

Suiren (睡蓮) による計算科学アプリケーションの性能評価

中里 直人¹

概要：Suiren(睡蓮)は、メニーコアプロセッサ PEZY-SC をアクセラレータとして採用するスーパーコンピュータシステムである。PEZY-SC プロセッサは、Multiple Instruction Multiple Data(MIMD)方式のプロセッサであり、Processing Element(PE)を1024個搭載している。PEZY-SC プロセッサは、これまでGPUなどのアクセラレータで実行してきたアプリケーションを、OpenCL 準拠のプログラミングモデルによって実行可能である。Suiren は、ホスト計算機とアクセラレータからノード全体を、液浸で冷却する手法により、LINPACK ベンチマークでの電力効率が非常に高い。我々は、Suiren で様々な計算科学アプリケーションの性能評価をおこなった。具体的には、高精度天文多体シミュレーション、ツリー法/SPH 法による粒子シミュレーション、津波進化シミュレーション、多倍長精度計算浮動小数点演算エミュレーションである。特に、高精度天文多体シミュレーションでは、Suiren の4ノード(PEZY-SC プロセッサ 32チップ)にて9.18 TFLOPS の性能を得た。

キーワード：OpenCL, アクセラレータ

1. はじめに

PEZY-SC プロセッサは、PEZY Computing 社により開発された Multiple Instruction Multiple Data(MIMD)方式のプロセッサであり、ホスト計算機と組み合わせて演算アクセラレータとして利用する。PEZY-SC プロセッサを採用したスーパーコンピュータシステムとして、2014年秋に高エネルギー加速器研究機構の Suiren(睡蓮)が稼働を始め、2015年春からは、高エネルギー加速器研究機構で Suiren Blue(青睡蓮)、理化学研究所で Shoubu(菖蒲)が稼働している。いずれのシステムも、ホスト計算機とアクセラレータからなるノード全体を液体に浸して放熱する冷却機構を採用しているため、エネルギー効率が高いシステムである。2015年8月のGreen500(LINPACK ベンチマークによる消費電力あたりの性能ランキング)では、Shoubu, Suiren Blue, Suiren が最上位を占めた。そのため PEZY-SC での LINPACK ベンチマークの実装や、それに伴う倍精度演算行列積の実装についてはよく最適化されている。一方で、最初の稼働システムである Suiren が利用可能になってから1年足らずのため、他の計算科学アプリケーションの移植は現在進行中か、今後進められていく予定である。

本研究報告では、我々がこれまでアクセラレータシステム用に実装してきたいくつかの計算科学アプリケーションを PEZY-SC プロセッサに移植した経緯と、それらの性能

評価について報告する。

2. PEZY-SC プロセッサ

本章では PEZY-SC(Super Computing) プロセッサおよび Suiren のアーキテクチャ概略とそのプログラミング手法について説明する。

2.1 システムアーキテクチャ

PEZY-SC プロセッサは、多数の演算コア(Processing Element; PE)を集積した MIMD プロセッサである。個々の PE が自身のプログラムカウンタを保持しているところが、GPU のような Simultaneous Instruction Multiple Data(SIMD) プロセッサとは異なる。PE は単精度/倍精度浮動小数点演算器に加えて、整数 ALU、レジスタとローカルメモリからなり、PE が4個で Village と呼ばれるブロックを構成する。Village ブロックには、2PE ごとの1次キャッシュがある。Village ブロックが4個で City ブロックを構成し、City ブロックには共用の2次キャッシュと関数演算用の特殊演算ユニット(SFU)がある。さらに、City ブロックが16個で Prefecture ブロックを構成し、Prefecture ブロックごとに3次キャッシュがある。PEZY-SC プロセッサは、4個の Prefecture ブロックからなり、全部で1024個の PE を搭載する。ひとつの PE あたりの浮動小数点演算のスループットは、単精度の場合4演算(二つの積和算)であり、単精度の場合2演算(ひとつの積和算)である。733 MHz で動

¹ 会津大学大学院コンピュータ理工学研究科

作する場合、プロセッサ単体のピーク性能は単精度/倍精度演算で、それぞれ 3/1.5 TFLOPS である。PEZY-SC プロセッサには、その他に、メモリコントローラと PCI Express インターフェイス、制御用の CPU コア 2 個が搭載されている。

Suiren(機種名 ExaScaler-1)の計算ノードは、Intel Xeon-2600v2 を 2 個と PEZY-SC プロセッサが 4 個搭載された PEZY-SC Quad Board からなる。PEZY-SC Quad Board は、ホスト CPU と PCI Express x16 で接続され、1 プロセッサあたり 32GB の DDR3 メモリを搭載する。Suiren のノードは、PEZY-SC Quad Board を 2 台搭載し、Suiren 全体では 32 ノードからなる。よって、PEZY-SC プロセッサの総数は 256 チップである。これらのノード間は Infini-Band FDR で接続されている。ノードは、8 ノードずつ液浸冷却用筐体に収められており、液体を循環させることで PEZY-SC Quad Board だけでなくホスト CPU や他の基盤を直接冷却する。2015 年 7 月に発表された TOP500 の結果では、Suiren は Rmax 性能 206.6 TFLOPS(Nmax 983,040) で 366 位であった。TOP500 のために、アクセラレータ用に最適化された並列 LU 分解コード [12] をベースとした PEZY-SC プロセッサ用最適化と、PEZY-SC プロセッサ用の DGEMM ライブラリが開発された。

2.2 プログラミング手法

PEZY-SC プロセッサをアクセラレータとして利用するためには PZCL を利用する。PZCL は OpenCL のサブセットからなるプログラミング環境であり、OpenCL と同様に演算カーネルを記述し、ホストプログラムから API を介して演算カーネルを起動する。おおむね OpenCL 1.0 の規格に準拠しているが、具体的には以下の違いがある。

- OpenCL ではカーネルの記述に、記憶領域指定用のキーワードなどが拡張された C 言語を利用するが、PZCL ではそのような拡張のない通常の C 言語で記述する。
- PZCL ではカーネル記述の際に、階層的な PE とキャッシュの構造に対応した API により、明示的にキャッシュ読み書きを制御できる。
- PZCL では OpenCL のカーネル用ビルトイン関数に未対応のものがある。
- OpenCL ではカーネルをオンラインでもオフラインでもコンパイル可能であるが、PZCL ではオフラインコンパイルのみサポートされており、アプリケーション実行時にカーネルバイナリをロードする必要がある。

これらの差異のために、元々 OpenCL C API で記述されたアプリケーションは、カーネル記述とホストプログラムでの API 呼び出し部分を修正する必要がある。

我々は、条件コンパイルとマクロとを組み合わせることで、OpenCL と PZCL 用のソースコードを共通化した。本報告での性能評価は、全て OpenCL が動作する CPU や

GPU をターゲットとして開発されたものであり、カーネル記述に PEZY-SC プロセッサに特化した最適化はしていない。

3. アプリケーション性能評価

本章では Suiren に移植した計算科学アプリケーションの紹介と、Suiren でのそれらの性能評価について報告する。いずれの場合も、これまで我々が OpenCL をサポートしたアクセラレータをターゲットとして計算高速化をおこなったアプリケーションであり、これまで各社の GPU およびマルチコア CPU システムや Intel Xeon Phi で動作している。

3.1 高精度天文多体シミュレーション

星団や銀河中心などの天体の進化をシミュレーションするためには、粒子間に働く重力相互作用を精密かつ高速に計算する必要があるだけでなく、粒子の軌道を数値積分する際に適した手法 (Hermite 積分法) を利用する必要がある [6], [10]。シミュレーションでの粒子数を N とした時、Hermite 積分法をアクセラレータによって高速化する場合、粒子の軌道積分の演算量は $O(N)$ のためホスト計算機で計算し、粒子間の相互作用の演算量は $O(N^2)$ のためアクセラレータで計算する [11], [15]。なお、Hermite 積分法では、粒子間の重力相互作用 (\vec{A} , 位置ベクトルの時間による二階微分) だけでなく、位置ベクトルの時間による三階微分 (\vec{J}) を粒子の位置ベクトルと速度ベクトルから直接計算する必要がある。よって、GPU による Hermite 法の高速化 [3] と同様に、我々は \vec{A} と \vec{J} を計算する OpenCL カーネルを実装し、それをライブラリ (OCLG6A) として利用可能とした。OCLG6A では、様々なカーネルのバリエーションが利用可能であるが、PEZY-SC プロセッサでの性能評価には、Array of Structure のデータ構造を採用し、倍精度演算と単精度演算を組み合わせたカーネルを採用した。OCLG6A を MPI により並列化された ϕ -GRAPE [5] と組み合わせることで、Suiren の複数ノードを使って性能評価をおこなった。なお、 ϕ -GRAPE での軌道積分は、積分タイムステップごとに、条件に基づいて選択された粒子を順次積分していくブロック化積分法を採用しており、演算量がマイステップ常時 $O(N^2)$ となるわけではない。

図 1 は ϕ -GRAPE による、粒子分布が Plummer 球の場合の軌道積分した場合の性能をしめす。横軸は Plummer 球の粒子数 N を、縦軸は相互作用当たり 60 演算で換算した性能である。演算に利用した単位は、ポテンシャルの重力加速度定数を 1 で規格化した単位系を採用し、重力ポテンシャルのソフトニングパラメータ $\epsilon = 10^{-4}$ とした。いずれの場合の計算結果も、時間 $t = 0$ から $t = 0.5$ まで積分して全エネルギーがこのパラメータの範囲内でよく保存していることを確認した。

どの場合もホスト計算機の MPI プロセスあたりひとつ

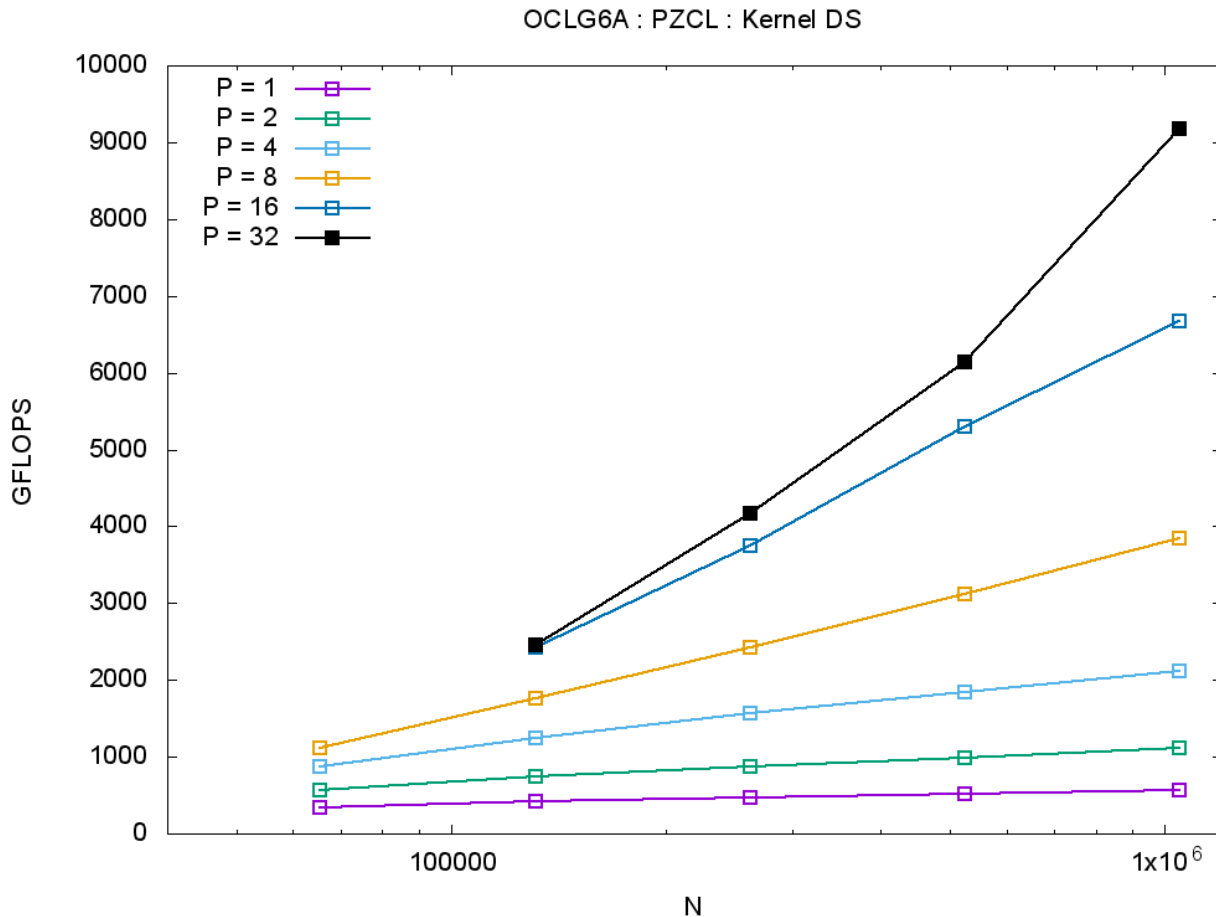


図 1 OCL6GA と ϕ -GRAPE の Suiren での性能評価.

の PEZY-SC プロセッサを利用する場合であり、 N とプロセス数 P を変えて性能評価をおこなった。問題サイズが十分大きい場合は ($N \geq 524288$), P を大きくすると性能はほぼ比例して向上する。 $P = 32, N = 1048576$ の場合、演算性能は 9.18 TFLOPS であり、1 プロセスあたり ~ 290 GFLOPS に相当する。同じ演算カーネルを AMD Radeon R9 280X(単精度演算性能 3.48 TFLOPS) にて実行した場合、 $P = 1, N = 32768$ で最大で約 900 GFLOPS であった。よって、PEZY-SC プロセッサに特化した最適化による性能向上の余地がある。一方で、マルチプロセスによる MPI アプリケーションとしては、現状のカーネルを利用した場合、Suiren 全体を利用 ($P = 256$) することで、 > 60 TFLOPS の性能になると予測される。

3.2 SPH 法による宇宙流体シミュレーション

Smoothed Particle Hydrodynamics(SPH) 法は宇宙における天体シミュレーションのために考案された流体シミュレーション手法である [4], [9]。SPH 法は Hermite 積分法と同じく、粒子間相互作用と軌道積分により系の進化を計算する。一方、粒子間相互作用は近接相互作用となるため、八分木データ構造を使うことで効率よく計算できる。

我々はこれまでに、SPH 法をアクセラレータ計算機で高速化するために、八分木データ構造を辿る (ツリーウォーク) 処理を OpenCL カーネルとして実現し、重力相互作用の場合の性能評価をおこなった [13]。さらに、この手法を SPH 法に拡張し (計算コード OTOO), GPU による宇宙における白色矮星の衝突・合体のモデル計算をはじめ可能とした [14], [16]。ここでは、重力相互作用と SPH 法の PEZY-SC プロセッサでの性能評価について報告する。以下の演算は全て単精度演算で実装している。

図 2 は、粒子を階層的な八分木に格納し、八分木を辿ることで粒子間の距離を容易に判定し、それにより十分また遠方の粒子はまとめて多重極展開にて置き換えることで演算量を削減する方法 (以下、ツリー法) で、粒子間の重力ポテンシャルを計算し軌道積分した場合の性能である。粒子分布としては、中心に向けて密度構造のある Plummer 球の場合と、一様密度の Uniform の場合を示す。それぞれの場合の粒子分布で、 $N = 0.5, 1, 2, 4, 8$ メガ粒子の場合について、1 ステップ当たりの計算時間を示す。TREE, KERNEL, PCIe はそれぞれ、1 ステップあたり、粒子の八分木構造を計算する時間、PEZY-SC プロセッサで八分木を辿りながら重力ポテンシャルを計算する時間 (この評価では $P = 1$),

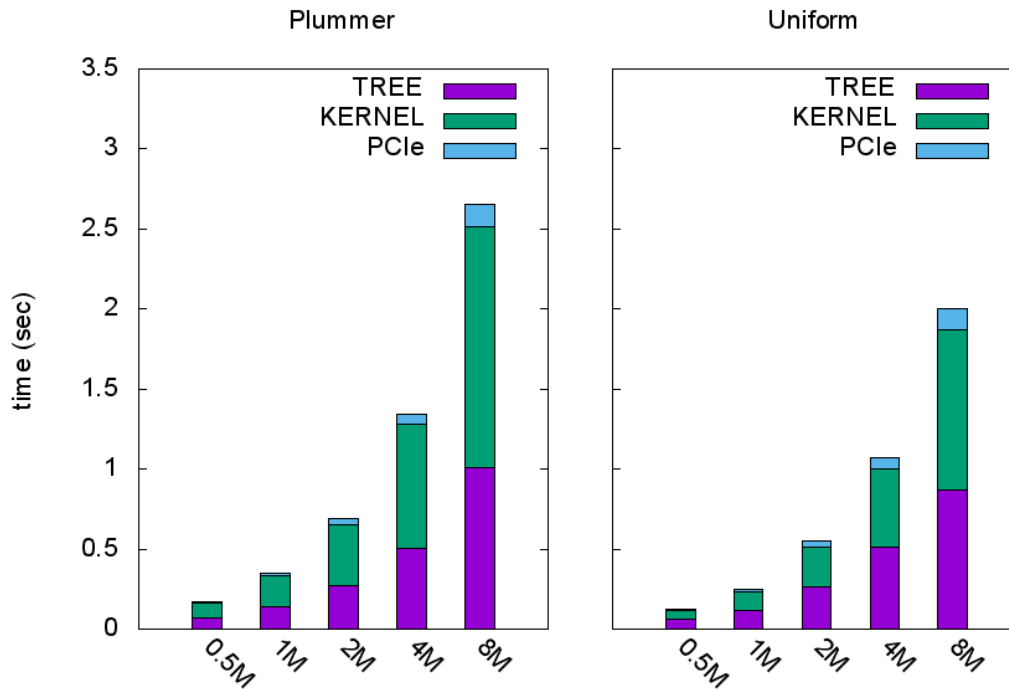


図 2 ツリー法による重力相互作用計算の性能評価. 左右それぞれの結果は Plummer 球と一様球の場合に対応する.

そして、ホスト計算機と PEZY-SC プロセッサとのデータ I/O にかかる時間をしめす。計算時間はいずれの場合にもほぼ N に比例する。以上により、ツリー法による粒子シミュレーションは PEZY-SC プロセッサで有効に計算できることがわかった。

表 3.2 は、OTOO により単独の白色矮星をシミュレーションした場合の演算時間を示す。利用する PEZY-SC プロセッサの数 P を変化させて、 $N = 0.5, 1, 2, 4$ メガ粒子の場合について、1 ステップあたりの計算時間を示している。この計算では 1 ステップあたりの計算には、粒子の八分木構造を計算する時間に加えて、SPH 法による流体力学計算および重力ポテンシャル計算を含む。そのため、図 2 の重力ポテンシャル計算のみの場合と比べて、OTOO では粒子当たりの演算量が多いため、同じ N では計算時間がよりかかる。Suiren ではノードあたり $P = 8$ まで利用可能である。

OTOO では複数の PEZY-SC プロセッサを使う際には、八分木構造をアクセラレータ数で均等に分割した八分木を構築し、個々の PEZY-SC プロセッサにて計算する。理想的には P に反比例して計算時間が短縮するはずであるが、カーネル実行時間以外にも、ホストプログラムで必要な演算処理 (ツリー構築および状態方程式の計算) があるため、一見、1 ステップあたりの計算時間は理想的にはスケールしていない。例えば $N = 4$ メガ粒子の場合、ホストでの演算処理には約 1.4 秒であった。この処理時間は P には依存しないので、 $P = 1, 2, 4, 8$ の場合に、PEZY-SC プロセッサ

で SPH および重力ポテンシャル計算をおこなうための時間 (データ I/O とカーネル実行時間の合計) は、それぞれ $\sim 2, 0.8, 0.4, 0.6$ 秒となる。現状の構成では、 $P = 2, 4$ までは、複数の PEZY-SC プロセッサを利用することが効果的なことがわかった。

3.3 津波進化シミュレーション

海底での地震が起きた後、それによって生じる津波の進化を精密にかつ高速に予測することは、沿岸距離の長い日本では災害予防のために必要である。我々は、そのようなシミュレーションのひとつである MOST(Method of Tsunami Splitting) [17], [18] を OpenMP, OpenCL, OpenACC, CUDA によって並列化し GPU などのアクセラレータやで高速化した [8], [19]。ここでは OpenCL による MOST の実装を、PEZY-SC プロセッサで動作させた場合の性能評価について報告する。以下の演算は全て単精度演算で実装している。

MOST は、浅水方程式を解くための手法であり、計算スキームとしては 2 次元のステンシル計算となる。波の高さと二つの方向の速度を更新するために、現在の格子点を含めてその上下左右の 5 点のデータが必要である。 [8], [19] では、2 次元配列データのアクセス方法を工夫することで OpenMP による性能評価を CPU 及び Intel Xeon Phi でおこなった。MOST の OpenCL 実装では、元のアルゴリズムを変更し、各格子点毎に独立して並列に更新するように

P	0.5M	1M	2M	4M
1	3.922895e-01	8.193518e-01	1.662591e+00	3.382097e+00
2	2.756883e-01	5.440593e-01	1.086922e+00	2.210079e+00
4	2.216704e-01	4.461828e-01	9.096476e-01	1.843587e+00
8	2.214832e-01	4.851834e-01	9.945402e-01	2.058704e+00

表 1 PEZY-SC プロセッサでの SPH 法による白色矮星シミュレーションの性能評価。最初の列は利用した PEZY-SC プロセッサの数を示す。計算時間の単位は秒。

N_x	1 ステップ	格子点あたり
500	6.544603e-03	2.617841e-08
1000	2.504481e-02	2.504481e-08
2000	9.910859e-02	2.477715e-08
3000	2.213808e-01	2.459787e-08
4000	3.926668e-01	2.454167e-08
5000	6.118429e-01	2.447372e-08
6000	8.716483e-01	2.421245e-08
7000	1.197728e+00	2.444343e-08
8000	1.520423e+00	2.375661e-08
10000	2.428314e+00	2.428314e-08

表 2 PEZY-SC プロセッサでの MOST の性能評価。 N_x は計算領域の一方の格子点数。計算時間の単位は秒。

OpenCL カーネルを記述した。この方法では、各スレッドは格子点の集合であるブロック (サイズ $a \times b$ とする) を担当し、各スレッドがアクセスする格子点データは、各方向では 3 点ステンシルのため $(a+2) \times (b+2)$ となる。ブロックサイズ a, b を変更することで、カーネル当たりの演算密度を変化することが可能であり、PEZY-SC プロセッサのようにデータアクセスにキャッシュの利用が重要な場合はより高性能が期待できる。

表 3.3 は、MOST スキームを、問題サイズを変えながら計算した場合の計算時間を示す。今回の性能評価では $a = b = 1$ の場合のみを示す。 N_x は計算領域の一方の格子点数であり、全格子点数は N_x^2 である。この評価では、 $P = 1$ として、MOST による計算を 300 ステップ計算してから、それから 1 ステップあたりの計算時間を求めた。なお、300 ステップの計算の間はホスト計算機とのデータ I/O をしていない。

格子点当たりの計算時間は $\sim 2.4 \times 10^{-8}$ 秒でほぼ一定である。同じコードを AMD Radeon R9 280X (単精度演算性能 3.48 TFLOPS) および NVIDIA Tesla K20c (単精度演算性能 3.52 TFLOPS) で実行した場合、 $N_x = 10000$ の時の、格子点当たりの計算時間は 4.63×10^{-9} および 1.26×10^{-9} 秒であった。現状では、PEZY-SC プロセッサでの MOST の性能は、GPU と比べると劣っているといえる。

3.4 多倍長精度浮動小数点演算の性能評価

我々はこれまで、素粒子物理学の応用で高速な演算が必要とされるファインマン・ダイアグラムの直接計算を、多倍長精度浮動小数点演算で高速化するために様々な手

法 [21], [23] を検討してきた。多倍長精度演算を実現する手法として、今日主に利用されている手法は、(a) 浮動小数点演算 (FP 演算) による多倍長演算手法と、(b) 整数演算による浮動小数点エミュレーション (例えば GNU MPFR[2]) の二種に分類される。ここで、多倍長精度計算浮動小数点演算手法とは、IEEE 754-2008 で規定されている倍精度 binary64 フォーマット (仮数部 $n_{\text{man}} = 53$ ビット、指数部 $n_{\text{exp}} = 11$ ビット) と比べて、仮数部のビット幅が大きい場合を指す。

上記 (a) の手法は、FP 演算の丸め誤差を補償する手法 [1], [7] をベースにしておき、倍精度変数を 2 語利用する double-double (DD) 方式 ($n_{\text{man}} = 105$, $n_{\text{exp}} = 11$ に相当) は、現在の様々な計算機において高速に実行できる [20], [22]。上記 (b) の手法は、整数演算により複数語からなる仮数部の演算を、四則演算それぞれの場合について筆算と同様のアルゴリズムでおこなった。この FP 演算のエミュレーションによる多倍長演算手法では、原理的には指数部、仮数部ともに任意のビット長を利用することができる。ここでは [21] にて報告した (b) の手法による OpenCL カーネル実装 ($(n_{\text{man}} = 210, n_{\text{exp}} = 30)$) の場合に、演算性能の基礎的な評価について報告する。

我々の既存の報告 [21] と同様に、OpenCL API により得られるカーネル実行時間 (これにはホストとアクセラレータ間のデータ転送時間は含まない) により、四則演算ごとに性能を計測した。表 3.4 に、PEZY-SC プロセッサと様々な CPU/GPU での (b) の手法による性能評価の結果を示す。性能の単位は MFLOPS である。2 つ目のコラムは各 GPU の単精度 FP 演算による理論演算性能を示す。この性能評価では、演算ユニットを可能な限り利用するように十分大きな要素数の入力値に対してカーネルを実行して、その実行時間を計測し、総演算数を実行時間で割り、切り捨てることで演算性能を計算した。Xeon E5-2670 のみが CPU であり、他は全てアクセラレータである。Xeon E5-2670 の結果は Sui ren のノードではないが、同じく 2 CPU 構成でトータルで 16 コアのシステムである。除算については、仮数部を直接除算する手法 (除算と示す)、逆数の初期値を単精度で推定するニュートン法 (除算 F と示す)、逆数の初期値を倍精度で推定するニュートン法 (除算 D と示す) の 3 パターンについて比較をした。また、最後の列 (「4 演算」) は、入力に変数について 4 演算 (加算 3 回、乗算 1 回) を続

けておこなった場合の性能をしめす。結果、PEZY-SC プロセッサの性能は演算の種類により比較的高性能の場合もあるのがわかった。特にPEZY-SC プロセッサでは「4 演算」の場合の性能が1 演算の場合よりも高速である。このことから、PEZY-SC プロセッサをファイマン・ダイアグラムの直接計算に適用した場合、高性能であることが予想される。現在、多倍長精度計算浮動小数点演算による現実的な場合での数値積分について検証中である。

3.5 Suiren での今後の課題

本章では、3 種の計算科学アプリケーションと、多倍長精度計算浮動小数点演算について、OpenCL 用に記述されたコードをPEZY-SC プロセッサで実行し、その性能評価をおこなった。現状ではPEZY-SC プロセッサのアーキテクチャに特化した最適は特におこなっていないが、どの場合でも元のコードに大きな変更をする必要なく実行可能であった。GPU で同じコードを実行した場合と比べて、PEZY-SC プロセッサの性能が十分発揮できていない場合がある。考えられる理由および、現状の性能におけるボトルネックや制限は以下の通りである。

- DRAM とのメモリ帯域が最新の GPU と比べると低い。
- 単精度でも倍精度演算でも、ピーク性能となるのは積和算をおこなう場合であり、今回のアプリケーションでは積和算の占める割合が多くはない。また、この点、カーネル用コンパイラに命令最適化の余地がある。
- 単精度演算の場合、GPU(280X と K20c) はいずれもスレッドあたりのベクトルレーン数が1 なのに対して、PEZY-SC プロセッサでは単精度演算は2 命令まで同時実行可能であり、ベクトルレーン数が実質的に2 である。
- 多倍長精度計算浮動小数点演算については整数演算の組み合わせでおこなうため、カーネル用コンパイラに命令最適化の余地がある。

今後、以上のようなボトルネックなどについてより詳しく検討し、Suiren 全体を利用する大規模アプリケーションを実現する予定である。

4. まとめ

本研究報告では、我々がこれまで主に GPU で実行してきた様々な計算科学アプリケーションを MIMD 型 PEZY-SC プロセッサを採用したアクセラレータ型スーパーコンピュータ Suiren にて性能評価をおこなった。OpenCL で記述された計算科学アプリケーションは大きな修正なく、PEZY-SC 用のプログラミング環境である PZCL で動作した。今回は、全てのケースにおいて PEZY-SC プロセッサに特化した最適化はおこなわずに性能評価をおこなった。

高精度天文多体シミュレーションでは、OpenCL で記述されたライブラリ OCLG6A を、既存の MPI 並列化コード

ϕ -GRAPE と組み合わせ、Suiren の 4 ノードまでを利用し、9.18 TFLOPS の性能を得た。ツリー法による重力ポテンシャル計算と SPH 法のシミュレーションでは、OpenCL による八分木のツリーワークは比較的効率よく PEZY-SC プロセッサで実行可能なことがわかった。また、OpenCL による SPH 法シミュレーションコード OTOO による白色矮星シミュレーションでは、複数の PEZY-SC プロセッサを使った場合の性能評価をおこなった。津波進化シミュレーションの PEZY-SC プロセッサでの性能は、他のアクセラレータと比べて性能があまりよくないため、アーキテクチャに特化した最適化が必要である。最後に、整数演算による多倍長浮動小数点演算エミュレーション手法の PEZY-SC プロセッサでの性能は、他のアクセラレータと比べて遜色はないこともわかった。

今後、Suiren や他の PEZY-SC プロセッサを採用したスーパーコンピュータにて、大規模計算科学アプリケーションを実行するために、PZCL によるプログラミングの制限やボトルネックの解消、またアーキテクチャに応じた最適化手法を調べる必要がある。

参考文献

- [1] Dekker, T.: A Floating-Point Technique for Extending the Available Precision, *Numerische Mathematik*, Vol. 18, pp. 224–242 (1971).
- [2] Fousse, L., Hanrot, G., Lefèvre, V., Pélissier, P. and Zimmermann, P.: MPFR: A Multiple-precision Binary Floating-point Library with Correct Rounding, *ACM Trans. Math. Softw.*, Vol. 33, No. 2, pp. 1–15 (online), DOI: 10.1145/1236463.1236468 (2007).
- [3] Gaburov, E., Harfst, S. and Zwart, S. P.: SAPPORO: A way to turn your graphics cards into a GRAPE-6, *New Astronomy*, Vol. 14, No. 7, pp. 630 – 637 (online), DOI: <http://dx.doi.org/10.1016/j.newast.2009.03.002> (2009).
- [4] Gingold, R. A. and Monaghan, J. J.: Smoothed Particle Hydrodynamics: Theory and Application to Non-spherical Stars, *Monthly Notices of the Royal Astronomical Society*, Vol. 181, pp. 375–389 (1977).
- [5] Harfst, S., Gualandris, A., Merritt, D., Spurzem, R., Portegies Zwart, S. and Berczik, P.: Performance analysis of direct N-body algorithms on special-purpose supercomputers, *New Astronomy*, Vol. 12, pp. 357–377 (online), DOI: 10.1016/j.newast.2006.11.003 (2007).
- [6] Heggie, D. and Hut, P.: *The Gravitational Million-Body Problem: A Multidisciplinary Approach to Star Cluster Dynamics*, Cambridge University Press (2003).
- [7] Knuth, D.: *The Art of Computer Programming vol.2 Seminumerical Algorithms*, Addison Wesley, Reading, Massachusetts, first edition (1998).
- [8] Kono, F., Nakasato, N., Hayashi, K., Vazhenin, A., Sedukhin, S., Nagasu, K., Sano, K. and Titov, V.: PARALLELIZATION OF TSUNAMI SIMULATION ON CPU, GPU AND FPGAS, *Supercomputing '15: Proceedings of the 2015 ACM/IEEE conference on Supercomputing*, ACM, pp. 1–2 (2015).
- [9] Lucy, L. B.: A Numerical Approach to the Testing of the Fission Hypothesis, *Astronomical Journal*, Vol. 82, pp. 1013–1024 (online), DOI: 10.1086/112164 (1977).

	SP 性能	加算	乗算	除算	除算 F	除算 D	4 演算
Xeon E5-2670	3.3e5	247	189	16.7	34.5	19.2	254
Radeon R9 R280X	4.2e6	729	1825	22	47	241	461
FirePro W9100	5.2e6	942	525	28.6	52.0	28.1	597
FirePro W8100	4.2e6	980	450	28.1	52.2	47.1	610
PEZY-SC	3.0e6	144	130	49.3	68.4	60.1	423

表 3 MYFP 方式 ($n_{\text{man}} = 210$, $n_{\text{exp}} = 30$) の性能評価. 単位は MFLOPS.

- [10] Makino, J.: Optimal order and time-step criterion for Aarseth-type N-body integrators, *Astrophysical Journal*, Vol. 369, pp. 200–212 (online), DOI: 10.1086/169751 (1991).
- [11] Makino, J. and Taiji, M.: *Scientific Simulations with Special-Purpose Computers—the GRAPE Systems*, Scientific Simulations with Special-Purpose Computers—the GRAPE Systems, by Junichiro Makino, Makoto Taiji, pp. 248. ISBN 0-471-96946-X. Wiley-VCH, April 1998. (1998).
- [12] Makino, J., Daisaka, H., Fukushige, T., Sugawara, Y., Inaba, M. and Hiraki, K.: The performance of GRAPE-DR for dense matrix operations, *Procedia Computer Science*, Vol. 4, pp. 888 – 897 (online), DOI: <http://dx.doi.org/10.1016/j.procs.2011.04.094> (2011). Proceedings of the International Conference on Computational Science, ICCS 2011.
- [13] Nakasato, N.: Implementation of a parallel tree method on a GPU, *Journal of Computational Science*, Vol. 3, No. 3, pp. 132 – 141 (online), DOI: <http://dx.doi.org/10.1016/j.jocs.2011.01.006> (2012).
- [14] Sato, Y., Nakasato, N., Tanikawa, A., Ken'ichi-Nomoto, Maeda, K., Hachisu, I.: A Systematic Study of Carbon-Oxygen White Dwarf Mergers: Mass Combinations for Type Ia Supernovae, *The Astrophysical Journal*, Vol. 807, No. 1, p. 105 (online), available from (<http://stacks.iop.org/0004-637X/807/i=1/a=105>) (2015).
- [15] Sugimoto, D., Chikada, Y., Makino, J., Ito, T., Ebisuzaki, T. and Umemura, M.: A Special-Purpose Computer for Gravitational Many-Body Problems, *Nature*, Vol. 345, pp. 33–35 (1990).
- [16] Tanikawa, A., Nakasato, N., Sato, Y., Ken'ichi-Nomoto, Maeda, K., Hachisu, I.: Hydrodynamical Evolution of Merging Carbon-Oxygen White Dwarfs: Their Pre-supernova Structure and Observational Counterparts, *The Astrophysical Journal*, Vol. 807, No. 1, p. 40 (online), available from (<http://stacks.iop.org/0004-637X/807/i=1/a=40>) (2015).
- [17] Titov, V.: Numerical modeling of tsunami propagation by using variable grid, *Proceedings of the IUGG/IOC International Tsunami Symposium*, pp. 46–51 (1989).
- [18] Titov, V. and Gonzalez, F.: Implementation and testing of the method of splitting tsunami (MOST) model, *NOAA Technical Memorandum ERL PMEL-112* (1997).
- [19] 河野郁也, 中里直人, 林 憲作, Alexander, V., Stanislav, S.: MOST 法による津波シミュレーションの OpenMP 並列化とその性能評価, 研究報告ハイパフォーマンスコンピューティング(HPC), Vol. 2014, No. 24, pp. 1–6(オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110009827575/>) (2014).
- [20] 山田 進, 佐々成正, 今村俊幸, 町田昌彦: 4 倍精度基本線形代数ルーチン群 QPBLAS の紹介とアプリケーションへの応用, 情報処理学会研究報告. 計算機アーキテクチャ研究会報告, Vol. 2012, No. 23, pp. 1–6 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110009490634/>) (2012).
- [21] 中里直人: 整数演算による多倍長浮動小数点演算エミュレーションの GPU での性能評価, 情報処理学会研究報告. [ハイパフォーマンスコンピューティング], Vol. 2015, No. 8, pp. 1–6 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110009877717/>) (2015).
- [22] 中里直人, 石川 正, 牧野淳一郎, 湯浅富久子: アクセラレータによる四倍精度演算, 情報処理学会研究報告. [ハイパフォーマンスコンピューティング], Vol. 2009, No. 39, pp. 1–7 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110007995434/>) (2009).
- [23] 台坂 博, 中里直人, 石川 正, 湯浅富久子: 多倍長専用計算機 GRAPE9-MPX の拡張とその性能評価, 情報処理学会研究報告. [ハイパフォーマンスコンピューティング], Vol. 2015, No. 7, pp. 1–7 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110009877716/>) (2015).