

# 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

辻 美和子<sup>1,a)</sup> Weill, Jean-Christophe<sup>3</sup> Nominé, Jean-Philippe<sup>3</sup> 佐藤 三久<sup>1</sup>

**概要:** Arm アーキテクチャは、組み込みやモバイル機器に多く使用されているプロセッサアーキテクチャであるが、近年では、京コンピュータの後継機であるスーパーコンピュータ富岳向けに富士通が開発した A64FX や、Arm ベースのスーパーコンピュータとしてはじめて 1 ペタフロップスを越えた Astra に搭載されている ThunderX2 など、HPC 分野にも進出している。ThunderX2 Arm プロセッサは、現在利用可能な中ではもっとも高性能な Arm ベースのプロセッサである。本稿では、ThunderX2 Arm プロセッサにおけるミニアプリケーションの性能評価を行った。また同時マルチスレッディングやスレッドアフィニティの効果についても検証した。

## Performance Evaluation of ThunderX2 Arm Processor for the Fiber Miniapp Suite

**Abstract:** Recently, the Arm architecture, which has been widely used in embedded and mobile devices, is now also adopted in HPC systems such as the Marvell's ThunderX2 Arm processor in the Astra, the Fujitsu's A64FX in the supercomputer Fugaku. The ThunderX2 Arm processor is the most efficient Arm-based processor available today. This paper shows the result of mini applications evaluation on the ThunderX2 Arm processor. The effects of the simultaneous multithreading and thread affinity are also investigated.

### 1. はじめに

Arm アーキテクチャは、組み込みやモバイル機器に多く使用されるプロセッサアーキテクチャであるが、近年では、京コンピュータの後継機であるスーパーコンピュータ富岳向けに富士通が開発した A64FX や、Arm ベースのスーパーコンピュータとしてはじめて 1 ペタフロップスを越えた Astra に搭載されている ThunderX2 など、HPC 分野にも進出している。

本稿では、現在利用可能な Arm ベースのプロセッサではもっとも高性能である、ThunderX2 Arm プロセッサの効率的な使用に関する知見を得るために、仏・原子力代替エネルギー庁 (CEA) の Init スーパーコンピュータに導入された ThunderX2 パーティションにおいて、ミニアプリ

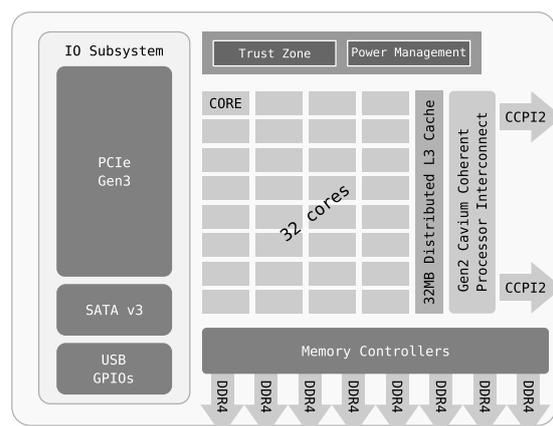


図 1 ThunderX2 Overview

ケーションセット Fiber [3] の各アプリケーションの性能評価を行った。

### 2. ThunderX2

#### 2.1 ハードウェア諸元

本章では、本評価に用いた ThunderX2 ARM プロセッ

<sup>1</sup> 理化学研究所計算科学研究センター  
RIKEN Advanced Institute for Computational Science  
<sup>2</sup> 筑波大学  
University of Tsukuba  
<sup>3</sup> CEA  
<sup>a)</sup> miwako.tsuji@riken.jp

サ [5] の概要について述べる。

図 1 に Marvell ThunderX2 のプロセッサの概要を示す。1 ノードは 2 ソケットからなり、1 ソケットには Armv8.1 コアが 32 コア搭載されている。各コアはアウト・オブ・オーダー (OoO) 実行が可能であり、1,2 もしくは 4 スレッドの同時マルチスレッディング (SMT) をサポートする。ただし、本稿で用いたプロセッサでは、最大 2 スレッドまでの SMT のみをサポートするよう設定されている。周波数は最大で 2.5GHz まで複数のオプションがあるが、本稿で用いたプロセッサは 2.2GHz で動作する。メモリは 8 チャンネルの DDR4-2666 であり、容量は 256GB/チップ、転送速度は 320GB/s である。ソケット間は、Cavium Coherent Processor Interconnect 2 (CCPI2) で接続されており、最大転送速度は 600Gbs (25Gbs x 24) である。ノード間の通信は Infiniband EDR である。

図 2 に、ThunderX2 チップの内部構造のブロックダイアグラムを示す。各コアは 2 つの 128-bit NEON 浮動小数点ユニット、およびそれぞれ 32KB の L1D/L1I キャッシュ、256KB の L2 キャッシュなどを持つ。4 コアで 1 ブロックを構成し、各ブロックは 2MBx2 スライスの L3 キャッシュを持つ。1 ソケットは 8 ブロック 32 コアからなるので、ソケットあたりの L3 キャッシュは合計で 32MB である。ブロックは、双方向のコヒーレント・リング・インターコネクタで接続されている。リング内には、CCPI2 インタコネクタ、DDR4、IO コントローラバスへのリンクのためのポートが存在する。PCIe は Gen3 である。

## 2.2 SMT と Affinity

本節では、ThunderX2 ARM 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。本稿で用いたシステムでは、KMP\_AFFINITY 環境変数を設定することで、ランタイムに OpenMP スレッドをコアや SMT にバインドすることが可能である。

KMP\_AFFINITY 環境変数の構文は以下である：

```
KMP_AFFINITY=[<modifier>,...]<type>\n                [,<permute>] [,<offset>]
```

本稿では必須の引数である type およびオプションである modifier の granularity について、パラメータを変えて性能を評価した。type では、compact および scatter が考慮された。compact は  $n+1$  スレッドを  $n$  スレッドのなるべく近くに割り当て、scatter はシステム全体にわたってスレッドを均等に分配する。granularity では、fine および core が考慮された。core はスレッドを荒いレベルであるコアにバインドし、fine はスレッド (サポートされている場合は各 SMT) にバインドする。

図 3 に、それぞれの KMP\_AFFINITY の指定に関して、各コアに 2SMT をサポートする場合に、4 つの OpenMP スレッドがどのようにバインドされるかを示す。本システ

ムでは、1 プロセッサに 32 コア存在し、SMT のスレッド ID はラウンドロビンで割り当てられるため、コア 0 には 0 および 32 の、コア 1 には 1 および 33 のスレッド ID が割り当てられている。granularity で core を指定した場合、OpenMP スレッドは、対応する物理コアの上のスレッドセットにバインドされる。それに対して、fine を指定した場合には、各 OpenMP スレッドは、1 つのスレッドにバインドされる。type に compact を指定した場合、スレッド 0 と 1、およびスレッド 2 と 3 は同じコアに割り当てられる。このとき、granularity が core であれば、OpenMP スレッド 0 はスレッド ID {0,32} に、OpenMP スレッド 1 はスレッド ID {0,32} に割り当てられ、granularity が fine であれば、OpenMP スレッド 0 はスレッド ID 0 に、OpenMP スレッド 1 はスレッド ID 32 に割り当てられる。type が scatter の場合には、granularity が core であれば、OpenMP スレッド 0 はスレッド ID {0,32} に、OpenMP スレッド 1 はスレッド ID {1,33} に割り当てられ、granularity が fine であれば、OpenMP スレッド 0 はスレッド ID 0 に、OpenMP スレッド 1 はスレッド ID 1 に割り当てられる。

## 3. 性能評価

### 3.1 Fiber miniapp suite

Fiber miniapp suite[3] は、Feasibility Study on Future HPC Infrastructures [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。

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、本実験では対象にしなかった。

### 3.2 システム諸元

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ェ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、2 章で述べた 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、一部のアプリケーションでは、比較のために 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。それぞれの諸元を表 2 にまとめた。

### 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、2 つの異なるサイズの

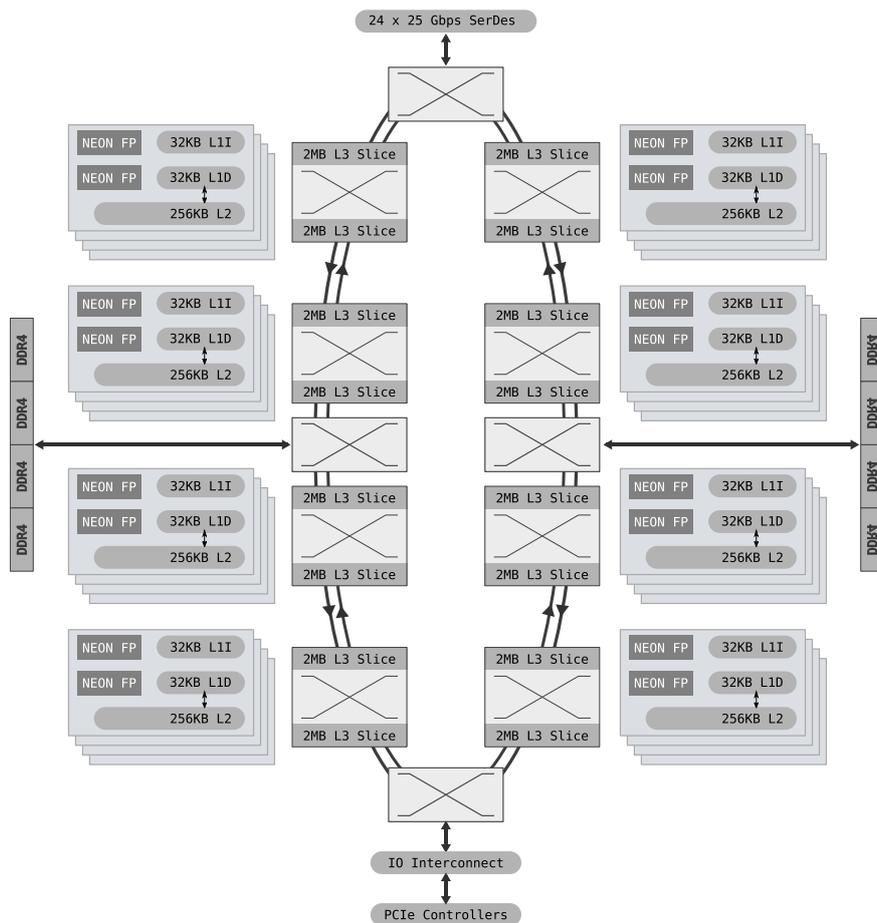


図 2 The block diagram of the internal architecture of the ThunderX2 chip

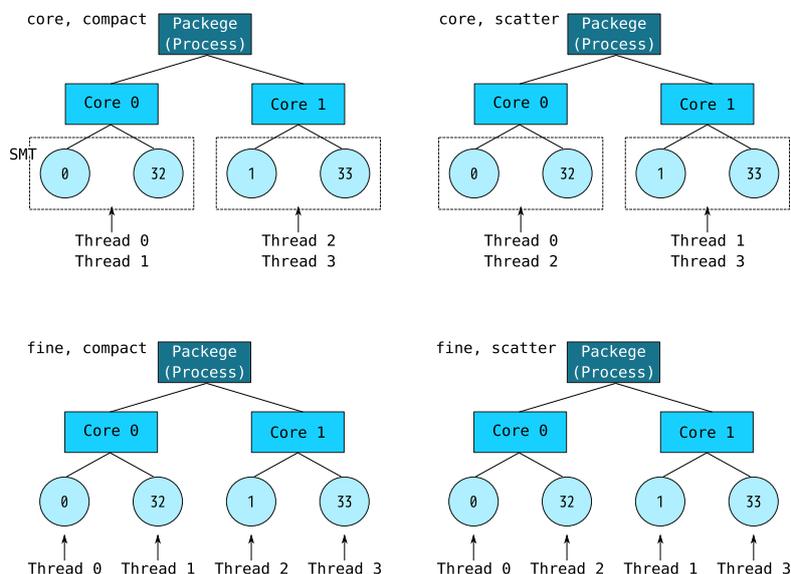


図 3 OpenMP Thread Affinity, 2SMT, # of OpenMP threads = 4

データを対象とした。格子サイズのうち、最初の三次元の格子は 8x8x8 づつ分散並列処理される。

図 4-7 に、TX2 および SKL クラスタにおける実験の結果を示す。X 軸がスレッド数を、Y 軸が実効性能 (GFlops) を示す。MPI のプロセス数はデータサイズ毎に固定である。図 3 で議論したアフィニティの設定について、スレッ

ド数を変化させた場合の GFlops を調査した。すべての場合で、granularity の設定は結果に大きく影響を与えなかった。scatter と compact では、OpenMP スレッド数が少ない場合には、scatter のほうが高い性能を示した。これは、OpenMP スレッド数が少ない場合、scatter では 1 物理コアにまずは 1 スレッドづつを配置するが、compact では 1

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

表 2 Arm & Intel SKL partitions in the Inti supercomputerz

	Arm Cluster	Intel Cluster
CPU	Marvell ThunderX2	Intel Xeon Platinum 8168 (Skylake)
# of cores / socket	32	24
# of sockets	2	2
SMT	2	2
CPU GHz	2.2 GHz	2.7 GHz
Peak performance / node	1126.4 GFlops	4147.2 GFlops
Memory	DDR4	DDR4
Capacity / node	256 GB	192 GB
Bandwidth / node	320 GB/s	-
# of nodes	28	22
Compiler	arm-compiler 19.0.0	intel/17.0.6.256
MPI library	openmpi 2.0.4	mpi/openmpi/2.0.4

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 (Arm TX2)	1 (64 cores)	4 (256 cores)
ノード数 (Intel SKL)	1 (48 cores)	6 (288 cores)*1
MPI プロセス数	1	64
OpenMP スレッド数 (Arm TX2)	1 ~ 64	1 ~ 8
OpenMP スレッド数 (Intel SKL)	1 ~ 96	1 ~ 8

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 (Arm TX2)	1 (64 cores)	4 (256 cores)
ノード数 (Intel SKL)	1 (48 cores)	4 (192 cores)
MPI プロセス数 (Arm TX2)	8	64
MPI プロセス数 (Intel SKL)	8	64
OpenMP スレッド数 (Arm TX2)	2 ~ 16	1 ~ 12
OpenMP スレッド数 (Intel SKL)	2 ~ 8	1 ~ 6

物理コアに最大 2 スレッドを割り当てるため物理コアを効率的に使用できないためである。

class1 では, Arm TX2 は 32 スレッドを fine,scatter で, Intel SKL では 24 スレッドを fine,scatter で用いた時が, もっとも性能が高かった. このとき, それぞれ 1 ソケットの各コアに 1 スレッドが割り当てられている。

class2 では, TX2 は 8 スレッドを fine,scatter で, SKL では 4 スレッドを core,scatter で用いた時が, もっとも性能が高かった。

性能比は, class1 では TX2:SKL=1:1.85 で SKL のほうが高く, class2 では 1:1.28 で, SKL のほうが使用ノード数が多いことを考えればほぼ同等であった. これは, class1 は, ノードあたりのデータ量が小さいことから, キャッシュミスが少く, 最大演算性能が実行効率に影響を与えるいっぽう, class2 はノードあたりの格子点数が増加してい

るため, メモリプレッシャが大きくなり, 演算性能に加えてメモリバンド幅が実行効率に影響を与えるためであると考えられる。

本実験で, granularity の設定は実効性能に大きな影響を与えないことがわかったため, 以降の実験では granularity について考慮しない。

### 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。

図 8-9 に実験の結果を示す. X 軸がスレッド数を, Y 軸が実行性能 (GFlops) である. TX2 は, FFVC-256 では, プロセスあたり 8 スレッドを scatter で割り当てたときが,

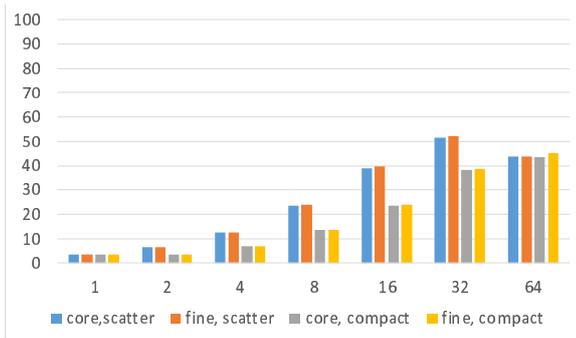


図 4 CCS-QCD class1, TX2  
Y-Axis is GFlops, X-Axis is the number of threads

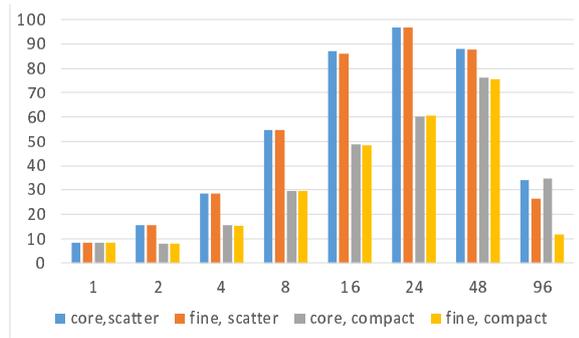


図 5 CCS-QCD class1, SKL  
Y-Axis is GFlops, X-Axis is the number of threads

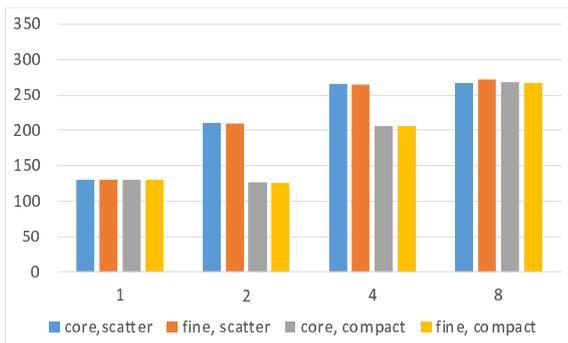


図 6 CCS-QCD class2, TX2  
Y-Axis is GFlops, X-Axis is the number of threads

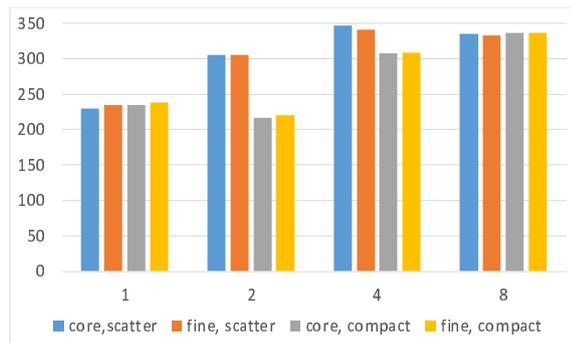


図 7 CCS-QCD class2, SKL  
Y-Axis is GFlops, X-Axis is the number of threads

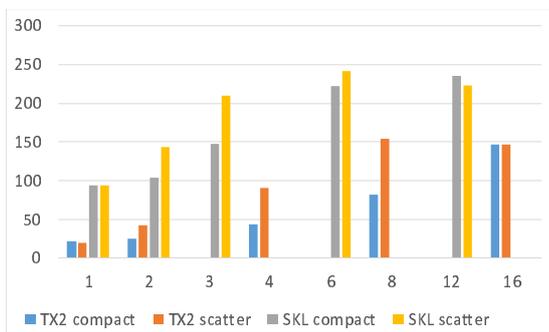


図 8 FFVC-256, Arm TX2 & Intel SKL  
Y-Axis is GFlops, X-Axis is the number of threads

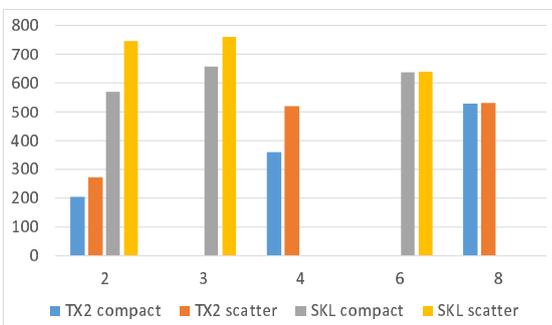


図 9 FFVC-1024, Arm TX2 & Intel SKL  
Y-Axis is GFlops, X-Axis is the number of threads

もっとも高い性能を示した。このとき、ノード内のすべての物理コアで 1 スレッドが実行されている。プロセスあた

り 16 スレッドを scatter もしくは compact で割り当てた時もほぼ同様の結果を示した。FFVC-1024 では、プロセスあたり 8 スレッドを scatter で割り当てたときが、もっとも高い性能を示した。このとき、ノード内のすべての物理コアで 2 スレッドが実行されている (FFVC-256 とはノードあたりのプロセス数が違うことに注意)。同じく 8 プロセスを compact で割り当てたとき、および 4 スレッドを scatter で割り当てたときもほぼ同等の結果を示した。

SKL は、FFVC-256 では、プロセスあたり 6 スレッドを scatter で割り当てた時が、もっとも高い性能を示した。FFVC-1024 では、プロセスあたり 3 スレッドを scatter で割り当てたときがもっとも高い性能を示した。いずれも、OpenMP の各 1 スレッドが、1 つの物理コアを使用している場合である。

FFVC-256 では、TX2 と SKL の最大実行効率の比率は、TX2:SKL=1.0:1.57 であった。FFVC-1024 は、TX2:SKL=1.0:1.43 であった。

### 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, 11 に、実験の結果を示す。X 軸がスレッド数、

表 5 mVMC のデータセットと実験環境

	tiny	middle
NSplitSize	1	4
ノード数 (Arm TX2)	1 (64 cores)	4 (256 cores)
ノード数 (Intel SKL)	1 (64 cores)	4 (192 cores)
MPI プロセス数	1	64
OpenMP スレッド数 (Arm TX2)	8 ~ 128	2 ~ 8
OpenMP スレッド数 (Intel SKL)	8 ~ 96	2 ~ 6

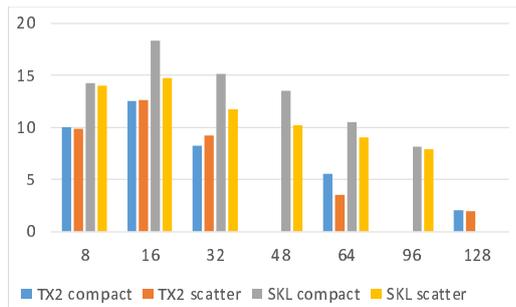


図 10 mVMC tiny, Arm TX2 & Intel SKL  
 Y-Axis is GFlops, X-Axis is the number of threads

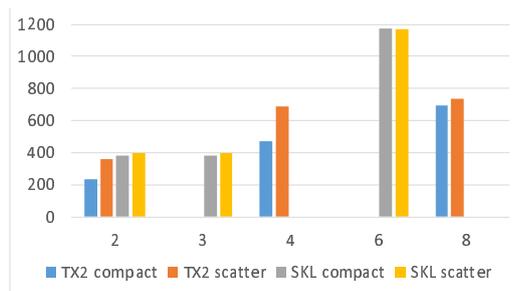


図 11 mVMC middle, Arm TX2 & Intel SKL  
 Y-Axis is GFlops, X-Axis is the number of threads

Y 軸が実効性能 (GFlops) である。TX2 は、データセット tiny では、16 スレッドを scatter で割り当てたときに最大の実行効率を示した。以降はスレッド数をおおきくするごとに性能が低下した。これは、データが小さすぎて、OpenMP 並列化にともなうオーバーヘッドのほうが大きくなっていると考えられる。データセット middle では、8 スレッドを scatter で割り当てたときに最大の実行効率を示した。これはすべてのコアが 2SMT を実行している状態である。8 スレッドを compact で割り当てた場合、4 スレッドを scatter で割り当てた場合も、良好な性能を示した。

SKL は、データセット tiny では、TX2 と同様に最大のコア数より少ない 16 スレッドを compact で割り当てたときにもっとも高い性能を示した。データセット middle では、6 スレッドを compact で割り当てた時に最大の実行効率を示し、同じく 6 スレッドを scatter で用いた時もほぼ同等の性能を示した。TX2 と同様に、このときすべてのコアが 2SMT を実行している。

最大実行効率の比率は、tiny で TX2:SKL=1.0:1.45 であ

表 6 NTChem のデータセットと実験環境

	h2o	taxol
ノード数 (Arm TX2)	1 (64 cores)	4 (256 cores)
ノード数 (Intel SKL)	1 (48 cores)	4 (192 cores)
MPI プロセス数	2 ~ 16	16 ~ 64
MPI プロセス数	2 ~ 12	4 ~ 48
OpenMP スレッド数 (Arm TX2)	8	8
OpenMP スレッド数 (Intel SKL)	8	8
BLAS library (Arm TX2)	armpl-19.0.0	armpl-19.0.0
BLAS library (Intel SKL)	mkl/17.0.6.256	mkl/17.0.6.256

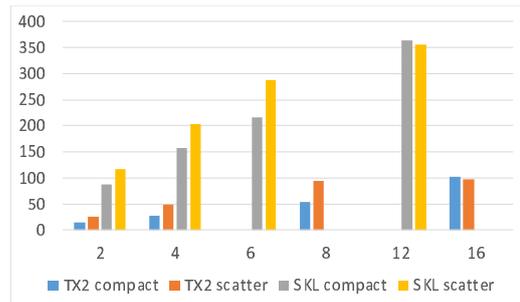


図 12 NTChem h2o, Arm TX2 & Intel SKL  
 Y-Axis is GFlops, X-Axis is the number of processes

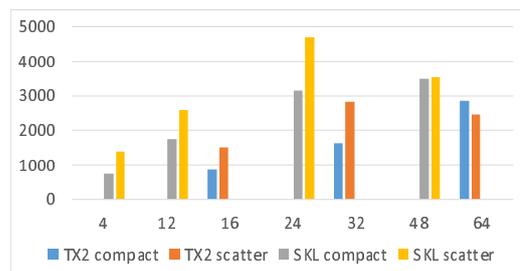


図 13 NTChem taxol, Arm TX2 & Intel SKL  
 Y-Axis is GFlops, X-Axis is the number of processes

り、middle では TX2:SKL=1.0:1.59 であった。

### 3.6 NTChem

NTChem 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。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 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。これまでのアプリケーションとは異なり、スレッド数を固定し、MPI 列数が変化していることに注意されたい。

図 12, 13 に実験の結果を示す。X 軸がプロセス数、Y 軸が実効性能 (GFlops) である。TX2 は、データ h2o では、16 プロセス、compact の時が最大の性能を示し、16 プロセスおよび 8 プロセスの compact もほぼ同等の性能を示した。データ taxol は、64 プロセス、compact の時が最大の性能を示し、32 プロセスの scatter もほぼ同等の性能を

表 7 NICAM-DC

	gl05r100z40
ノード数	1
MPI プロセス数	10
ソケット数	1 ~ 2

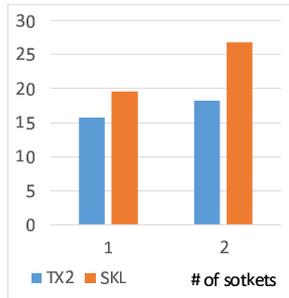


図 14 NICAM gl05r100z80, Arm TX2 & SKL  
Y-Axis is GFlops, X-Axis is the number of sockets

示した。また、最大の実行効率は 2860GFlops であり、これは理論ピーク性能の 63 パーセントである。

SKL は、データ h2o では、12 プロセス、compact のとき最大の性能を示した。データ taxol では 24 プロセス、scatter のときに最大の性能を示した。このときの実行効率は、4695.49GFlops であり、これは理論ピーク性能の 28 パーセントである。

TX2 と SKL との性能比は、h2o で TX2:SKL=1.00:3.54, taxol で TX2:SKL=1.00:1.64 であった。

### 3.7 NICAM-DC

NICAM-DC-MINI[4] 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。

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。しかしながら、TX2, SKL とともに、良好なスレッド並列コードを生成することができなかったため、ここでは MPI 並列のみを考慮し、1 ノードの 1 ソケット、および 2 ソケットを用いて性能評価を行った。1 ソケットのとき、各 MPI プロセスはなるべく近隣の物理コアに割り当てられ、2 ソケットのときは MPI プロセスどうしはなるべく離して配置した。

図 14 に実験の結果を示す。NICAM は、要求 BF の高いアプリケーションであることから、TX2 および SKL とともに、2 ソケットを用いた場合の方が高い性能を示した。それぞれの性能比は 1 ソケットの場合は TX2:SKL=1.00:1.25, 2 ソケットの場合は 1.00:1.47 であった。

### 3.8 FFB

FrontFlow/blue (FFB)-MINI は、有限要素法による熱流

表 8 FFB

	sample
ノード数	1
MPI プロセス数	8
ソケット数	1 ~ 2

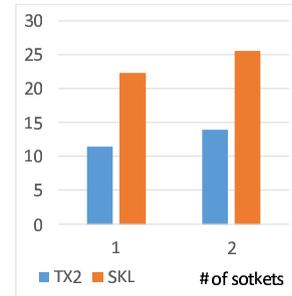


図 15 FFB, Arm TX2 & SKL  
Y-Axis is GFlops, X-Axis is the number of sockets

表 9 各ミニアプリで最大の実行効率を与えたスレッド数と Affinity

Application	Data	Procs x threads, SMT/core	Affinity
CCS-QCD	class1	1p x 32t, -	scatter
	class2	64p x 8t, 2t	scatter
FFVC	256	8p x 8t, 1t	scatter
	1024	64p x 8t, 2t	scatter
mVMC	tiny	1p x 32t, -	scatter
	middle	64p x 8t, 2t	scatter
NTchem	h2o	16p x 8t, 2t	compact
	taxol	64p x 8t, 2t	compact

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、MPI 並列のみを考慮し、1 ノードの 1 ~ 2 ソケットを用いて性能評価を行った。

FFB のデータセットと設定を表 8, 結果を図 15 に示す。FFB も NICAM と同様に要求 BF の高いアプリケーションであり、2 ソケットを用いたほうが高い性能を示した。性能比は 1 ソケットの場合は TX2:SKL=1.00:1.94, 2 ソケットの場合は 1.00:1.84 であった。

### 3.9 まとめ

表 9 に、CCS-QCD, FFVC, mVMC および NTChem で、最大の実行効率であったときの、スレッド数とアフィニティを示す。表から、多くの場合で、1 コアに 2SMT を割り当てることで最大の効率を達成している。アフィニティとしては、NTChem 以外は scatter がより高い性能を示した。これは、NTChem の多くを占める数値演算ライブラリが、compact 向けによりチューニングされているからではないかと考えられる。ただし、3.3-3.6 で述べたように、1 コア 2SMT で compact, 1 コア 1SMT で scatter, 1 コア 2SMT で compact の間には大きな性能の差はなかった。

表 10 に各ミニアプリケーションに対する TX2 と SKL

表 10 Comparison of Arm TX2 and Intel SKL

App	QCD		FFVC		mVMC		NTChem		NICAM	FFB
	1	2	256	1024	tiny	middle	h2o	taxol	gl05r100z40	sample
TX2 GFlops	52.2	271	154	531	12.6	736	103	2859	18.3	13.9
TX2 % of Peak	4.64	3.01	13.7	11.8	1.12	16.3	9.14	63.5	1.62	1.23
SKL GFlops	96.8	347	241	760	18.3	1172	363	4695	26.9	25.6
SKL % of Peak	2.22	1.39	5.83	4.58	0.44	7.07	8.78	28.3	0.65	0.617
SKL/TX2	1.85	1.28 *2	1.57	1.43	1.45	1.59	3.54	1.64	1.47	1.84

の性能をまとめた。SKLは実行性能でTX2を上回る一方、TX2は理論ピーク性能でより高い数値を示した。

#### 4. おわりに

本稿では、Fiber ミニアプリケーションセットを用いて、Arm ThunderX2 プロセッサの性能評価を行い、同時マルチスレッディングやスレッドアフィニティの効果についても検証した。OpenMPによる並列化がされていないアプリケーションや、データサイズが小さすぎて多数コアでの実行に向かない場合を除き、ほとんどのアプリケーション・データで、SMT2 スレッド・アフィニティ scatter で使用したときがもっとも高い効率を示した。しかし、SMT2 スレッド・アフィニティ compact および、SMT1 スレッド・アフィニティ scatter もほぼ同等の効率を示した。すなわち、Arm TX2 の効率的な使用のためには、すべての物理コアに1つ以上のOpenMP スレッドを割り当てること肝要であり、SMTによりスレッド数を増やしても性能改善は僅かであった。

また、Intel Skylake プロセッサとの性能比較を行った。比較に用いたSKLとTX2との理論ピーク性能比は、TX2:SKL=1.00:3.68であるが、アプリケーション実行効率の比率は、1.00:1.43 ~ 3.56 (ノード数を揃えた比較をしていない CCS-QCD-class2 を除く)であった。SKLはすべてのアプリケーション・データでTX2よりも高い性能を示す一方、ピーク性能比ではTX2が高いという結果になった。

#### 参考文献

- [1] T. Boku, K. Ishikawa, K. Minami, Y. Nakamura, F. Shoji, D. Takahashi, 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] N. Maruyama. Mini-App Effort in Japan. In *SC13 BoF: Library of Mini-Applications for Exascale Component-Based Performance Modeling*, 2013.
- [4] 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 *Proceed-*

*ings of 2014 International Conference for High Performance Computing, Networking, Storage and Analysis (SC 2014)*, 2014.

- [5] ThunderX2 Arm-based Processors. <https://www.marvell.com/server-processors/thunderx2-arm-processors/>.