Vol.2021-HPC-178 No.5
2021/3/15

A64FX プロセッサにおける Fiber ミニアプリスイートの性能評価

辻 美和子[1,a)]　佐藤 三久[1]

概要：本稿では，Fiber ミニアプリスイートを用いて，スーパーコンピュータ富岳における A64FX プロセッサの性能評価を行った．各アプリケーション・データセットについて，さまざまな MPI プロセス数，OpenMP スレッド数，MPI プロセス割り当て手法，OpenMP スレッド割り当て手法を検討した．また，Intel Skylake プロセッサおよび Marvell ThunderX2 プロセッサを持つシステムとの性能の比較を行った．A64FX は，TX2 や SKL と比較して，一部のアプリケーションの小さいデータセットを実行したときに性能が低かったものの，他のアプリケーションやデータセットでは，ノードあたりの演算理論ピーク性能に勝る SKL と同等もしくは高い性能を示した．

Performance Evaluation of A64FX Arm Processor for the Fiber Miniapp Suite

Abstract: In this paper, we have evaluated the performance of A64FX processor in supercomputer Fugaku using Fiber miniapp suite. We have ingestigated various numbers of MPI processes, OpenMP threads as well as different methods to assign MPI processes and OpenMP threads. Moreover, we have compared the performance of A64FX with those of Intel Skylake and Marvell ThunderX2. Whereas the performance of A64FX is not good for some applications with small data sets, it better than or compatible with the Intel Skylake for others.

1. はじめに

理化学研究所計算科学研究センターを中心として開発されたーパーコンンピュータ「富岳」は，「京」コンピュータの後継機であり，2020 年より試験運用を，2021 年 3 月より正式運用を開始した．富岳は，Tofu ネットワークなど京の特徴を継承しつつ，プロセッサを SPARC-V9 アーキテクチャに基づくものから Armv8.2-A アーキテクチャ・Scalable Vector Extension (SVE) 拡張に基づくものに，メモリを DDR3 から HBM2 に変更するなど，最新のテクノロジによる高い演算性能とメモリバンド幅を実現したシステムである．

富岳は 2021 年の正式運用開始に向けて，共用前評価のための運営を行っている[*1]．本稿では，富岳の共用前評価環境においてミニアプリケーションセット Fiber[6] の各アプリケーションを実行し，性能評価を行った．

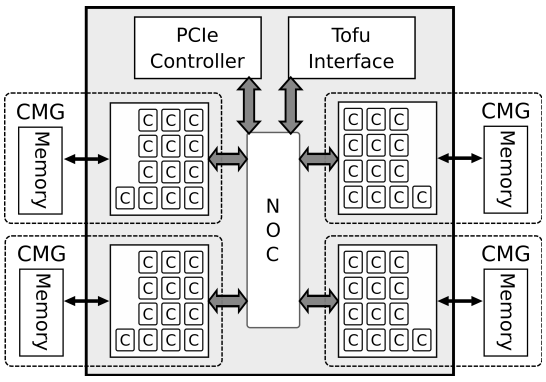

図 1　A64FX Overview

2. A64FX

2.1 ハードウエア諸元

本章では，本評価に用いた A64FX プロセッサ [3] の概要について述べる．A64FX は，Armv8.2-A 命令セットアーキテクチャーを HPC むけに拡張した Scalable Vector

[1] 理化学研究所計算科学研究センター
RIKEN Advanced Institute for Computational Science
[a)] miwako.tsuji@riken.jp

[*1] 本稿が出版される頃には正式運用を開始しているはずである

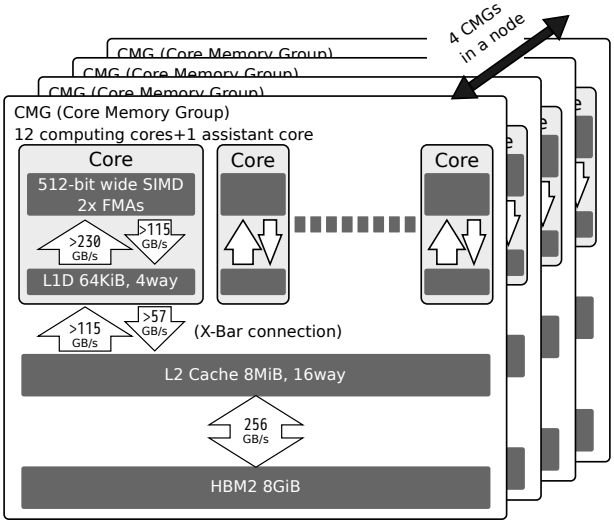

図 2 The block diagram of the internal architecture of the A64FX

Extension (SVE) に基づいて設計されたプロセッサである．

図 1 に A64FX プロセッサの概要を，**図 2** に詳細を示す．

1 ノードは 1 ソケットからなり，1 ソケットは 4 つの Core Memory Groups (CMG) と呼ばれる L2 キャッシュおよび HMB2 メモリコントローラを共有するコアグループからなる．1CMG には 12 個の計算コアと 1 もしくは 0 個のアシスタントコアが含まれる．CMG はリング上の Network On Chip (NOC) により接続されている．ある CMG に属するコアが，他の CMG にある L2 キャッシュやメモリにアクセスする場合は，NOC を経由してアクセスする．インターコネクトへのアクセスついても同様に NOC を経由する．CMG 内のコアはクロスバーを介して L2 キャッシュを共有している．

各コアは 512bit SIMD の Fused Multiply Add (FMA) が可能な浮動小数点演算器を 2 つ持つ．倍精度の場合 1 サイクルで実行される浮動小数点演算数は最大で 32 であるから，1 秒間に 1 ノードが実行する浮動小数点演算数は最大で

$$32 \times 12(\text{cores}) \times 4(\text{CMGs}) \times \text{frequency}$$

である．富岳はノーマルモード (2.0GHz) ／ブーストモード (2.2GHz) のように周波数を選択できるが，周波数が 2GHz の場合はノードあたり 3.072TFLOPs である．各コアは 64KiB の L1 キャッシュを持ち，CMG 内の 12 コアは 8MiB の L2 キャッシュを共有する．各 CMG は 256GB/s でアクセス可能な 8GiB の HBM2 メモリへのメモリコントローラを持つ．ノード全体のメモリ容量は 32GiB，メモリバンド幅は 1024GB/s である．

2.2 コンパイラ

スーパーコンピュータ富岳では以下のコンパイラが利用可能である：

- 富士通コンパイラ (C/C++/Fortran) trad モード
  富士通コンパイラテクノロジに基づくコンパイラ．A64FX のパフォーマンスを最大限に引き出すコードを生成する
- 富士通コンパイラ (C/C++) clang モード
  オープンソースのコンパイラインフラストラクチャである LLVM/Clang に基づき富士通が開発したコンパイラ．trad モードより新しい言語仕様をサポートする
- Clang/LLVM (C/C++)
  オープンソースのコンパイラ
- GCC (C/C++/Fortran)
  オープンソースのコンパイラ
- Arm Compiler (C/C++/Fortran) オープンソースのコンパイラインフラストラクチャである LLVM/Clang/Frang に基づき Arm が開発したコンパイラ

本稿では，主に富士通コンパイラ (C/C++/Fortran) trad モードを取り扱うものとする．

2.3 Affinity

本節では，A64FX プロセッサがサポートするスレッド・アフィニティー・インターフェイスについて述べる．富士通コンパイラ trad モードでコンパイルされた OpenMP のプログラムにおいて，実行時にコアとスレッドをバインドするには，環境変数 `FLIB_CPU_AFFINITY` もしくは `FLIB_CPUBIND` を用いる．`FLIB_CPU_AFFINITY` では，スレッド番号 0 版から番に割り当てるコアの番号を記述する方法と，割り当てるコアの範囲とそのインターバルを記述する方法がある．前者の場合，たとえば `FLIB_CPU_AFFINITY="12,14,15,22,...,59"` のように記述する．後者の場合，`FLIB_CPU_AFFINITY="12-59:12"` のように:の後に増分を記述する．この例では 1CMG1 スレッドのみを割り当てている．各 CMG にインターバルする場

合は
`FLIB_CPU_AFFINITY="12-48:12,13-49:12,14-50:12,15-51:12,16-52:12,17-53:12,18-54:12,19-55:12,20-56:12,21-57:12,22-58:12,23-59:12"` とする．

指定されるコア数がスレッド数より少ない場合，スレッドは最初に指定されたコア番号に戻る．ここでコア番号が 12 から始まることに注意されたい．

`FLIB_`のほかに，`GOMP_`を用いる指定方法もあるが，本稿では扱わない．

富岳では，1CMG 内に 1 プロセス 12 スレッドを立ち上げる OpenMP/MPI ハイブリッドプログラミングでの使用を推奨している [8] が，上述のように 1 ノードに 13 スレッド以上を使用する場合に加えて，より少ないスレッド数が適したアプリケーションのために，MPI プロセスの CMG への割り当てを指示する環境変数 `OMPI_MCA_plm_ple_memory_allocation_policy` が用意されている．以下のオプションが可能である：

- `simplex`：CMG を 1MPI プロセスで専有する
- `share_cyclic`：CMG を他のプロセスと共有する．ランク番号の順に cyclic で MPI プロセスを CMG 割り当て，すべての CMG に割り当てたら最初の CMG に戻る
- `share_band`：CMG を他のプロセスと共有する．別途ジョブスクリプトで指定されるノード毎の最大 MPI プロセス数から CMG 毎の最大プロセス数を決定し，ランク番号の順にプロセスに CMG を割り当てる．CMG あたりの最大プロセス数に達したとき次の CMG に割り当てる．

図 3 に，MPI プロセスおよび OpenMP スレッドの割り当ての例を示す．この例では 1 ノードに，4 もしくは 8 MPI プロセス，各プロセスが 3 OpenMP スレッドを使用する OpenMP/MPI のハイブリッドプログラミングである．MPI プロセス割り当て手法に `simplex` を指定した場合，1 ノードに 4 以上の MPI プロセスを割り当てることはできない．

3. 性能評価

本章では測定条件および実験結果について述べる．なお，富岳共用前評価環における評価結果は，スーパーコンピュータ「富岳」の供用開始時の性能などの結果を保証するものではない．また，すべてのアプリケーションは，広範に使用されるコンパイラ最適化オプションを付加した上で As Is でコンパイルし，A64FX に向けたチューニングは行っていない．他のシステムでの比較実験についても同様である．

3.1 Fiber miniapp suite

Fiber miniapp suite[6] は，Feasibility Study on Futre HPC Infrastractures [2] のアプリケーション部会で検討されたフルアプリケーションから抜粋されたミニアプリケーションの集合である．**表 1** に，Fiber に含まれるミニアプリケーションとその概要を示す．

CCS-QCD[1] は，格子 QCD クローバ・フェルミオンソルバである．FFVC-MINI は，三次元非定常非圧縮性熱流体シミュレーションを行う．NICAM-DC-MINI[9] は，非静力学正 20 面体格子大気モデル Non-hydrostatic ICosahedral Atmospheric Model (NICAM) より抜き出したミニアプリである．mVMC-MINI は，電子系の基底状態を表現する仮想波動関数を構成することで強相関電子系の物理的特性を分析する．NTChem-MINI は，第一原理量子化学計算により分子や電子の構造を計算する．FrontFlow/blue (FFB)-MINI は，有限要素法による熱流体解析プログラムである．これらのアプリケーションのうち，ゲノムシーケンスの解析を行うアプリケーションである NGS-Analyzer は，I/O インテンシブであることが知られているため，プロセッサへの評価を主体とした本実験では対象にしなかった．

3.2 システム諸元

比較のために本実験では，仏・原子力代替エネルギー庁 (CEA) の Inti スーパーコンピュータを用いた．Inti は，アプリケーションやシステムソフトウエアなどをさまざまなシステムで開発し評価するための複合的システムで，SKL, KNL および V100 などの異なるアーキテクチャのパーティションからなる．本稿では，Arm ThunderX2 プロセッサ (以下，TX2 と略す) からなるパーティションと，Intel Skylake プロセッサ (以下，SKL と略す) からなるパーティションを用いた．

富岳を含めたシステムの諸元を**表 2** にまとめた．

性能の比較においては，同じアプリケーション，データセットであっても使用されるノード数やコア数が異なることがあること，A64FX はノードあたり 1 ソケットに対して TX2 と SKL は 2 ソケットであること，TX2 は 128bit SIMD であるのに対して SKL と A64FX は 512bit SIMD であること，すべてのプロセッサで周波数が異なること，に注意されたい．

3.3 CCS-QCD

CCS-QCD[1] は，格子 QCD クローバ・フェルミオンソルバである．

本実験では，**表 3** に示すように，class1 および class2 の 2 つの異なるサイズのデータを対象とした．

図 4 に，CCS-QCD class1 を A64FX の 1 ノードで実行した結果を示す．MPI プロセス数は 1 に固定されている．OpenMP スレッド数を 1 ～ 48 とし，スレッド間隔も適宜変化させた．図から，スレッド数が最大の 48 のときに最大の性能を示した．また，スレッド間隔が狭いほうがより高

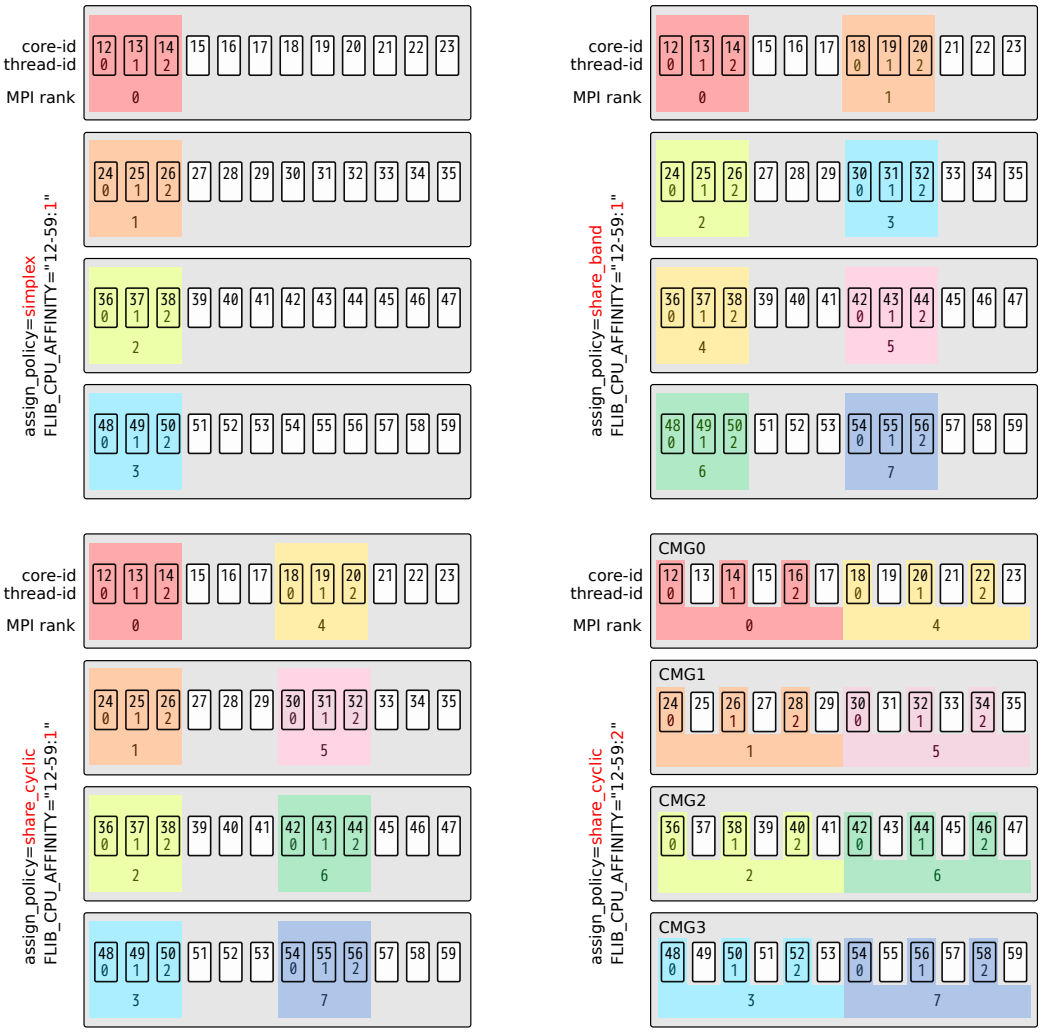

図 3 Examples of MPI rank and OpenMP thread assignments. In case of 4/8 MPI processes and 3 OpenMP threads. simplex and 12-59:1 (top, left), share_band and 12-59:1 (top, right), share_cyclic and 12-59:1 (bottom, left) and share_cyclic and 12-59:2 (bottom, right)

表 1 Fiber miniapp suite

Application	Area	Characteristics
CCS-QCD 1.2.1	Quantum chromodynamics	structured grid Monte Carlo
FFVC-MINI 1.0.1	thermo-fluid analysis	3 dimensional cavity flow
NICAM-DC-MINI 1.0	climate	structured grid, stencil
mVMC-MINI 1.0	material science	many variable variational Monte Carlo
NGS-Analyzer-MINI 1.0.1	analyzes output data generated by a next-generation genome sequencer	workflow of multiple tasks I/O intensive
NTChem-MINI 1.1	quantum chemistry	dense matrix calculation
FFB-MINI 1.0.1	thermo-fluid analysis	finite element method, unstructured grid

い性能を示したが，差は僅かであった．**図 5** に，CCS-QCD class2 を A64FX の 8 および 16 ノードで実行した結果を示す．MPI プロセス数はいずれも 64 であり，OpenMP スレッド数は 6 および 12 である．すべてのコアが使用されるため，スレッド間隔は常に 1 である．OpenMP スレッド数が 6 のとき，1CMG に 2MPI プロセスが実行される．プロセスの配置方式は share_band および share_cyclic を比較したが，これらによる違いはほぼ見られなかった．また，ノード数が 2 倍のとき性能は 1.5 倍程度となった．

図 6 に，CCS-QCD の class1 と class2 を TX2, SKL および A64FX で実行した結果を示す．スレッド数および Affinity はそれぞれのプロセッサで最も高い性能を示したものが選ばれた．他の設定のときの TX2 および SKL の性能については，著者らによる関連研究 [13] に記されている．他のアプリケーションも同様である．class1 のときは SKL が 1 ノード 48 コア中 24 スレッドを用いてもっとも高い性

表 2 Arm & Intel SKL partitions in the Inti supercomputer and the supercomputer Fugaku

	Arm Cluster	Intel Cluster	Fugaku
CPU	Marvell ThunderX2	Intel Xeon Platinum 8168 (Skylake)	A64FX
# of cores / socket	32	24	48
# of sockets	2	2	1
SMT	**2**	**2**	1
CPU GHz	2.2 GHz	2.7 GHz	2.0 GHz
Peak performance / node	1126.4 GFlops	4147.2 GFlops	3072.0 GFlops
Memory	DDR4	DDR4	HBM2
Capacity / node	256 GB	192 GB	32 GB
Bandwidth / node	320 GB/s		1024 GB/s
# of nodes	28	22	158,976
Compiler	arm-compiler 19.0.0	intel/17.0.6.256	tcsds-1.2.28a tcsds-1.2.29
MPI library	openmpi 2.0.4	mpi/openmpi/2.0.4	tcsds-1.2.28a tcsds-1.2.29

表 3 CCS-QCD のデータセットと実験環境

	class1	class2	
Lattice Size	8x8x8x32	32x32x32x32	
ノード数	1 (48 cores)	8 (384 cores)	16 (768 cores)
MPI プロセス数	1	64	64
OMP スレッド数	1 ∼ 48	6	12

表 4 FFVC のデータセットと実験環境

	256	1024
Global Domain	256x256x256	1024x1024x1024
ノード数	1 (48 cores)	8 (384 cores)
MPI プロセス数	8	64
OMP スレッド数	2 ∼ 6	1 ∼ 6

表 5 mVMC のデータセットと実験環境

	tiny	middle	
NSplitSize	1	4	
ノード数	1 (48 cores)	4 (192 cores)	8 (384 core)
MPI プロセス数	1	64	64
OMP スレッド数	1 ∼ 48	1 ∼ 3	1 ∼ 6

能を示す一方，class2 では A64FX と SKL がもっとも高い性能を示した．ただし，class2 のとき，TX2 は 4 ノードであるのに対して，SKL は 6 ノード 288 コア中の 256 コアを使用した場合，A64FX は 8 ノード 384 コアを使用している．また，SKL のあたりのノードピーク性能は A64FX の約 4/3 倍である．

3.4 FFVC

FFVC-MINI は，三次元非定常非圧縮性熱流体シミュレーションを行うアプリケーションである．

表 4 に，本実験で用いたデータおよびパラメータを示す．データサイズは 256^3 および 1024^3 とした．

図 7 および**図 8** にデータセット 256^3 および 1024^3 に対する実験結果を示す．いずれの場合も，スレッド数の増加に対して良好なスケーリングを示し，すべてのコアを使用した場合がもっとも性能が高かった．また，スレッド割り当てインターバルおよび MPI プロセス割り当て手法による性能の差は小さかったが，share_band が僅かに share_cyclic を上回った．

図 9 に同じデータセットに対する TX2, SKL, および A64FX の性能比較を示す．前者はすべて 1 ノードでの結果であるが，後者では TX2 と SKL が 4 ノードを用いたが，A64FX では 8 ノードを用いた．これは，このデータセットが，A64FX の 4 ノードでの利用可能メモリ量をわずかに上回るメモリ量を必要としたためである．データセット 1024^3 において，A64FX は使用ノード数は TX2 や SKL の倍であるものの性能は 2.7 倍であり，高い性能を示していることがわかる．FFVC は CCS-QCD よりも要求 BF が高く [11]，A64FX の高バンド幅が効果的であったと考えられる．

3.5 mVMC

mVMC-MINI は，電子系の基底状態を表現する仮想波動関数を構成することで強相関電子系の物理的特性を分析する．**表 5** に本実験で用いたデータセットと並列パラメータを示す．

図 10 に，データセット tiny を A64FX の 1 ノードで実行した結果を示す．スレッド数は 1 ∼ 48 とし，スレッド数 2 ∼ 24 のときは複数のスレッド間隔で評価したが，8 スレッドを間隔 1 で実行したときが最大の実行効率を示した．これは，データセットが非常に小さく，スレッド並列化のオーバーヘッドのほうが大きくなるからであると考えられる．データセット tiny に関しては，他のシステムでも同様の傾向を示した [13]．またスレッド数が同じとき，スレッ

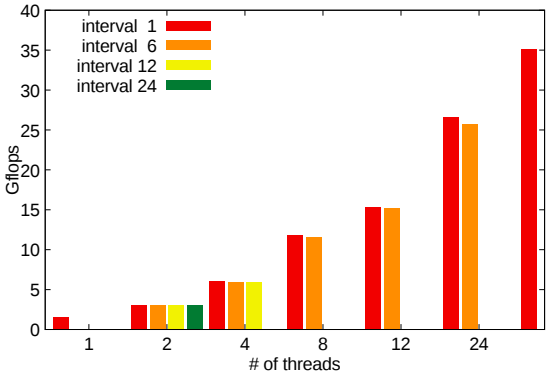

図 4 CCS-QCD class1, A64FX for thread interval 1, 2, 4, 8, 12, 48. Y-Axis is GFlops, X-Axis is the number of threads

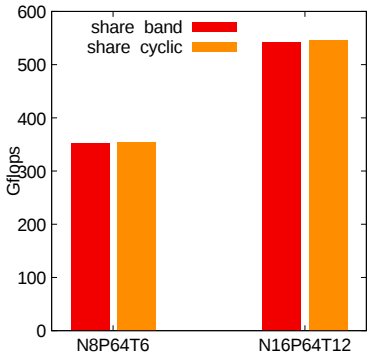

図 5 CCS-QCD class2 on A64FX. 8/16 nodes.
Y-Axis is GFlops, In the X-Axis, N is the number of nodes, P is the number of processes and T is the number of threads

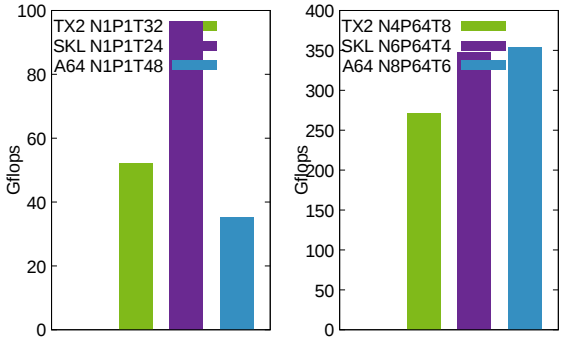

図 6 CCS-QCD class1(left) and 2(right) on TX2, SKL and A64FX. 凡例の N はノード数，P は MPI プロセス数，T は OpenMP スレッド数

ド間隔が小さいほうが高い性能を示し，スレッド間隔の差による性能差も他のアプリケーションよりも大きかった．

図 11 に，データセット middle を A64FX の 8 ノードで実行した結果を示す．データセット middle については，データセット tiny の結果をふまえてスレッド間隔はすべて 1 とした．また，MPI プロセス割当について share_band と share_cyclic の 2 通りを調査したが，share_cyclic のほうが高い性能を示した．

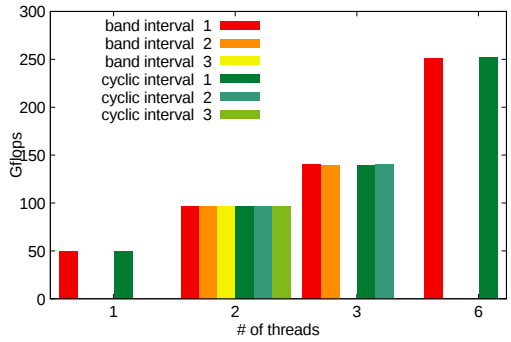

図 7 FFVC 256^3, A64FX for each thread interval
Y-Axis is GFlops, X-Axis is the number of threads

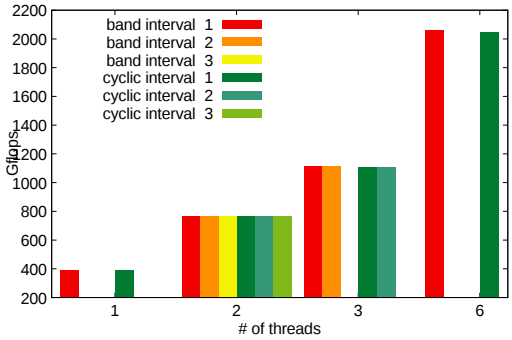

図 8 FFVC 1024^4, A64FX. 8 nodes for each thread interval
Y-Axis is GFlops, X-Axis is the number of nodes, processes and threads.

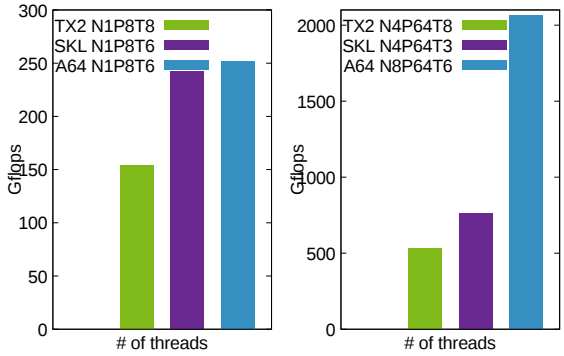

図 9 FFVC 256^3(left) and 1024^3(right) on TX2, SKL and A64FX. 凡例の N はノード数，P は MPI プロセス数，T は OpenMP スレッド数

図 12 に，TX2, SKL および A64FX の 4 ノードでデータセット middle を実行したときの比較を示す．MPI プロセスはすべて 64 で，OpenMP スレッド数はそれぞれ最も高い性能を示したケースを選択した．TX2 が 8，SKL が 6，A64FX が 3 である．ここで，TX2 と SKL はハイパースレッディングが有効であり，1 コアあたり 2 スレッドが割り当てられている．SKL がもっとも高い性能を示し，A64FX はもっとも性能が低い性能を示した．

3.6 NTChem

NTChem は，第一原理量子化学計算により分子や電子

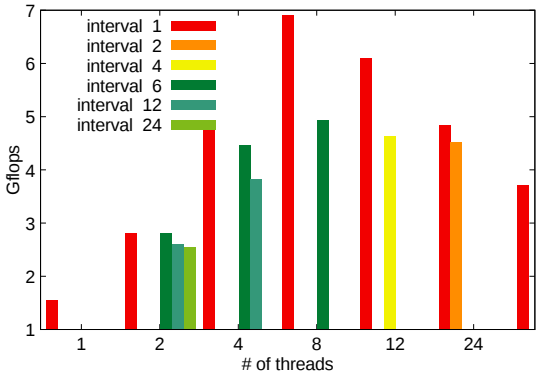

図 10 mVMC, tiny, A64FX for each thread interval
Y-Axis is GFlops, X-Axis is the number of threads

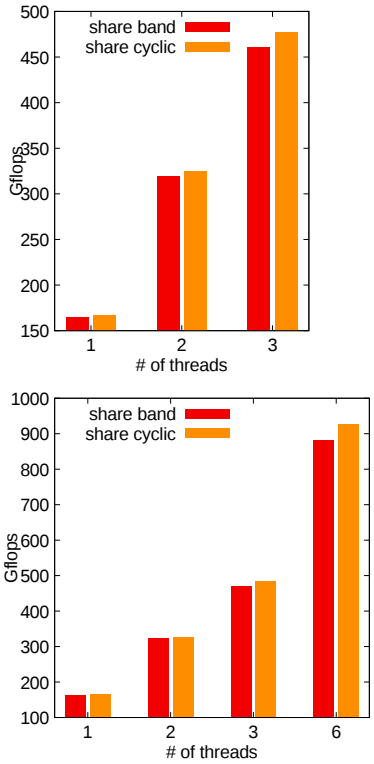

図 11 mVMC middle. A64FX, 4nodes (top) and 8 nodes (bottom).
Y-Axis is GFlops, X-Axis is the number of threads

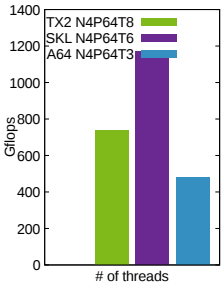

図 12 mVMC middle, TX2, SKL and A64FX

の構造を計算する．NTChem では行列の計算を多く行っており，BLAS ライブラリの性能が，NTChem の実行効率に大きく影響することが知られている．

表 6 NTChem のデータセットと実験環境

	h2o	taxol
ノード数	1 (64 cores)	4 (192 cores)
MPI プロセス数	1 ∼ 6	4 ∼ 24, 48
OMP スレッド数	4, 8	4, 8
BLAS library	SSL2BLAMP	SSL2BLAMP

表 6 に本実験で用いたデータセットと並列パラメータを示す．

図 13 にデータセット H2O に対する A64FX1 ノードの結果を示す．上が 4 スレッド，下が 8 スレッドであり，MPI プロセス数は 1 ∼ 12 および 1 ∼ 6 に変化させた．4 スレッド 12 プロセスのほうが，8 スレッド 6 プロセスよりも高い性能を示した．これは 8 スレッドの場合，同じプロセスのスレッドが複数 CMG にまたがって実行されるからであると考えられる．MPI プロセス割り当てとしては，share_band のほうが高い性能を示した．また，NTChem は計算の多くを数値計算ライブラリの行列演算 (DGEMM) が占めることから，高度にチューニングされたライブラリにより高いピーク性能比が期待されたが，実行効率はピーク性能の 5 パーセント程度にとどまった．**図 14** にデータセット taxol での結果を示す．データセット H2O と同様に，MPI プロセス割り当てにおいて，share_band のほうが高い性能を示した．

図 15 に，デーセット H2O および taxol を実行した場合の TX2, SKL, A64FX の性能比較を示す．ノード数は H2O がすべて 1，taxol がすべて 4 である．MPI のプロセス数およびスレッド数は，それぞれのプロセッサで最良の結果を示した設定を選択して表示した．いずれも場合も SKL がもっとも高い性能を示し，A64FX は TX2 と同程度の性能であった．特にデータセット H2O では，A64FX は SKL と比較して低い性能を示した．

図 16 に NTChem に関する各プロセッサのカーネルの実行時間の内訳を示す．図から，A64FX では，他のプロセッサと比較して Tran3c1 の比率が大きく，小さな問題でこの傾向がより顕著である．

図 17 に，カーネル Tran3c1 および RMP2Engery の Performance Analysis (PA) データを示す．図より，Tran3c1 では RMP2Engery と比較して，浮動小数点待ち時間が長く，浮動小数点演算パイプラインの busy rate も低いことから，Tran3c1 においては浮動小数点命令のスケジューリングもしくは・およびレイテンシに問題があると考えられる．また，整数の読み込みに対する L1D キャッシュアクセス待ち時間も全体に長い．問題サイズが小さいときに，Tran3c1 の占める割合が大きくなり，A64FX ではこれに由来する浮動小数点演算待ち時間が大きくなり，全体の性能の低下につながっていると考える．一方で，主要なカーネルであり，一般に大きな問題でより大きな割合を占める

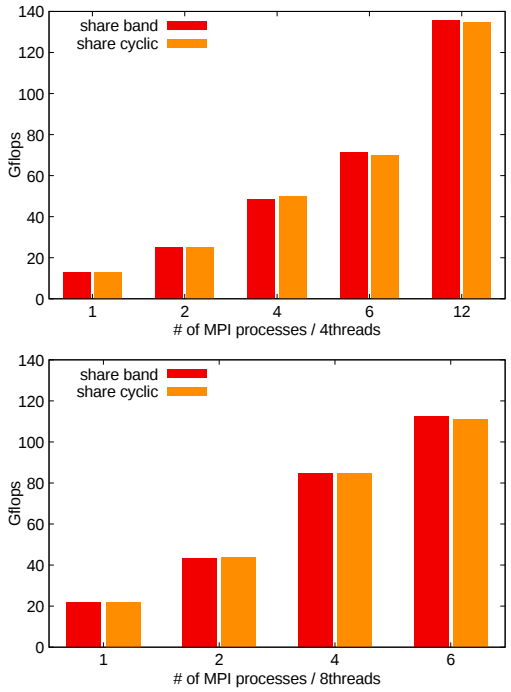

図 13 NTChem, H2O, A64FX. Using 4threads (top) and 8 threads (bottom) in a single node. Y-Axis is GFlops, X-Axis is the number of processes

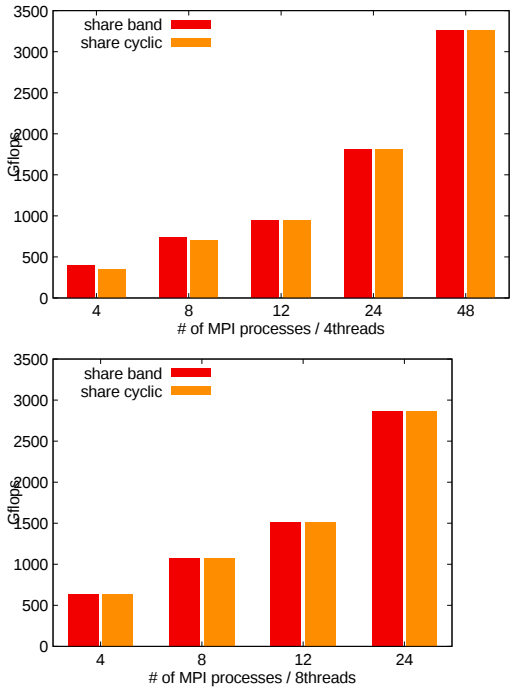

図 14 NTChem taxol, A64FX. Using 4 threads (top) and 8 threads (bottom) in 4 nodes.
Y-Axis is GFlops, X-Axis is the number of processes

RMP2Engery においては，A64FX は SKL と同等もしくはより高い性能を示している．

3.7 NICAM-DC

NICAM-DC-MINI[9] は，非静力学正 20 面体格子大気

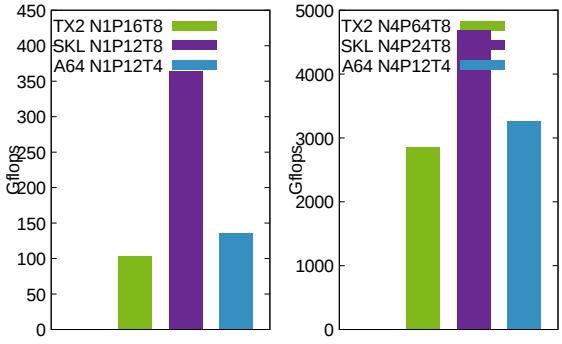

図 15 NTChem, H2O and taxol on TX2, SKL and A64FX. 凡例の N はノード数，P は MPI プロセス数，T は OpenMP スレッド数

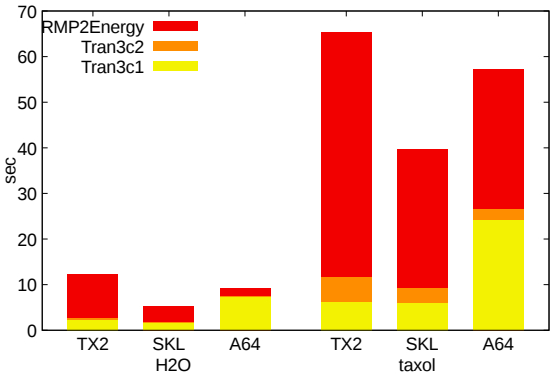

図 16 NTChem の実行時間の内訳

表 7 NICAM-DC

	gl05rl00z40	gl05r00z80
ノード数	2	1
MPI プロセス数	10	5
OMP スレッド数	1 ～ 9	1 ～ 9

モデル Non-hydrostatic ICosahedral Atmospheric Model (NICAM) より抜き出したミニアプリである．

本実験で用いたデータセットを**表 7** に示す．本実験で用いた NICAM は，OpenMP などによる陽なスレッド並列化を行っておらず，ノード内はコンパイラの自動並列化機能による．本データセットは MPI プロセス数が 10 および 5 に指定されているため，1 ノードあたりを 5 MPI プロセスとし，1 プロセスあたりの OpenMP スレッド数を 1 から 9 とした．

NICAM-DC, gl05rl00z40 の結果を**図 18** に，gl05rl00z80 の結果を**図 19** に示す．前述のように本コードは OpenMP などによるレッド並列化がなされていないが，コンパイラの自動並列化機能により生成されたバイナリは，スレッド数 4 程度まではスレッド数の増加にともなう性能向上を得た．いずれのデータセットもほぼすべての場合で，プロセス割り当て手法では，share_cyclic が share_band よりもや

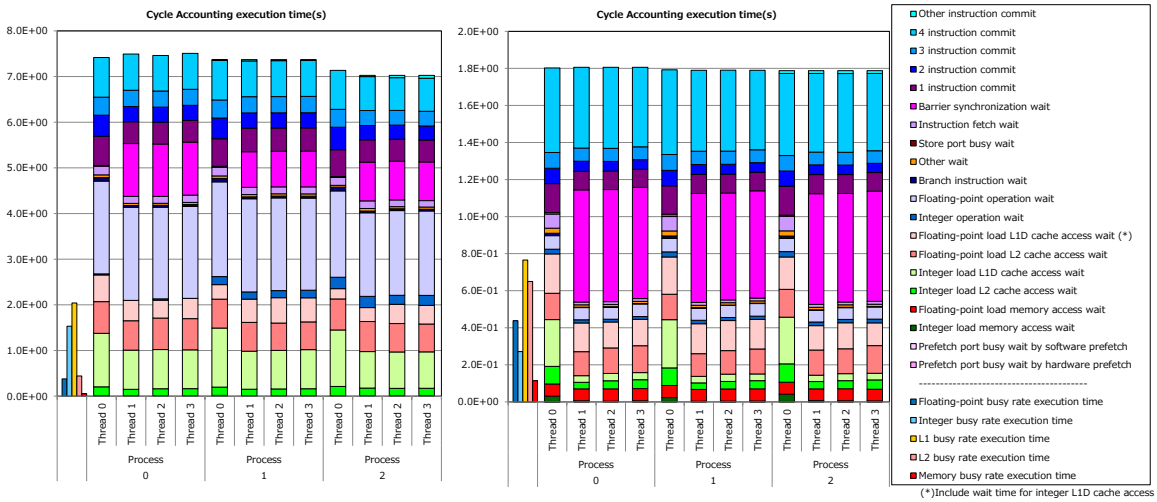

図 17 Performance Analysis (PA) of NTChem, H2O. Tran3c1 is shown at left and RMP2Engergy is shown at Right.

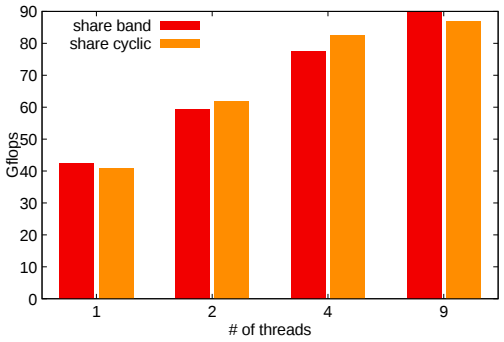

図 18 NICAM-DC, gl05rl00z40. Y-Axis is GFlops, X-Axis is the number of threads

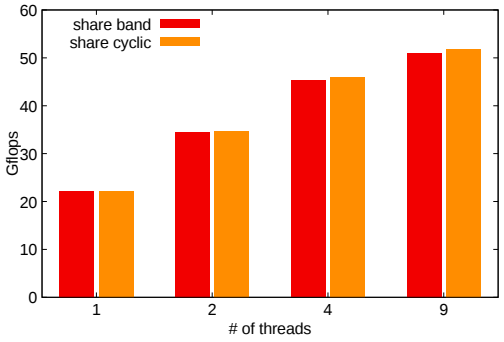

図 19 NICAM-DC, gl05rl00z80. Y-Axis is GFlops, X-Axis is the number of threads

や高い性能を示した．本コードは，TX2 および SKL のコンパイラでは自動並列化による効果がほぼ見られなかったため，比較については省略する．

3.8 FFB

FrontFlow/blue (FFB)-MINI は，有限要素法による熱流体解析プログラムである．NICAM と同様に陽なスレッド並列化がなされていないため，コンパイラの自動並列機能を用いたによるスレッド並列化を行った．**表 8** に本実験で用いたデータセットおよび実験環境を示す．

表 8 FFB

	sample
ノード数	1
MPI プロセス数	8
OpenMP スレッド数	1 ～ 6

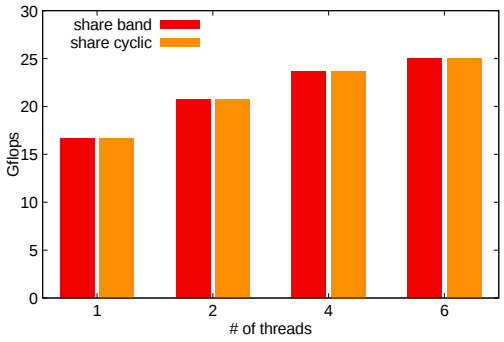

図 20 FFB, sample data. Y-Axis is GFlops, X-Axis is the number of threads

図 20 に実験の結果を示す．4 スレッドまではスレッド数の増加による性能向上が見られたが，向上の度合いは FFB と同様にコンパイラの自動並列化を用いた NICAM-DC よりも小さかった．MPI プロセス割り当て手法については，2 スレッド以外で share_cyclic のほうが高い性能を示したが，差は 1 パーセント程度だった．

3.9 議論

本実験では，OpenMP のスレッド間隔や MPI のプロセス割り当てによる性能の変化について調査した．スレッド間隔については，メモリバンド幅律速と言われているアプリケーションも含めて，スレッド間隔は近いほうが良いかもしくはスレッド間隔は性能に影響しなかった．MPI

の割り当て手法による性能の変化は全体に小さかったが，CCS-QCD，mVMC，および FFB において share_cyclic のほうが高い性能を示した．FFVC，および NTChem において share_band のほうが高い性能を示した．NICAM は問題サイズによって異なった．

他のプロセッサとの比較においては，A64FX は，NTChem と CCS-QCD の小規模データセット，および mMVC で低い性能を示した．原因としては，A64FX は命令レイテンシが大きい [3] が，小さい問題サイズにおいてこれを隠蔽する命令スケジューリングが難しいためだと考えられるが，詳しい検証は今後の課題とした．一方で，FFVC では他よりも高い性能を示した．NICAM および FFB ではコンパイラの自動並列化機能を用いた．自動最適化により生成されたバイナリは，他の環境と比較して高い性能を示したが，OpenMP による並列化を行っているアプリケーションと比較してスケーラビリティが低かった．

4. 関連研究

A64FX に関する性能評価としては，小田嶋ら [7], [12]，Jackson ら [4] によるベンチマークおよびミニアプリでの評価，児玉ら [5] による電力性能に着目した評価などがある．本研究は，Fiber ミニアプリスイートにおける同一のアプリケーション，データセットを用いて，京コンピュータから富岳までの国内外の多様なシステム，プロセッサの性能評価を行っており [10]，幅広いプロセッサとの比較が可能である点に特色がある．

5. おわりに

本稿では，Fiber ミニアプリスイートを用いて，スーパーコンピュータ富岳における A64FX プロセッサの性能評価を行った．各アプリケーション・データセットについて，さまざまな MPI プロセス数，OpenMP スレッド数，MPI プロセス割り当て手法，OpenMP スレッド割り当て手法を検討した．mVMC の tiny を除いて，プロセス割り当て手法，スレッド割り当て手法による大きな性能の違いはなかったが，多くのアプリケーション・データセットに対してコンパクトなスレッド割り当てが僅かに高い性能を示した．A64FX は，TX2 や SKL と比較して，CCS-QCD および NTChem で小さいデータセットを実行したときに性能が低く，mVMC では全体に性能が低かった．他のアプリケーションでは，ノードあたりの演算理論ピーク性能に勝る SKL と同等もしくは高い性能を示した．OpenMP によるスレッド並列化がなされていない NICAM および FFB については，コンパイラによる自動並列化を試みたが，いずれもスレッド数の増加に対して 4 スレッド程度までは良好な性能向上を得た．今後の課題としては，A64FX での性能が低かったアプリケーションおよびデータセットについて，プロファイラなどを用いて詳細に原因を分析することが挙げられる．

謝辞

本研究は，富岳共用前評価環境を通じた理化学研究所のスーパーコンピュータ「富岳」の計算資源，および CEA 理研共同研究契約に基づいて CEA が提供するクラスタの計算資源の提供を受け，実施しました．

参考文献

[1] T. Boku, K. Ishikawa, K. Minami, Y. Nakamura, F. Shoji, D. Takahasi, M. Terai, A. Ukawa, and T. Yosie. Multi-block/multi-core SSOR preconditioner for the QCD quark solver for K computer. In *Proceedings of The 30th International Symposium on Lattice Field Theory*, 2012.

[2] Feasibility Study on Future HPC Infrastructures (Application Working Group). Computational Science Roadmap http://hpci-aplfs.aics.riken.jp/, 2014.

[3] Fujitsu Limited. A64FX Microarchtecture Manual. https://github.com/fujitsu/A64FX/blob/master/doc/A64FX_Microarchitecture_Manual_en_1.3.pdf.

[4] A. Jackson, M. Weiland, N. Brown, A. Turner, and M. Parsons. Investigating applications on the a64fx. In *EAHPC-2020 - Embracing Arm: a journey of porting and optimization to the latest Arm-based processors, 2020 IEEE International Conference on Cluster Computing (CLUSTER)*, 2020.

[5] Y. Kodama, T. Odajima, E. Arima, and M. Sato. Evaluation of power controls on supercomputer fugaku. In *Energy Efficient HPC State of the Practice Workshop, 2020 IEEE International Conference on Cluster Computing (CLUSTER)*, 2020.

[6] N. Maruyama. Mini-App Effort in Japan. In *SC13 BoF: Library of Mini-Applications for Exascale Component-Based Performance Modeling*, 2013.

[7] T. Odajima, Y. Kodama, M. Tsuji, M. Matsuda, Y. Maruyama, and M. Sat. Preliminary performance evaluation of the fujitsu a64fx using hpc applications. In *EAHPC-2020 - Embracing Arm: a journey of porting and optimization to the latest Arm-based processors, 2020 IEEE International Conference on Cluster Computing (CLUSTER)*, 2020.

[8] M. Sato. The Supercomputer “Fugaku” and Arm-SVE enabled A64FX processor for energy efficiency and sustained application performance. https://press3.mcs.anl.gov/atpesc/files/2020/07/ATPESC-2020-Track-1-Talk-2-sato.pdf, 2020.

[9] M. Terai, H. Yashiro, K. Sakamoto, S. ichi Iga, H. Tomita, M. Satoh, and K. Minami. Performance optimization and evaluation of a global climate application using a 440m horizontal mesh on the K computer. In *Proceedings of 2014 International Conference for High Performance Computing, Networking, Storage and Analysis (SC 2014)*, 2014.

[10] M. Tsuij, W. T. Kramer, W. Jean-Christophe, N. Jean-Philippe, and M. Sato. A new sustained system performance metric for scientific performance evaluation. 2021.

[11] 小村, 鈴木, 三上, 滝澤, 松田, 丸山. Fiber ミニアプリの性能評価. 情報処理学会研究報告, No. 2014-HPC-145, p. On Line. 情報処理学会, 2014.

[12] 小田嶋, 児玉, 辻, 松田, 丸山, 佐藤. HPC ベンチマーク

プログラムによる A64FX プロセッサ試作機の性能評価. 情報処理学会研究報告, No. 2020-HPC-173, p. On Line. 情報処理学会, 2020.

[13] 辻, W. Jean-Christophe, N. Jean-Philippe, 佐藤. ThunderX2 Arm プロセッサにおける Fiber ミニアプリスイートの性能評価. 情報処理学会研究報告, No. 2019-HPC-171, p. On Line. 情報処理学会, 2019.