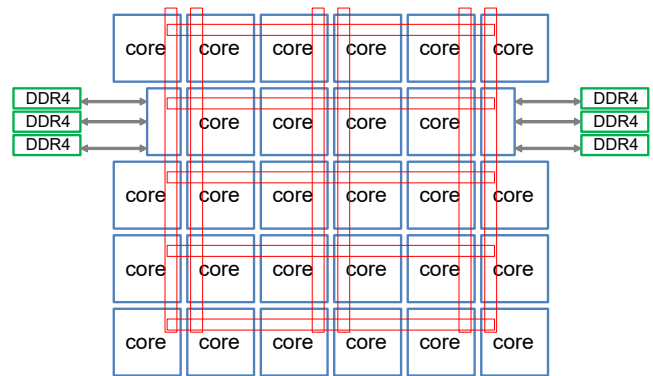


図 5 Xeon (Cascade Lake) の構成概要

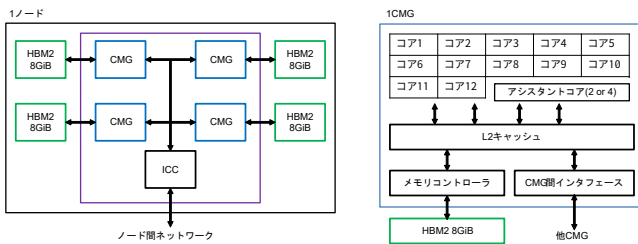


図 2 A64FX の構成概要

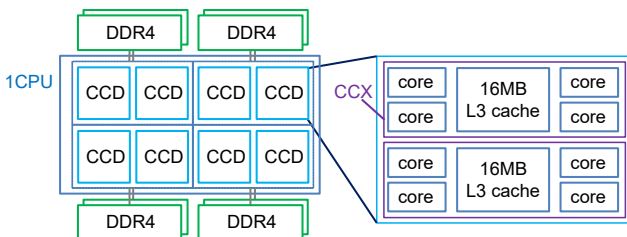


図 3 EPYC (Rome) の構成概要

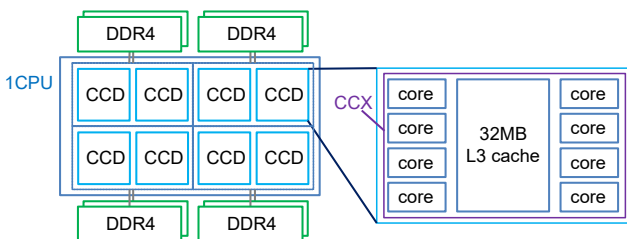


図 4 EPYC (Milan) の構成概要

用が想定されている。Ampere Altra のコア数は1ソケットに80(64コアのバリエーションも存在する)であり、A64FX

をはじめとする現在の主要なCPUと比べて多い。近年の高性能計算(スーパーコンピュータ)向けメニーコアCPUとしてはIntel社のXeon Phiプロセッサ(Knights Landingアーキテクチャ)が知られているが、その最大コア数が72であることから、Ampere Altraのコア数が非常に多いことがわかる。ただしKnights Landingが1コアあたり4スレッド同時実行可能であるのに対してAmpere Altraは1コアあたり1スレッド実行のみ対応している。また、Ampere Altraと同じARM系のCPUであるA64FXは48の計算コアに加えて2つまたは4つのアシスタントコアを搭載することでOSや通信の処理を主演算から切り離しているが、Ampere Altraにはそのようなコアや機能は搭載されていない。

CPU上の計算コア同士および計算コアとメモリとの接続方法は性能に大きな影響を与える重要な要素であり、それぞれに特徴がある。

Ampere Altraの構成概要を図1に示す。Ampere Altraに搭載された80の計算コアはcoherent meshで接続されており、DDR4メモリと計算コアとの距離は均等である(完全に均等ではないにせよ、大きく異なる構成ではない)。キャッシュメモリについては各コアに1MBのL2キャッシュと、ソケット全体で32MBの共有キャッシュを備えている。

A64FXの構成概要を図2に示す。A64FXに搭載された

48 コアは 12 コア=1CMG(Core Memory Group) ごとに分かれており、各 CMG に対して HBM が接続されている。そのため CMG に直結したメモリへのアクセスと異なる CMG に接続されたメモリへのアクセスには大きな性能差がある。プログラムの最適化においてはプロセスやスレッドの配置などに注意が必要であり、利用者の感覚は 4 ソケット CPU を利用する際の感覚に近い。キャッシュメモリについては CMG ごとに 8 MB の L2 キャッシュを備えており、CMG を跨いだ共有キャッシュは搭載していない。

EPYC(Rome[6], Milan[7]) の構成概要を図 3 と図 4 に示す。EPCY は内部のコアが幾つかのブロック (Core Complex (CCX), Core Complex Die (CCD)) に分かれているため計算コア同士の接続が均等ではなく、Ampere Altra よりも A64FX の構成に近いと言える。Rome と Milan の大きな違いは、図中に示したとおり、L3 キャッシュの共有の仕方にある。Rome は 4 コア単位で L3 キャッシュを共有しているのに対して、Milan は 8 コア単位で L3 キャッシュを共有している。L2 キャッシュについては Rome も Milan も同様にコアあたり 512KB 備えている。DDR4 メモリの接続については 2 つの CCD ごとに 2 チャンネルのメモリと接続されるため、A64FX 同様に 4 ソケット CPU を利用するような使い方が適している。

Cascade Lake[8] の構成概要を図 5 に示す。Cascade Lake はコア数により構成が若干異なるが、18 コアを超えるモデルは本図の構成が基本となっている。Cascade Lake の内部のコアはメッシュ接続 (図中の赤線) されており、各計算コアとメモリとの距離はおおよそ均等である。そのため、A64FX よりも Ampere Altra に近い構成であると言える。キャッシュについては、各コアに 1MB の L2 キャッシュと、コア数 \times 1.375MB の共有 L3 キャッシュを搭載している。

今回利用した Ampere Altra の動作周波数は 3.00 GHz であり、利用状況に応じて周波数を引き上げる機能 (いわゆるターボブースト機能) は有していない。この動作周波数は、2.00 GHz または 2.20 GHz のモデルが存在する A64FX と比べると非常に高速である。EPYC や Xeon の動作周波数は様々なバリエーションがあるが、今回挙げている (性能評価に用いている) CPU と比べると、通常の動作周波数で比べると Ampere Altra が最も動作周波数が高いが、ターボブースト時の動作周波数は EPYC や Xeon の方が高い。一般的に理論演算性能は通常の動作周波数で比較するため Ampere Altra の理論演算性能は Cascade Lake と EPYC の間に位置しているが、シングルコアしか動作していないなどターボブースト機能が有効になる状況での性能を比較した場合は Ampere Altra の性能が相対的により低くなる点に注意が必要である。

CPU に接続されたメインメモリについては、Ampere Altra は PC サーバで一般的な DDR4 を採用している。1

ソケットあたりのチャンネル数は最大 8 であり、DDR4-3200 メモリに対応することから、1 ソケットあたりの理論メモリ転送速度は 204.8 GB/s である。他の CPU と比較すると、EPYC とは同じ構成と性能、Cascade Lake は同様に DDR4 メモリであるが動作周波数が低くチャンネル数が少ないため性能が劣る。A64FX は高速な HBM2 を採用しているため 1,024 GB/s という高い理論性能を有する一方、メモリ容量は 32 GB に制限されている。高速で大容量で安価なメモリが使えることが望ましいが、これらを両立させることは難しいのが現状である。

Ampere Altra は 1 サーバあたり 1 ソケットまたは 2 ソケットを搭載した状態で利用することが想定されており、本稿では 2 ソケットを搭載したサーバを用いている。ただし性能評価に十分な時間が取れなかったため、次章のベンチマークの多くは 1CPU 同士の比較となっている。

Ampere Altra の命令セットアーキテクチャは Arm v8.2+ である。Arm v8.2-A + SVE を採用している A64FX とは、ベースのアーキテクチャは同じ Arm v8 であるが、富士通社が独自に開発し A64FX に搭載した HPC 向けの機能等は含まれていない。対応する SIMD 命令長についても、A64FX では最大 512bit に拡張されているのに対して、Ampere Altra は 128bit である。

以上のように、Ampere Altra は他の現世代の CPU とはバランスや構成が異なる点が多い。ただし、計算コア同士の接続方法やメモリとの接続形態は今回挙げた中ではシンプルな構成であり、特に CPU 内部のコアが幾つかのブロックに分かれていることを考慮する必要がある A64FX や EPYC と比べると、利用する際に考慮すべき事項は少なく、容易に利用できる CPU であると考えられる。

3. 性能評価

本章では各 CPU でプログラムを実行した際の性能について報告する。対象プログラムとしては、HPC 分野でよく利用されるベンチマークプログラムを中心に以下を用いる。

- STREAM ベンチマーク
- DGEMM 性能と HPL ベンチマーク
- SpMV 性能と HPCG ベンチマーク
- GKV ベンチマーク
- OpenFOAM (オープン CAE 学会チャンネル流ベンチマーク)

性能比較に用いた計算機環境は表 1 に準じており、以下では各計算機環境をそれぞれ Ampere Altra、Rome、Milan、A64FX、CLX と呼ぶ。

3.1 STREAM ベンチマーク

STREAM ベンチマーク [9] はメモリ転送性能の測定に広く用いられているベンチマークである。STREAM ベンチマークには Copy, Scale, Add, Triad の 4 つの評価指標

があるが、ここでは Triad の値を示す。Triad は、スカラ倍した配列を他の配列に加算し、その結果をさらに他の配列に代入するという内容 (図 6) である。

```
#pragma omp parallel for
for (j=0; j<STREAM_ARRAY_SIZE; j++)
    a[j] = b[j]+scalar*c[j];
```

図 6 STREAM Triad のソースコード

使用したコンパイラと主なオプションは以下の通りである。

Ampere Altra gcc 9.3.1, -O3 -fopenmp

-march=armv8.2-a

Rome gcc 10.2.1, -O3 -fopenmp -march-znver2

Milan gcc 10.2.1, -O3 -fopenmp -march-znver2

A64FX frtpx 4.5.0, -Kfast,openmp,zfill

CLX icc 19.1.3.304, -O3 -axCORE-AVX512

-qopenmp

Ampere Altra のベンチマーク結果を図 7 に示す。計算コアに対するスレッドの割り当て方は環境変数 OMP_PROC_BIND に TRUE か SPREAD を設定することで制御した。compact 40M は compact なアフィニティ設定 (片方の CPU ソケットの全コアにスレッドを割り当ててからもう一方の CPU ソケットにもスレッドを割り当てる、OMP_PROC_BIND=TRUE) で配列サイズが 40,000,000、spread 40M は spread なアフィニティ設定 (全 CPU ソケットに均等にスレッドを割り当てる、OMP_PROC_BIND=SPREAD) で配列サイズが 40,000,000 という実行条件を意味している。さらに numactl -l を指定することで CPU コアに近いメモリを利用している。STREAM ベンチマークは配列サイズが大きすぎる場合や小さすぎる場合に異常な性能値が出るため、問題サイズをいくつか変えて測定した中で妥当な性能となったものを選んだ。Ampere Altra の結果は、compact 40M の場合、利用するコア数を 1 コアから順に増やしていくと性能が向上し、およそ 25 コアから 80 コア利用した際に約 180 GB/s で一旦頭打ちになっている。この範囲での最大性能は 38 コア使用時の 180 GB/s であった。その後、80 コアを超えると 2 ソケットを利用したメモリアクセスとなるため性能が向上し、156 コア使用時に最大 336 GB/s を達成している。これは compact AFFINITY の挙動としては妥当であり、理論メモリ転送性能に対して 1 ソケットで 87.8%、2 ソケットで 82.2% という性能比についても良好な性能であると言える。一方で spread 40M の場合、少ないスレッド数でも両ソケットに接続されたメモリを使えるため高い性能が得られ、51 コア使用時に最大 350 GB/s を達成している。しかし、より多くの計算コアを用いた場合には性能値が上下し、100 コアから 140 コア程度の範囲では 300 GB/s に届かないという結果となった。多くのコアを用いた際に性能が安定しないという結果は特別珍しい

傾向ではないが、一般のプログラムを実行する際には注意する必要がある傾向であると言える。

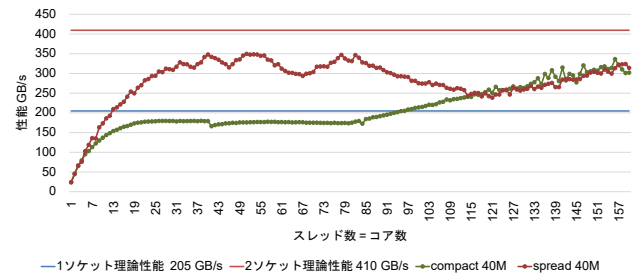


図 7 STREAM Triad の性能 : Ampere Altra

EPYC (Rome, Milan) のベンチマーク結果を図 8 に示す。Ampere Altra と同様にコアの割り当て方を変えて性能を測定した。これらの CPU は 1 コアあたり 2 つのスレッドを同時に実行可能であるため、利用スレッド数を増やすと各コアに 2 つめのスレッドが割り当たる。1 つのコアに複数のスレッドが割り当たるような状況では STREAM の性能は低下する傾向にあることが知られている。Ampere Altra 同様に OMP_PROC_BIND を設定してプログラムを実行した結果、全体的には Rome と Milan の傾向は類似しており、compact AFFINITY ではスレッド数を増やすに従って徐々に性能が向上、spread AFFINITY ではより少ないスレッド数で高い性能を得られている。ただし、Rome の性能が非常に低く、2 ソケット利用しているにも関わらず 1 ソケット相当の性能しか得られていない。この原因については、Rome と Milan は 1 ソケットが複数の CCD から構成されておりメモリも CCD ごとに接続されているため、それにあわせて 1 CPU ソケットあたり複数の NUMA ノードという構成となっているのが望ましいのに対し、今回利用した環境では 1 CPU ソケットあたり 1 NUMA ノードという構成になっており、これが原因で最適なメモリアクセスが行えなかったためであると予想している。コンパイラに原因がある可能性も考慮し Intel コンパイラでも性能を確認したが、やや振る舞いは異なるものの、やはり十分な性能が得られなかったため、NUMA 設定の影響があると考えられる。NUMA 構成を適切に変更した際の性能については改めて調査予定である。Milan については、コアあたり 1 スレッドが使われているスレッド数 128 まではほぼ理想的な性能の振る舞いとなっている。一方コアあたり 2 スレッド時の性能は低めであり、メモリバンド幅の観点からはコアあたり 1 スレッドで利用するべきである。今回測定した範囲での最大性能は、Rome では spread AFFINITY 253 スレッドの 172 GB/s(42%)、Milan では spread AFFINITY 32 コアの 357 GB/s(87%) であった。

A64FX と CLX のベンチマーク結果を図 9 に示す。

A64FX については、常に環境変数 XOS_MMM_L_PAGING_POLICY=demand:demand:demand

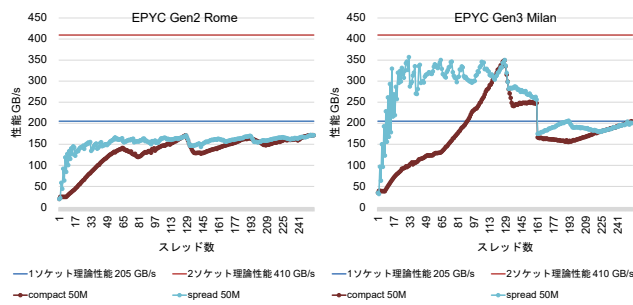


図 8 STREAM Triad の性能 : EPYC (Rome, Milan)

を設定、さらに `numactl -C 12-59 -c 4-7` で利用するスレッドとメモリを指定して実行した。OMP_PROC_BIND 環境変数を指定しない状態では compact AFFINITY 相当となり、12 スレッドまでは 1 つの CMG 内で実行、それ以上のスレッド数では 12 スレッド増加するたびに利用する CMG が増えるというスレッド割り当てとなる。OMP_PROC_BIND=SPREAD を指定すると spread AFFINITY 相当となり、他の CPU と同様に全 CMG にまたがって分散配置となる。compact AFFINITY の場合、6 スレッドで約 200GB/s に達すると一旦性能が頭打ちとなり、その後さらにスレッド数を増やすに従って徐々に性能が向上する。48 コア全て利用した際に最大 824 GB/s を達成、このときの理論メモリ転送性能に対する性能比は 81% である。spread AFFINITY の場合は全 CMG に 6 スレッド割り当たる 24 スレッドの時点で約 800 GB/s に到達し、その後はやや性能が上下するが 700 GB/s 以上を保ち、44 コア全てにスレッドが割り当たった時点で最大性能 828 GB/s(81%) を達成した。Ampere Altra や Milan と比べると最大の性能比がやや低いものの、そもそも理論メモリ性能が非常に高いため、非常に高い性能が得られている。

CLX については、Intel コンパイラを用いたため、環境変数 `KMP_KMP_AFFINITY` を設定することでスレッド割り当て方法を変更して性能を評価した。compact AFFINITY では、利用するスレッド数が増えることで性能が向上するが 10 スレッド程度で一旦性能が頭打ちになり、20 スレッドを超えると 2 ソケット目も利用することで性能が向上するという妥当な傾向となっている。最大性能は、compact AFFINITY 1 ソケットでは 16 スレッド実行時に 102 GB/s(72%)、compact AFFINITY 2 ソケットでは 40 スレッド実行時に 191 GB/s(68%) であった。2 ソケットを均等に利用する balanced AFFINITY では、スレッド数が 8 以下の場合には compact AFFINITY とあまり変わらない性能であるが、20 スレッド程度まで性能が向上し、それ以上は性能が横ばいとなっている。最大性能は 25 スレッド実行時に 193 GB/s(68%) であり、やや低めの理論性能比ではあるが、妥当な傾向であると言える。

以上のように、Ampere Altra の STREAM ベンチマーク性能を他の CPU と比較してみると、CLX と近い傾向

が確認できた。Ampere Altra と CLX は、コア構成がメッシュ接続で DDR4 メモリ、さらに 1 コアあたり 1 スレッドを割り当てて利用しているという類似性を持つため、妥当な結果であると考えられる。その中でも Ampere Altra の理論性能に対する実測値の割合は高く、優れたメモリ性能が発揮できる CPU であると判断できる。ただし、spread AFFINITY で多コア実行した際の性能がやや低いため、この構成でのプログラム実行には注意するべきである。

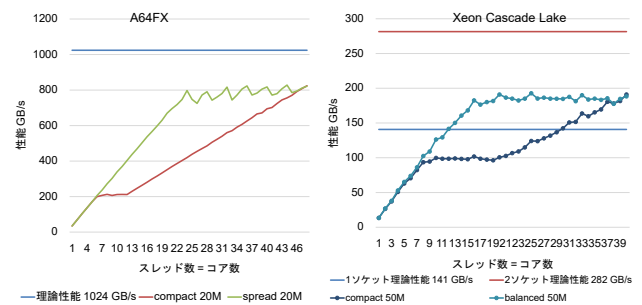


図 9 STREAM Triad の性能 : 「不老」 Type I (A64FX), II (Cascade Lake)

3.2 DGEMM 性能と HPL ベンチマーク

密行列演算性能の評価としては、TOP500 ランキング [10] でも利用されている High Performance Linpack(HPL)[11] ベンチマークが多く用いられている。HPL は LU 分解により連立一次方程式の求解を行う性能を求めるベンチマークであり、精度浮動小数点データに対する行列積和演算 (Level-3 BLAS DGEMM) の性能に大きく影響を受けることが知られている。しかし HPL はプログラムの書き換えや実行時パラメタの選択により大きく性能が変わるため、十分に最適化された環境とそうでない環境のスコアを並べるとミスリードとなる可能性がある。今回は各環境に向けたコードの最適化が行えていないため、HPL とあわせて BLAS ライブラリを用いた単純な DGEMM の性能も確認し、密行列演算性能の評価とする。

各計算機環境で利用したコンパイラと BLAS ライブラリの組み合わせ (コンパイル時オプションとリンクオプション) は以下の通りである。実行スレッド数はいずれも CPU1 ソケットに搭載された物理コア数と同数である。

```
Ampere Altra gcc 9.3.1 + Free Arm Performance
Libraries (ARMPL) V20.3, -O3 -fopenmp
-march=armv8.2-a -larmplmp
Rome gcc 10.2.1 + AMD Optimizaing CPU
Libraries 3.0.6, -O3 -fopenmp -march=znver2
-lblis-mt
Milan gcc 10.2.1 + AMD Optimizaing CPU
Libraries 3.0.6, -O3 -fopenmp -march=znver2
-lblis-mt
```

A64FX fccpx 4.5.0, -Kfast,openmp -SSL2BLAMP
CLX icc 19.1.3.304 + 付 属 MKL, -O3 -qopenmp
-axCORE-AVX512 -mkl

各計算機環境における DGEMM の性能を図 10 に示す。実行時間測定対象は `cblas_dgemm` 関数による計算時間である。対象行列としては一辺が 2048 の正方行列を用いた。実験の結果、Ampere Altra、Rome、Milan、CLX についてはいずれも 1000 GFLOPS 前後の近い性能値が得られた。特に Ampere Altra、Rome、Milan の性能が $AmpereAltra < Rome < Milan$ の順で僅差となっている点については、理論演算性能の関係を考慮すると妥当であると考えられる。一方で理論演算性能が低い CLX が理論演算性能に対して約 75%と高い性能を達成し Ampere Altra や Rome よりも高い性能を得ている点については、MKL による最適化が CPU の性能をより良く引き出せているためであると考えられる。A64FX は理論演算性能に遠く高い性能を達成し、他の CPU の追従を許していない。ただし理論演算性能に対する割合は約 56%であり、CLX の割合の高さが際立っている。

つづいて HPL の性能を図 11 に示す。HPL の性能は、Ampere Altra、Rome、Milan、CLX については HPL-2.3 をそのままビルドして測定した値、A64FX については HPL-2.3 をそのままビルドして測定したものと富士通社により最適化されたもの (TOP500 向けに最適化されたものを 1 ノードで実行したもの、図 11 では最適化版と記す) の値を示している。HPL の性能の大小はおおよそ DGEMM 性能の大小を反映しており、 $AmpereAltra < Rome \approx CLX < Milan < A64FX$ の関係が確認できる。これは妥当な傾向であると考えられるが、HPL には問題サイズやブロックサイズなど多数の設定パラメタがあり、プロセス数とスレッド数も性能に影響する。さらに各パラメタの最適値は実行環境によっても異なる。

今回はある程度のパラメタ調査を行ったなかで得られた最も良い性能を示しているが、いずれの環境についてもより良いパラメタがある可能性がある点には注意されたい。また、A64FX の最適化版と非最適化版を比べると性能が 30%強向上している。A64FX 以外についてもコード修正により同様に性能が向上する可能性があることも付記しておく。

3.3 SpMV 性能と HPCG ベンチマーク

HPL ベンチマークは計算機の性能を測定する指標として長く用いられてきたが、実問題の性能を反映できていない、十分な性能値を得ようとすると測定に時間がかかる、といった問題点が指摘されている。そこで、より実問題に近く短時間でも実行できるベンチマークとして、有限要素法から得られる疎行列を対象として共役勾配法 (Conjugate Gradient method, CG 法) を用いて連立一次方程式を解く

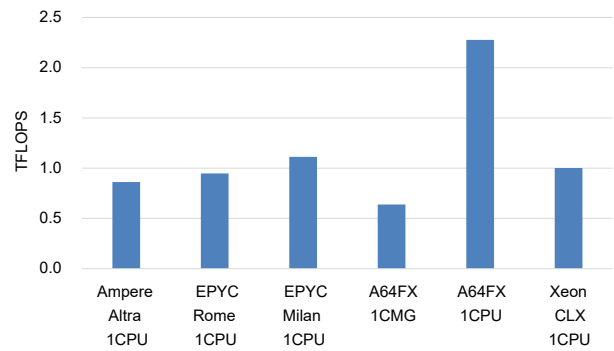


図 10 行列積和演算 (BLAS DGEMM) の性能 (一辺の長さが 2048 の正方行列)

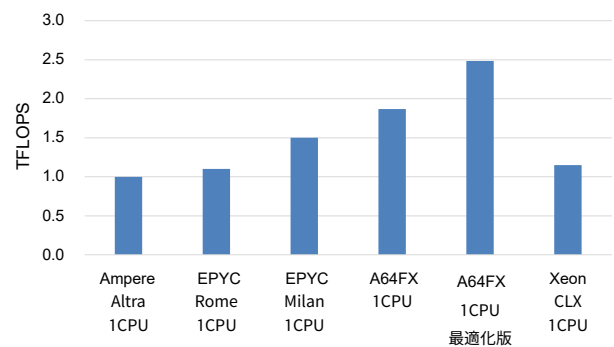


図 11 HPL の性能

性能を求めるベンチマークである HPCG ベンチマーク [12] が提案され、活用が進んでいる。

そこで、各計算機環境において HPCG ベンチマークの性能を測定し比較した。ただし HPCG も HPL 同様にプログラムの最適化が性能に大きく影響することから、Compress Row Storage (CRS) 形式で格納された簡単な疎行列とベクトルの積 (SpMV) の性能も測定して比較した。

SpMV の実装は、図 13 に示すような単純なプログラムで行った。対象とする行列は 3 次元の 7 点ステンシル計算で生じるような単純な構造の疎行列であり、空間領域は 300x300x300、約 1.4GB の容量である。コンパイラや最適化オプションは STREAM と同一のものを利用した。測定結果を図 13 に示す。疎行列ベクトル積の性能はメモリ転送性能に大きく影響を受けることが知られており、STREAM ベンチマークの結果を考慮すると妥当な結果が得られていると言える。すなわち、A64FX が他より高い性能を発揮し、Ampere Altra、Rome、Milan が近い性能、CLX がやや低い性能である。

```
#pragma omp parallel for private(j)
for(i=0; i<n; i++){
    double tmp = (double)0.0;
    for(j=irp[i]; j<irp[i+1]; j++){
        tmp += val[j] * vec[col[j]];
    }
    ans[i] = alpha * tmp;
}
```

図 12 SpMV プログラム

Rome と Milan では Intel コンパイラでも測定したが、GCC と比べて有意な性能差はなかった。また MKL や ARMLPL には今回のような SpMV 計算を行う計算カーネルが含まれているが、それらを用いても単純な実装に対して有意な性能向上は得られなかった。

つづいて HPCG の性能を図 14 に示す。HPCG の性能をベンチマークランキングに掲載するには 30 分以上の実行が必要であるが、ここでは参考値として 60 秒程度の短時間実行で得られた値を掲載している。ただし A64FX については富士通社により最適化されたもの (ベンチマークランキング登録用に最適化されたものの 1 ノード版) の値を示している。評価結果としては、Ampere Altra と Rome および Milan については僅差であるが、STREAM や SpMV で得られたメモリ性能の差に反して $AmpereAltra < Rome < Milan$ の関係となった。HPCG は疎行列演算性能を測定するものとはいえ疎行列計算カーネルのみの時間を測定するものではないため、演算性能の大小の方が強めに反映されたと考えるのが妥当であろう。

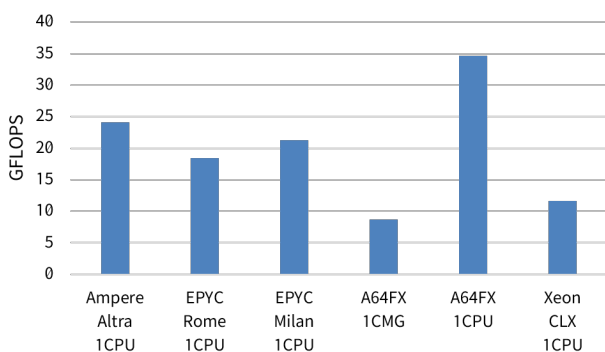


図 13 疎行列ベクトル積 (SpMV) の性能

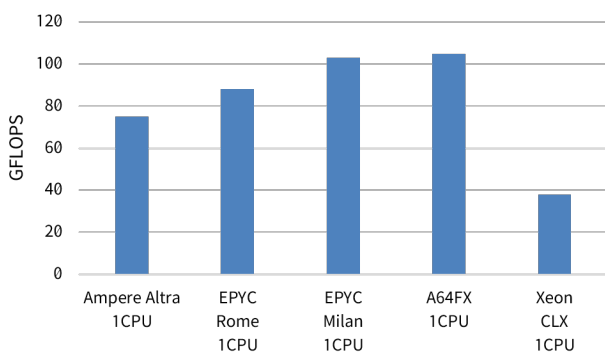


図 14 HPCG の性能

3.4 GKV ベンチマーク

GKV ベンチマークは、磁場閉じ込め核融合に向けたプラズマ乱流現象の解析のために核融合科学研究所にて開発されたプラズマ乱流解析コード GKV (GyroKinetic Vlasov

code) を元に、名古屋大学の片桐・渡邊らがそのカーネル部分を抜き出して作成したベンチマークである。本ベンチマークは以下の 4 つのプログラムにより構成される。

kernel1_fft FFT と MPI_Alltoall に関するベンチマーク。

MPI と OpenMP によるハイブリッド並列化コード。

kernel2_intgrl 配列のリダクションに関するベンチマーク。OpenMP によるスレッド並列化コード。

kernel3_diff45 4 次元および 5 次元の有限差分法の演算に関するベンチマーク。OpenMP によるスレッド並列化コード。

kernel4_diff123 1 次元, 2 次元および 3 次元の有限差分法の演算に関するベンチマーク。OpenMP によるスレッド並列化コード。

この GKV ベンチマークのうち 1 ノードでの実行に適している kernel2 から kernel4 について、各計算機環境で性能を測定した。なお、A64FX と CLX における性能は文献 [5] でも報告しているが、コンパイラのバージョンなどが一致していない点に注意されたい。

プログラムのコンパイルに用いたコンパイラと主なコンパイルオプションは以下の通りである。EPYC(Rome,Milan) では gcc よりも icc の方が有意に高い性能が得られたため、本章では icc による結果を採用している。また kernel2 についてはいずれも文献 [5] でも用いた最適化版コード (OpenMP により並列リダクションを行っている部分の指示文の使い方を変更したもの) を採用している。実行スレッド数はいずれも CPU1 ソケットに搭載された物理コア数と同数である。

Ampere Altra gcc 9.3.1, -O3 -fopenmp

-march=armv8.2-a

Rome icc 20211 Beta, -O3 -qopenmp -xHOST

Milan icc 20211 Beta, -O3 -qopenmp -xHOST

A64FX frptx (FRT) 4.5.0, -Kfast,parallel

-Kopenmp,largepage,loop_nofission

-Ksimd_nouse_multiple_structures,mfunc=2

CLX ifort (IFORT) 19.1.3.304, -O3 -qopenmp

-axCORE-AVX512

測定結果を図 15 から図 17 に示す。

全体的な傾向として、A64FX の実行時間が他と比べて特に短く、Ampere Altra と EPYC は $AmpereAltra < Milan < Rome$ の順になっている。CLX は、kernel2 と kernel4 では最も遅いが、kernel3 では Rome と Milan の間となっている。HPL や HPCG では Rome や Milan に劣っていた Ampere Altra だが、GKV ベンチマークでは逆転する結果となった。また、Ampere Altra の特徴として、有限差分法カーネルである kernel3 と kernel4 では EPYC に対して大きな差を付けているのに対して配列のリダクションがメインである kernel2 ではその差が小さくなっている。この原因としては、Ampere Altra はコア数が多いため、

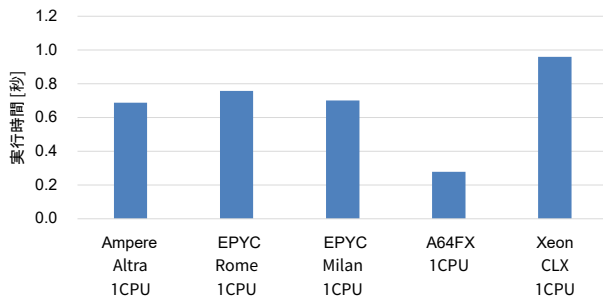


図 15 GKV ベンチマーク kernel2 の実行時間

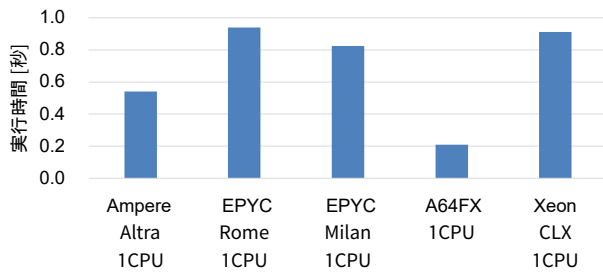


図 16 GKV ベンチマーク kernel3 の実行時間

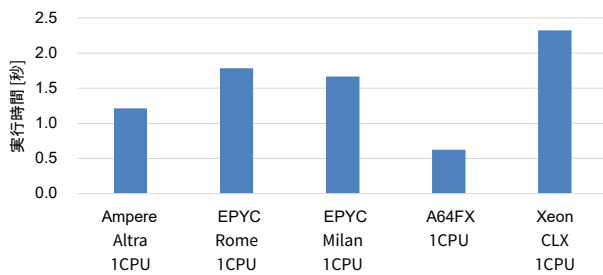


図 17 GKV ベンチマーク kernel4 の実行時間

kernel2 で重要な多数のスレッドによるリダクションの負荷が大きい可能性がある。これはコアの多い CPU では課題となる点の 1 つであり、さらなる性能解析や性能改善手法の検討の余地がある。

3.5 OpenFOAM

OpenFOAM[13] は流体計算向けのオープンソースソフトウェアであり、MPI 並列実行に対応していることから様々な並列計算環境にて利用されている。本章では多くのスパコンやクラウドにて測定した実績のあるオープン CAE 学会チャンネル流ベンチマーク [14] を実行し、その性能を比較する。本ベンチマークの解析条件を表 2 に示す。

Ampere Altra と EPYC については、OpenFOAM-v2006 を gcc 9.3.1/10.2.1 でビルドして測定した。A64FX と CLX についてはシステムにインストール済み (modulefile 提供済) の OpenFOAM-v2006 を用いた。問題サイズ (格子数) 0.37M, 3M, 24M それぞれの各 CPU による実行時間を図 18 に示す。

測定の結果、Ampere Altra の実行時間は、全体的に

表 2 解析条件

レイノルズ数 Re_τ	110
主流方向	一定の圧力勾配
主流・スパン方向	周期境界
ソルバ	pimpleFoam
乱流モデル	無し (laminar)
領域分割手法	scotch (周期境界面は同領域)
速度線形ソルバ	BiCG (前処理 DILU)
圧力線形ソルバ	PCG (前処理 DIC)
格子数	約 0.37M, 約 3M, 約 24M
MPI プロセス数/ノード	物理コア数と一致

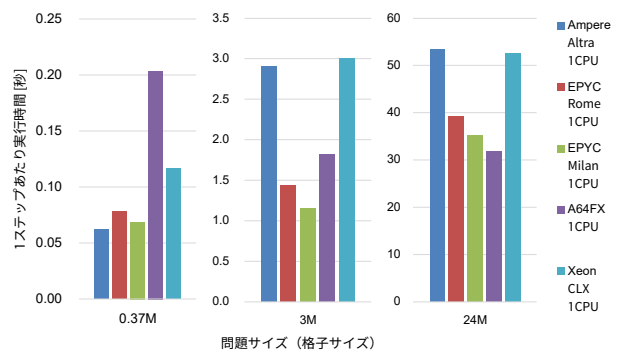


図 18 OpenFOAM ベンチマークの実行時間

り良好ではなく、理論性能に近い EPYC と比べて遅いという傾向が得られた。A64FX の性能も理論性能の高さに比べてあまり良い性能が得られていない (EPYC と比べると 3M では逆転、24M では勝っているが差が小さい、0.37M は問題サイズの小ささなど別の問題と思われる) ことも考慮すると、現在の OpenFOAM の実装またはコンパイラの最適化が ARM 系 CPU に適していない (十分ではない) 可能性があると考えられる。コンパイラや最適化オプションの精査、コードの修正により性能向上するかについては今後の課題とする。

4. おわりに

本稿では Ampere Altra を中心にいくつかの CPU の性能をベンチマークプログラムにて比較し、その結果について考察した。全体的な性能の傾向としては、今回比較対象とした EPYC や Xeon といった x86 系の CPU と比べるとおよそ理論性能 FLOPS および GB/s の大小にならった性能が得られた。一方、同じ ARM 系の CPU である Ampere Altra と A64FX を比較した場合は、高性能計算向けに最適化された A64FX の性能の高さが目立つ結果となった。

Ampere Altra の傾向として、性能測定範囲が特定カーネルに限られる DGEMM, SpMV, GKV ベンチマーク (kernel3, kernel4) では良好な性能が得られたのに対して、より広い範囲で処理が複雑になる HPL, HPCG, OpenFOAM ではやや劣った性能となった。具体的にどのような部分で

性能差が生じるのかの解析ができていないものの、Ampere Altra のコア数が多いことを考慮すると、十分な並列度のある計算カーネルでは良好な性能が得られる一方、それ以外の並列度の低い部分の割合が増えると性能を発揮できないというコアの多い CPU の弱点が露見している可能性がある。

今回は Ampere Altra 向けにプログラムの最適化は行っていない。どのような最適化手法が有効なのか、他の CPU と異なる点があるのかについては今後十分な調査が必要である。Ampere Altra は NUMA 最適化について考える必要がないという点では最適化が容易であることが期待できるが、リダクション処理が多い GKV ベンチマーク kernel2 でやや性能が低かったこともあり、コアの多い CPU ならではの工夫も必要となることがあるだろう。また、今回は STREAM を除いて 1 ソケット利用時の性能しか評価できていない。1 ノード単位の性能比較についても今後行う予定である。

謝辞 評価環境として Oracle Cloud Infrastructure を提供していただいた Oracle Cooperation に感謝します。

参考文献

- [1] スーパーコンピュータ「富岳」について — 理化学研究所 計算科学研究センター (R-CCS) <https://www.r-ccs.riken.jp/jp/fugaku> (accessed 2021-04-13)
- [2] スーパーコンピュータシステム — 名古屋大学 情報連携推進本部 <https://icts.nagoya-u.ac.jp/ja/sc/> (accessed 2021-04-13)
- [3] FUJITSU Processor A64FX : 富士通 <https://www.fujitsu.com/jp/products/computing/servers/supercomputer/a64fx/> (accessed 2021-04-13)
- [4] Ampere®Altra®Processor - Ampere Computing <https://amperecomputing.com/altra/> (accessed 2021-04-13)
- [5] 大島聡史、永井亨、片桐孝洋、“スーパーコンピュータ「不老」の性能評価”、情報処理学会研究報告 (SIGHPC-175)、pp.1-10、2020.
- [6] 2nd Gen AMD EPYC™Processors — EPYC™7002 Series — AMD <https://www.amd.com/ja/processors/epyc-7002-series> (accessed 2021-04-13)
- [7] AMD EPYC™7003 Series Processors — AMD <https://www.amd.com/ja/processors/epyc-7003-series> (accessed 2021-04-13)
- [8] Cascade Lake : 概要 <https://www.intel.co.jp/content/www/jp/ja/design/products-and-solutions/processors-and-chipsets/cascade-lake/2nd-gen-intel-xeon-scalable-processors.html> (accessed 2021-04-13)
- [9] MEMORY BANDWIDTH: STREAM BENCHMARK PERFORMANCE RESULTS <https://www.cs.virginia.edu/stream/> (accessed 2021-04-13)
- [10] Home — TOP500 Supercomputer Sites <https://www.top500.org/> (accessed 2021-04-13)
- [11] HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers <https://www.netlib.org/benchmark/hpl/> (accessed 2020-04-13)
- [12] <https://www.hpcg-benchmark.org> <https://www.hpcg-benchmark.org/> (accessed 2020-04-13)
- [13] OpenFOAM — Free CFD Software — The OpenFOAM Foundation <https://openfoam.org/> (accessed 2021-04-13)
- [14] The Open CAE Society of Japan / OpenFOAM-BenchmarkTest <https://gitlab.com/OpenCAE/OpenFOAM-BenchmarkTest> (accessed 2021-04-13)