

可変ベクトル長バイナリを用いた Arm SVE プロセッサ A64FX の評価

住元 真司^{1,a)} 稲田 由江¹ 三輪 英樹¹ 三吉 郁夫¹

概要：本稿では、Arm SVE プロセッサである A64FX 上で可変ベクトル長バイナリを用いた著名 HPC ベンチマークプログラムの評価について報告する。A64FX は Arm SVE アーキテクチャに対応しているため実行時に SIMD ベクトル長を変更できる上、独自にメモリバンド幅も変更可能である。これらの機能を用いて著名ベンチマークの実行特性を評価する。

1. はじめに

我々は、Arm の命令セットを採用した HPC 向け A64FX プロセッサを開発している。元々 Arm プロセッサは組み込みシステム向けに発展してきた経緯もあり、HPC 向けに必要な単一命令で複数データセットの演算を行う SIMD 命令は備えていなかった。このため、富士通は Arm 社に SIMD 拡張を提案し、リードパートナーとして共同開発を進めた。この結果開発されたのが Armv8-A Scalable Vector Extension (SVE) 拡張仕様 [1], [2], [3] である。Armv8-A SVE は Arm の ISA アーキテクチャとして標準仕様化されているため、Arm 社の開発ライセンスがあれば誰でも Armv8-A SVE 仕様のプロセッサを開発することが可能である。この Armv8-A SVE 仕様で開発されたのが A64FX プロセッサである。

Armv8-A SVE 仕様の SVE で規定される SIMD 演算仕様の特徴的なものとして、将来のベクタ長拡張に備えて、インストラクションセットではベクタ長を規定していない点がある。更に仕様で規定されている 128 ビットから 2048 ビットまでのベクタ長のうち、実際に提供するベクタ長以下の 128 ビットの倍数のベクタ長でも実行できることがハードウェア仕様として定義されている点がある。つまり、256 ビット以上のベクタ長をもつ SVE 仕様のプロセッサであれば一つのバイナリで 128 ビットと 256 ビットと複数のベクタ長で実行できる。

さて、A64FX がターゲットとする HPC 向けのアプリケーションにおいては、主に 2 つの特性がある。計算イン

テンシブとメモリバンド幅インテンシブである。しかし、実際のシステムにおいては、この 2 つの特性はシステムで提供される演算能力とメモリバンド幅の相対差に依存するためアプリケーションの性能を引き出すためには、定量的に把握する必要がある。しかし、定量的に把握するためには、専用のツールやアプリケーションの修正しての性能評価が必要になるため、実行するのは容易ではない。

アプリケーションの 2 つの特性を調べる手法として、ハードウェアで意図的に演算能力やメモリバンド幅を制限できれば、2 つの特性に準ずる性能差が観測できるはずである。

今回、Armv8-A SVE 仕様による演算能力の制御機能と A64FX が持つメモリバンド幅を制御する機能を用いて、各種基本ベンチマークを用いた性能評価を行ったので、その評価結果について報告する。

2. Armv8-A アーキテクチャと SVE 拡張仕様

Armv8-A アーキテクチャは、2011 年に策定された Arm のプロセッサアーキテクチャであり、Armv7 までは 32 ビットであったのが、新たに 64 ビット用の Aarch64 と既存の 32 ビット用の Aarch32 の 2 つの命令セットを持つ。Aarch64 では 31 個の汎用レジスタとプログラムカウンタ、スタックポインタを持っている。また、128 ビット SIMD である NEON とベクトル処理の VFP (Vector Floating-point Architecture) も搭載している。

SVE 拡張仕様は 2016 年に発表された Armv8-A 向けの SIMD 拡張仕様である。それまで Arm プロセッサは HPC 向けに必要な単一命令で複数データセットの演算を行う SIMD 命令は備えていなかった。このため、富士通は Arm 社に SIMD 拡張を提案し、リードパートナーとして共同開発を進めた。この結果開発されたのが Armv8-A

¹ 富士通株式会社
4-1-1, Kamikodanaka Nakahara-ku, Kawasaki 211-8588, Japan

^{a)} sumimoto.shinji@jp.fujitsu.com

Scalable Vector Extension (SVE) 拡張仕様 [1], [2], [3] である。Armv8-A+SVE は Arm の ISA アーキテクチャとして標準仕様化されているため、Arm 社の開発ライセンスがあれば誰でも Armv8-A+SVE 仕様のプロセッサを開発することが可能である。

SVE 拡張 (Scalable Vector Extension) 仕様は Armv8-A アーキテクチャに対する拡張仕様として定義されている。以下に、Armv8-A+SVE 仕様の概要を述べる。

- SVE 拡張仕様を満す実行バイナリではベクタ長は規定されず、実行時に規定されるので、ベクタ長が異なる Arm+SVE 仕様のプロセッサ間で同一の実行バイナリを実行することが可能である。(図 1 参照)
- ベクタレジスタ 32 個、プレディケートレジスタ 16 個、例外制御レジスタから構成される。
- ベクタ長が変更されてもレジスタ数は変わらず、それぞれのレジスタ長が変化する。
- 演算だけでなく、Scatter-Gather といったメモリアクセス系の命令セットも提供されている。
- SVE 拡張の詳細は文献 [2] を参照されたい。



図 1 SVE

以上のように、Armv8-A+SVE 仕様を用いることにより、一つの実行バイナリがベクタ長の異なるプロセッサにおいてもそれぞれのプロセッサのベクタ長で実行できるようになる。(図 1)

3. A64FX の概要

A64FX は富士通が開発した世界で最初の Arm V8+SVE に準拠したマイクロプロセッサである。A64FX の構成を図 2 に示す。

A64FX は Arm V8 仕様に適合し実行環境も SBSA と SBBR 仕様に準拠しているためオペレーティングシステムを含むバイナリ互換を有している。以下に A64FX の主な概要について述べる。

- 12 コアの計算用コアと 1 つのアシスタントコアに 1 つの HBM2 メモリから構成される CPU Memory Group(CMG) が 4 つと PCIe Unit, Tofu-D Unit から構成される。
- 計算用コアとアシスタントコアは Arm V8+SVE に準

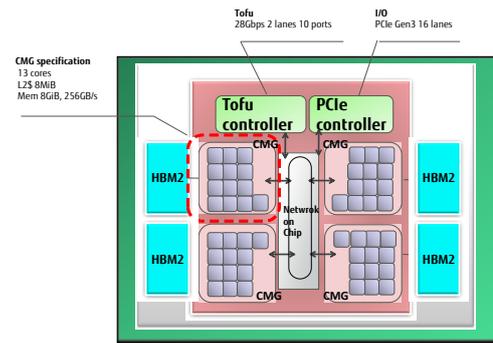


図 2 A64FX の構成

拠したプロセッサコアである。Arm V8 仕様は 32bit ならびに 64bit 仕様があるが、A64FX は 64bit 仕様のみを実装している。SVE 仕様におけるベクタ長は 512bit 長計算を提供する。

- HBM2 メモリは 32GB で 1 つあたり最大 256GB/s、A64FX 全体で 1TB/s のメモリバンド幅を提供する。
- PCIe Unit は PCI express Gen.3 16 レーンを提供する。
- Tofu-D Unit は Tofu 仕様のインタコネクタであり、1 リンク性能 6.8GB/s を 10 リンク提供するほか、ネットワークインターフェイスを従来の 4 基から 6 基に拡張している。また、バリアリソースの増強、キャッシュインジェクションによる低遅延通信を実現する。
- Arm ベースのサーバ仕様である Server Base System Architecture (SBSA) のハードウェア仕様に準拠し、ファームウェアレベルで Server Base Boot Requirements (SBBR) 仕様を満たすことにより、バイナリレベルでの OS Distribution の互換性を確保している。
- 省電力のための Power Monitor と Analyzer、そして、電力使用を制御する Power Knob を搭載している。
- Power Knob としては、CPU コア内で命令デコーダ幅、スカラーパイプライン数、ベクターパイプライン数、周波数制御の他、HBM2 バンド幅を 10% 単位での制御が可能である。

4. HPC アプリケーションの特性と調査手法

HPC アプリケーションの特性には、大きく計算インテンシブとメモリバンド幅インテンシブがある。これらの特性は相対的なものであり、どちらかの特性が顕著に表れることにより、それが律速となり、もう一方の特性を改善してもアプリケーションの性能が影響しなくなる。

計算性能インテンシブ: 相対的に計算性能を要求する
メモリバンド幅インテンシブ 相対的にメモリバンド幅性能を要求する

これらの特性を知ること、アプリケーションの最適化時には重要である。なぜなら、性能を律速している特性を知り、これを取り除くことが最適化の近道であるからで

ある。

この特性を知るためには、通常、性能プロファイラを用いるほか、アプリケーションを修正して特性を変更した実行バイナリにより行う手法がある。前者は市販ツールやオープンソースのツールを使うが一般的には専門の知識が必要であるため容易ではない。後者は、性能律速点を仮定して最適化を行い性能差を見るために、必ずしも合理的でないのが問題である。

5. A64FX による HPC アプリケーション特性評価

第4章で述べた HPC アプリケーション特性を調べるための手法として A64FX を用いた手法について述べる。これは、アプリケーションの2つの特性を調べる手法として、A64FX の持つハードウェアで意図的に演算能力やメモリバンド幅を制限することによる性能差を観測し、アプリケーションの性能特性を調べるものである。Arm v8-A SVE 仕様により、実行バイナリを修正することなく実行時にベクタ長を変更し演算性能を変化させ、さらに、A64FX のメモリバンド幅制限機構を用いることにより、同様に実行バイナリを修正することなく、メモリバンド幅も変化させることが可能である。

A64FX の持つハードウェアを用いて HPC アプリケーション評価を試行するために各著名ベンチマークプログラムを用いて行った。ベンチマークプログラムとして以下のプログラムを用いた。

STREAMS Triad: メモリアクセス主体のベンチマークで $a[i] = b[i] + c[i] * \text{SCALAR}$ 2つの行列の片方に定数を掛けたものをもう一つの行列に足し合わせ、3つ目の行列に格納するものである。[5]

DGEMM: 倍精度の行列の乗算アルゴリズムであり、ベクタ演算において高効率で実行できる線形代数の基本ルーチンとして知られている。最適化されたルーチンは計算インテンシブな特性を持つ。

Himeno Benchmarks: 著名な流体系ベンチマーク、メモリバンド幅律速になることが多い [6]

NAS Parallel Benchmarks: 流体系の8つの計算コアプログラムによるベンチマークプログラム集、プログラムにより特性が異なる。[7]

5.1 評価環境と測定パラメータ

本章では、可変ベクトル長バイナリを用いた A64FX の評価環境と評価パラメータについて述べる。

評価環境を以下に述べる。

Hardware: A64FX Prototype System (Single Node)

Compiler: Fujitsu Compiler(Development Version)

それぞれは、可変長バイナリを生成するオプションで実行バイナリを生成している。

評価パラメータとしてベクタ長とメモリバンド幅を以下のように設定した。

ベクタ長: 128, 256, 512 (ビット)

メモリバンド幅: 20, 40, 60, 80, 100 (%)

Arm v8-A+SVE 環境においては、SIMD 計算のベクタ長は実行時に決定される。ベクタ長は Arm v8+SVE 仕様で規定される特殊レジスタで設定される。具体的には Linux においてはプロセッサ制御用のシステムコールで設定される [4]。このため、ベクタ長を変更するためにプログラムを作成して利用している。また、メモリバンド幅はジョブプログラム実行コマンドのオプションの設定で変更している。

以上の評価環境と評価パラメータを用いて著名ベンチマークの評価を行った。

5.2 STREAMS Triad

図3,4に STREAMS Triad の評価結果を示す。これらの図はベクタ長 512 ビット、メモリバンド幅 100%を基準の 100とした相対性能を示す。図3が1 Threadでの結果、図4が12 Thread(1 CMG)の結果である。これ以降すべてのグラフにおいて横軸はメモリバンド幅の相対値、縦軸は性能の相対値となる。

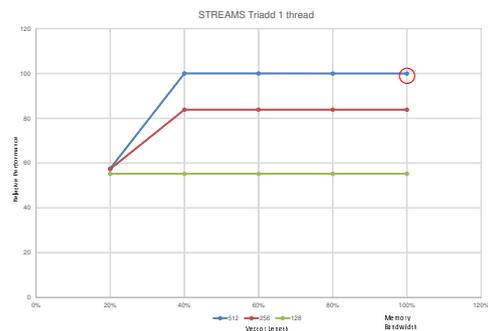


図3 STREAMS Triad 1 thread 512bit SIMD Relative

図3より、STREAMS Triad の1 Threadにおいては、メモリバンド幅が20%の結果以外はベクタ長を変更することによりベンチマーク性能の向上が見られる。つまり、A64FX 1CMGのメモリバンド幅の40%以上あれば、計算インテンシブとなり1 Threadでは十分ベンチマーク性能を引き出すことができると言える。

一方、図4のSTREAMS Triad 12 Threadの評価結果においては、ベクタ長を変更しても性能に変化がなく、メモリバンド幅が大きくなるにつれて性能が向上している。これは、12 Threadでの実行においては、メモリバンド幅の制約によりベクタ長を増やしてもベンチマーク性能が向上しないことを意味する。メモリインテンシブなプログラムでありメモリ律速な状況と言え。

同じ結果を相対性能をベクタ長 128 ビット、メモリバン

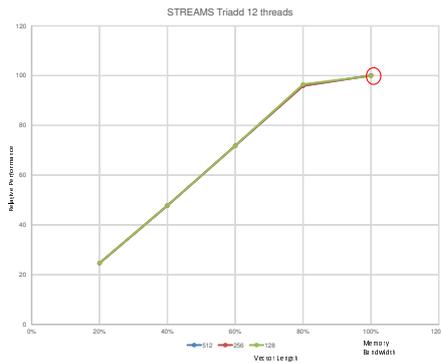


図 4 STREAMS Triad 12 thread (1 CMG) 512bit SIMD Relative

ド幅 100%を基準の 100 とした結果を図 5,6 に示す。図 5 が 1 Thread での結果においては、ベクタ長 128 ビットに比べて、ベクタ長 256 ビットの結果は最大 54%性能向上し、ベクタ長 512 ビットの結果は最大 80%性能向上することがわかる。

また、図 6 の 12 Thread(1 CMG) の結果においてはベクタ長の変更にも関わらずメモリ律速でベクタ長増加による性能が全く変化しないことがわかる。

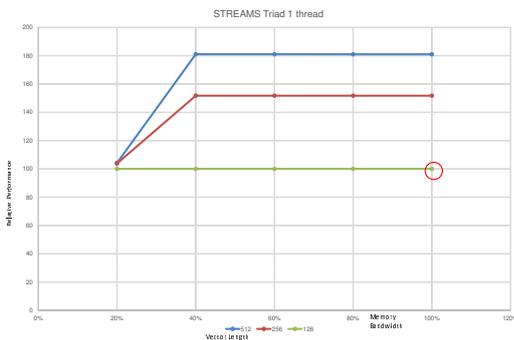


図 5 STREAMS Triad 1 thread 128bit SIMD Relative

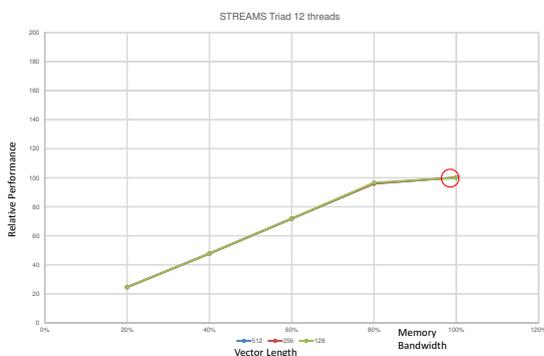


図 6 STREAMS Triad 12 thread (1 CMG) 128bit SIMD Relative

5.3 DGEMM

図 7 に DGEMM 12 Thread の評価結果を示す。これらの図はベクタ長 512 ビット、メモリバンド幅 100%を基準の 100 とした相対性能を示す。

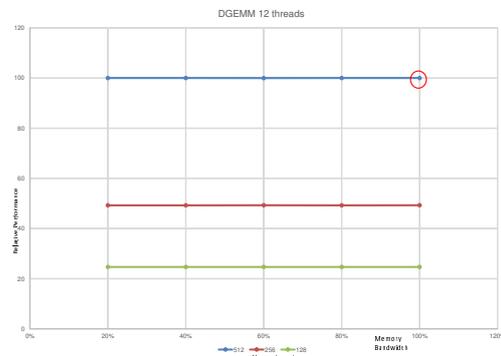


図 7 DGEMM 12 thread (1CMG)

図 7 よりわかることは、ベクタ長に比例した性能となっており、512 ビットの時を 100 とすると 256 ビット時に 1/2、128 ビット時に 1/4 となっている。更にはメモリバンド幅の変化については 20%に絞ったときでさえ、性能がほとんど変化しないほど計算インテンシブなプログラムとしては理想的な結果である。

5.4 Himeno Benchmarks

図 8,9 に Himeno Benchmarks の評価結果を示す。これらの図はベクタ長 512 ビット、メモリバンド幅 100%を基準の 100 とした相対性能を示す。

図 8 が 12 Thread(1 CMG) での結果である。図 8 より、まずベクタ長の変化により大きく性能が変化している点、さらにはベクタ長 512 ビット時においてはメモリバンド幅 20%時の性能低下が大きい点がある。ベクタ長が大きくなるほど、それに応じたメモリバンド幅が必要となるプログラムであることがわかる。

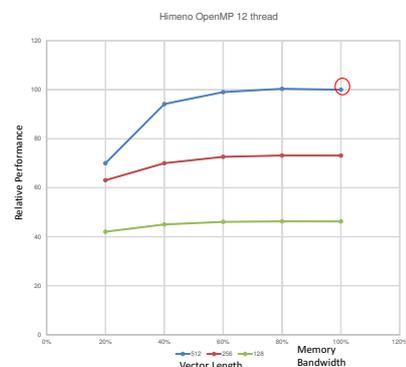


図 8 Himeno Benchmark 12 thread (1 CMG)

以上の傾向は Thread 数を増加した場合に顕著に表れる。

図9が48 Thread(4 CMG)の結果である。図9の結果においてはベクタ長128ビットにおいても20%までメモリバンド幅を絞ると性能低下が観測され、ベクタ長256ビット、512ビットの場合にはメモリバンド幅が増すにつれプログラム性能が向上する結果となった。

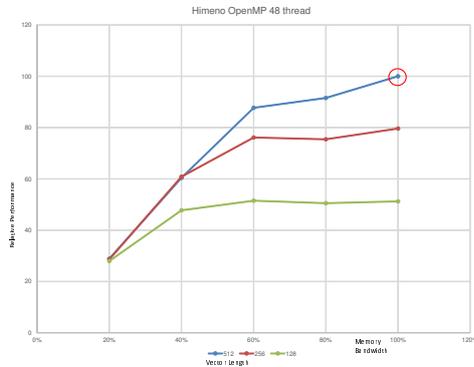


図9 Himeno Benchmark 48 thread (4 CMG)

5.5 NAS Parallel Benchmarks

NAS Parallel Benchmarksは8つのアプリケーションから構成され、それぞれにベクタ長とメモリバンド幅への要求特性が異なる。本節では、評価結果を元に幾つかの特性に分類して報告する。本節の評価結果はベクタ長512ビット、メモリバンド幅100%を基準の100とした相対性能である。またすべての結果はNAS Parallel Benchmark Open MPI版で48 ThreadクラスCの結果である。

5.5.1 ベクタ長とメモリバンド幅の影響なし

性能評価した結果、ベクタ長、メモリバンド幅の変化により相関関係がなかったものとして図10にEP, 図11にISの結果を示す。

図10にEPについては、調べた結果ループ内に関数呼び出しがありベクタ最適化が行われていなかったことがわかった。

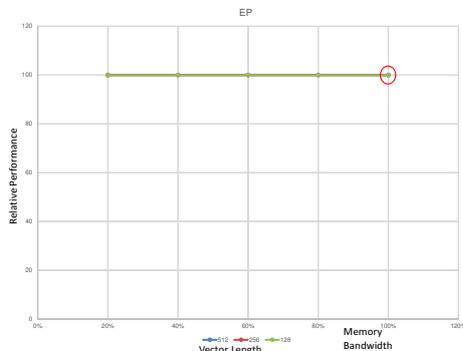


図10 NAS Parallel Benchmarks EP 48 thread

図11にISの結果については、プログラム自体が整数ソートであり整数部についてベクタ最適化が行われていな

いため、性能差がない結果となった。

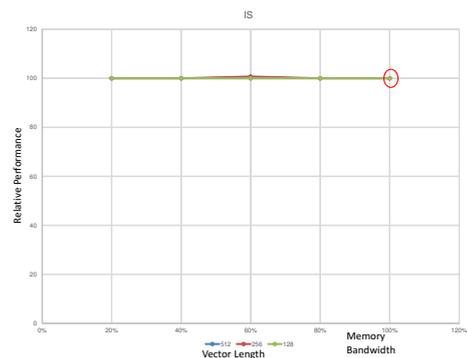


図11 NAS Parallel Benchmarks IS 48 thread

5.5.2 メモリバンド幅の影響なし：計算インテンシブ

性能評価した結果、メモリバンド幅の変化により相関関係がなかったものとして、図12にCG, 図13にBTの結果を示す。いずれも計算インテンシブなプログラムであると言える。

図12はCGの結果であるが、メモリバンド幅の変化には性能値に変化がなかった。ベクタ長256ビットまでは性能向上が著しいが、512ビットにおいては性能向上率が低下している。

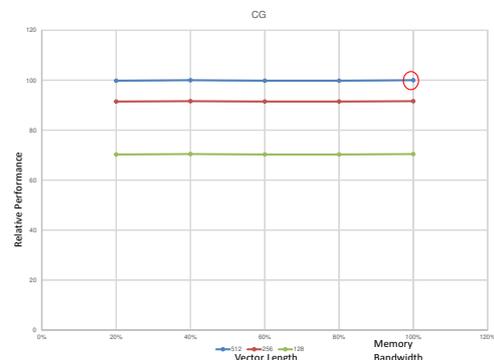


図12 NAS Parallel Benchmarks CG 48 thread

図13はBTの結果であるが、CGの結果に比べてベクタ長512ビットの際も比例した性能となっている。ベクタ長512ビット、メモリバンド幅20%の場合に5%程度性能劣化が見られる。

5.5.3 ベクタ長とメモリバンド幅の影響あり

性能評価した結果、メモリバンド幅の変化により相関関係があったものとして、図14にSP, 図15にMG, 図16にFT, 図17にLUの結果を示す。

(1)

以上のように、アプリケーションの特性に応じてベクタ長とメモリバンド幅を変化させることにより、プログラムへの影響度を評価できることがわかる。

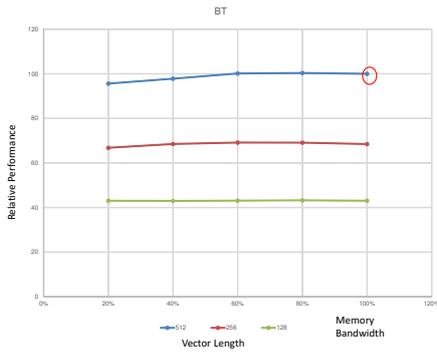


図 13 NAS Parallel Benchmarks BT 48 thread

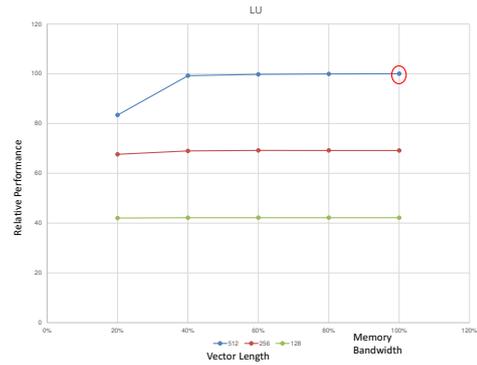


図 17 NAS Parallel Benchmarks LU 48 thread

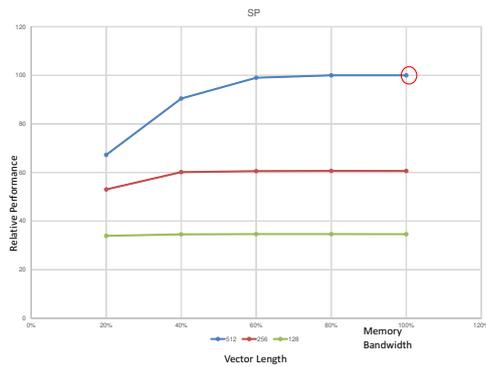


図 14 NAS Parallel Benchmarks SP 48 thread

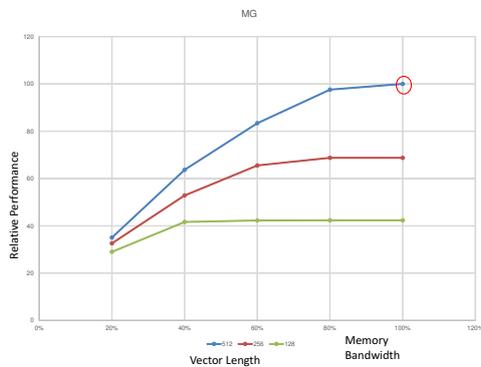


図 15 NAS Parallel Benchmarks MG 48 thread

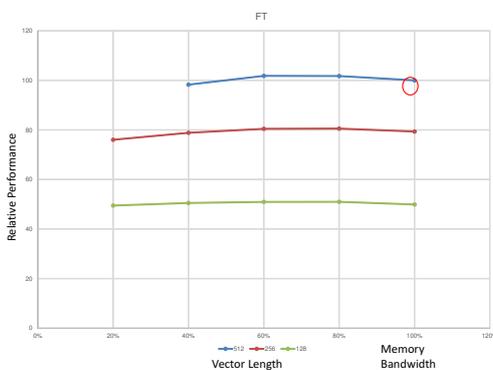


図 16 NAS Parallel Benchmarks FT 48 thread

6. おわりに

本稿では、Arm SVE プロセッサである A64FX 上で可変ベクトル長バイナリを用いた著名 HPC ベンチマークプログラムの評価について報告した。A64FX は Arm SVE アーキテクチャに対応し、かつ、メモリバンド幅を制御する機能を持っているため、ハードウェアで意図的に演算能力やメモリバンド幅を制限できる。

本機能を用いて各種著名ベンチマークを用いて、計算インテンシブかメモリインテンシブなのかを評価した。評価の結果、各ベンチマークプログラムの特性からみて妥当な結果を得た。

今後は、この手法を用いて実アプリケーションの評価を行う予定である。

参考文献

- [1] Technology Update: Scalable Vector Extension (SVE) for Armv8-A available from <https://community.arm.com/developer/tools-software/hpc/b/hpc-blog/posts/technology-update-the-scalable-vector-extension-sve-for-the-armv8-a-architecture> .
- [2] The ARM Scalable Vector Extension available from <https://alastairreid.github.io/papers/sve-ieee-micro-2017.pdf>
- [3] Specification of the ARM Scalable Vector Extension available from <https://developer.arm.com/docs/ddi0584/latest/arm-architecture-reference-manual-supplement-the-scalable-vector-extension-sve-for-armv8-a>
- [4] Scalable Vector Extension (SVE) core support for Linux: available from <https://lwn.net/Articles/717804>
- [5] STREAMS Benchmark: available from <https://www.cs.virginia.edu/stream/>
- [6] Himeno Benchmark Results: available from <http://w3cic.riken.go.jp/HPC/HimenoBMT/himenoDB1.pdf>
- [7] NAS Parallel Benchmarks: available from <https://www.nas.nasa.gov/publications/npb.html>