

# ポアソン方程式における FPGA アレイ実装回路と CUDA の比較

Performance Comparison of FPGA Array with CUDA on Poisson Equation

- 黎 江 (lijiang@sekine-lab.ei.tuat.ac.jp), 佐藤 一輝 (kazuki@sekine-lab.ei.tuat.ac.jp),  
高橋 健一 (takahashi@sekine-lab.ei.tuat.ac.jp), 田向 権 (tamukoh@cc.tuat.ac.jp),  
小林 祐一 (yu-koba@cc.tuat.ac.jp), 関根 優年 (sekinem@cc.tuat.ac.jp)  
東京農工大学 工学府 〒184-8588 東京都小金井市中町 2-24-16

Li Jiang, Kazuki Sato, Kenichi Takahashi, Tamukoh Hakaru, Yuuichi Kobayashi, Sekine Masatoshi  
Tokyo University of Agriculture and Technology 2-24-16 Naka-chou, Koganei-shi, Tokyo, 〒184-8588 Japan

Abstract In recent years, the examples which use FPGA or GPGPU for the HPC use are increasing. We propose an FPGA array which accumulated a lot of small cards with the three-dimensional I/O that installed large-scale FPGA. The FPGA array is suited to the scalable design, and it is possible to control from the host PC easily. As a contrast, we also structured CUDA system by GeForce 9800GT. In this paper, we implemented FPGA array and CUDA to calculate Poisson equation by the finite difference floating point number method, and the performance and power consumption are presented. We also discuss the result which from the different hardware architecture and the advantages between in FPGA and GPGPU.

## 1. はじめに

HPC とは、膨大な計算を行うことを指し、例えば、バイオ・流体解析・物性・気象などといった様々な分野でのシミュレーションにおいて必要とされる。現在、多くの HPC サーバに採用されるプロセッサは、x86 や POWER のアーキテクチャをベースにした汎用プロセッサである。汎用プロセッサは、過去の資産を生かすための拡張を続けた結果回路が複雑化し、例えば、消費電力の面で効率的ではなくなっている。

この問題を受け、近年では、特定の計算をアクセラレートするコプロセッサを搭載したり、ある用途に特化したプロセッサを HPC 用途に採用する動きも出現している。特に、本来は画像処理に用いるデバイスである GPU を汎用的な計算に利用する GPGPU (General Purpose computing on GPU) についての報告が多くなされるようになった<sup>(1)</sup>。しかしながら、GPGPU の演算能力の強さについての報告は多くなされているものの、GPU は基本的に単純なデータを一度に大量に処理することに特化しているため、その特性と合わないプログラムは性能を期待できないという問題を抱えている。また、HPC 用途としては、汎用性が十分とは言えない面がある。

このような動向の中、近年では、HPC (High Performance Computing) のために再構成可能な LSI である FPGA (Field Programmable Gate Array) を用いる研究が見られるようになった。近年、半導体プロセス技術は、大容量化、高性能化、低コスト化などの要求に応えるよう急速に進歩しており、半導体デバイスの 1 つである FPGA にも最新技術が導入されている。これまで以上のロジック素子とメモリの組み込みが可能となり、同時にチップ面積も低減され、FPGA の集積度・回路規模が向上している。FPGA が高性能かつ低価格となった背景を受けて、FPGA を用いて様々な研究が行われるようになってきている。

そこで我々は、FPGA アレイによる HPC システムを提案してきた。HPC の中でも、FPGA 中の演算回路のように、回路の再構成性を利用するような方法は、RHPC (Reconfigurable High Performance Computing) と呼ばれている。FPGA のような再構成可能なデバイスを HPC に用いる利点には、対象問題にあわせて最適な演算器をその都度用意することができる点がある。FPGA を HPC 用途に用いる欠点としては、PC とは違い演算環境が整っていない、CPU より動作周波数が低い、利用者が独自にシステムを構築する必要がある、といったことがある。我々は、それらの欠点を補うために、汎用ソフトウェアと FPGA を組み合わせた hw/sw 複合体という考え方を提案してきた<sup>(2)(3)</sup>。

本稿では、提案する FPGA アレイシステムの有用性を

示すため、ポアソン方程式の演算結果において、FPGA による演算結果と GPU 向けのプログラミング環境である CUDA による演算結果との比較について報告する。

## 2. 差分法によるポアソン方程式の解法

ポアソン方程式は、電磁気学の問題等に応用される、楕円型の偏微分方程式である。

ポアソン方程式を数値解析で解く方式は、データ並列型の大規模演算である。本稿では、これを提案システムの評価に使用する。

今回は、Fig.1 に示すような 2 次元平面の格子点上に電荷を配置したときの電場の様子をシミュレーションすることを想定する。

$\phi$  を電位、 $\rho$  を電荷密度とすると、ポアソン方程式は、次式で表される。

$$\Delta\phi = \rho \quad (1)$$

式 (1) は、反復法によって次のように変形できる。

$$\phi_{i,j}^{new} = \frac{\alpha}{4} \quad \alpha = h^2\rho + \phi_{i-1,j}^{old} + \phi_{i+1,j}^{old} + \phi_{i,j-1}^{old} + \phi_{i,j+1}^{old} \quad (2)$$

本稿では、提案 FPGA アレイシステム、およびその比較対象とする CUDA を用いるシステムによって、この式 (2) の演算を行う。

● : Node (格子点)

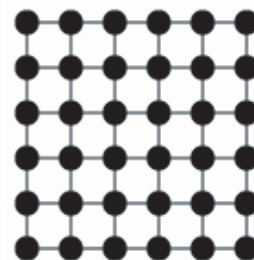


Fig. 1: ポアソン方程式演算の格子点のイメージ

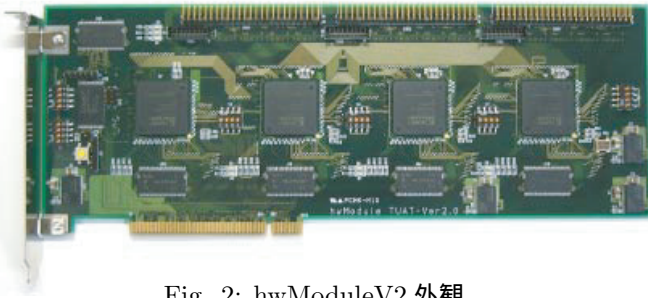


Fig. 2: hwModuleV2 外観

### 3. Reconfigurable HPC システム

#### 3.1 hw/sw 複合体

我々は、ハードウェアとソフトウェア、双方の利点を活かしたアプリケーションを実現するために、hw/sw 複合体という考え方を提案している。hw/sw 複合体は、FPGA 中の回路と、ホスト PC 上で動作するソフトウェア環境によって構成される。基本的に、演算負荷の高い処理は FPGA 中の回路によって行い、複雑な分岐処理といった回路が苦手とする処理はソフトウェアによって行う。これにより、今まで実現の難しかったアプリケーションの実現を容易にする。また、ソフトウェア部によって、ホスト PC から FPGA 中の回路への容易なアクセスが提供されるので、専門的な知識が無くとも回路利用が簡単に行えるようになっている。

HPC 用途に FPGA を利用する利点の一つに、対象となる問題にあわせて演算回路をスケラブルに用意することができるということがある。提案 FPGA アレイシステムでは、hw/sw 複合体を通してこの演算回路にアクセスできるようにすることで、スケラブルでありながらも容易な制御を可能にしている。

hw/sw 複合体を実現するために、本研究室では hwModule と呼ばれる FPGA ボードを使用している。我々は、この hwModule を用いて、hw/sw 複合体を用いたアプリケーションについての研究報告を行ってきた。本稿で述べる FPGA アレイシステムについても、hw/sw 複合体を用いる理由とその有用性についての報告がなされている (2)(3)。

#### 3.2 hwModule V2

提案 FPGA アレイシステムでは、hwModule V2 (Fig.2) を介して FPGA アレイとホスト PC とを接続する。hw/sw 複合体を hwModule V2 に実装し、ホスト PC からは hw/sw 複合体を通して FPGA アレイにアクセスする形を取る。hwModule V2 は、ホスト PC-FPGA アレイ間のゲートウェイのような役割を持ち、演算回路のリコンフィグレーション、マネージメントなどを行う。

#### 3.3 hwModule VS

FPGA アレイをスケラブルに実現するため、hwModule VS と呼ばれる小型の FPGA ボードを作成した。外観を Fig.3 に示す。この hwModule VS を、計算の規模に合わせて集積することで、スケラブル性を実現する。

hwModule VS は、Fig.4 に示す PE Board と Sub Board の 2 枚のボードによって構成されている。Fig.4 中に、複数の接続端子が示されている。この接続端子が、hwModule VS の 2 次元ないし 3 次元方向の積層を可能にする。

#### 3.4 FPGA アレイ

本稿では、積層された hwModule VS のことを FPGA アレイと呼ぶ。Fig.5 に 3 次元立体構成を取った FPGA アレイの外観を示す。

Fig.6 に、hwModule V2 を経由して FPGA アレイにアクセスする様子を示した。Software(CPU) は、ホスト PC のものである。hw/sw 複合体を使用することで、ユー

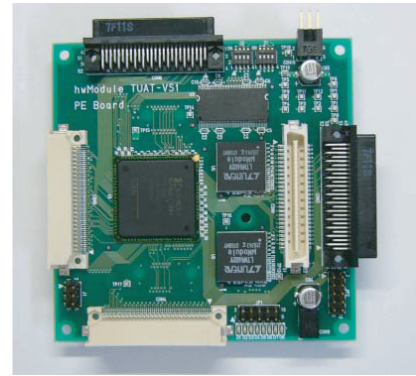


Fig. 3: hwModule VS 外観

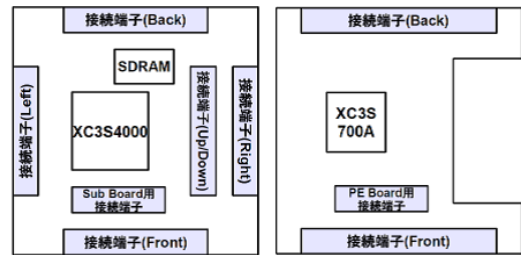


Fig. 4: 左 : PE Board 右 : Sub Board

ザーは FPGA アレイを簡便に利用することができる。

#### 3.5 実装したポアソン方程式演算回路について

Fig.7 に FPGA 中に実装したポアソン方程式演算回路の演算器コアの構成を示す。Fig.7 に示すように、演算器コアは加算器 4 個と除算器 1 個を持つ。この演算器コア (Processing Element:PE) は、40 段のパイプラインで構成されている。パイプライン構成にすることで、動作周波数の向上も実現されており、本実装では 66.6[MHz] の動作周波数となっている。各 PE の扱うデータは、IEEE754 に準拠した単精度浮動小数点数となっている。

演算にあたって、電位  $\phi$  や電荷  $\rho$  は、頻繁に参照、更新される。そのため、FPGA 内部に搭載される Block-RAM(BRAM) にその値を保存するようにした。Fig.8 に、演算回路全体のブロック図を示した。各 PE は、BRAM によって構成される Cache から値を読み出し、演算結果を書きこむような構成となっている。BRAM の容量は大きくはないものの FPGA 中の回路から高速に読み書きできる。BRAM による Cache を設けることで、ホスト PC のメインメモリや、hwModule 上の SDRAM へのアクセス回数を低減、パイプライン動作がなるべく連続して行えるようにしている。本実装においては、FPGA には Xilinx 社製の XC3S4000 を用いている。XC3S4000 に搭載されている BRAM は、2 ポート構成を取ることができるので、片方のポートを入力用、もう片方のポートを出力用としている。また、BRAM1 個あたりの容量は、32[bit] データ  $\times$  512[word] である。

2 次元の座標平面の演算上、必要とされる Cache について述べる。電荷  $\rho$  用には PE の数と同数個の BRAM を用意している。電位  $\phi$  については、計算点の隣接点のデータも必要であるため (PE の数) + 2 個の BRAM を用意する。本稿では、PE の数が 1 個の場合と 10 個の場合について実験を行う。PE が 1 個のときは、電荷  $\rho$  用の BRAM は 3 個、電位  $\phi$  用の BRAM は 1 個用意する。同様に、PE が 10 個のときは、電荷  $\rho$  用に 12 個、電位  $\phi$  用には 10 個用意する。

本実装におけるポアソン方程式演算回路の回路規模は、XC3S4000 において、PE1 個のとき回路使用率約 11[%]、

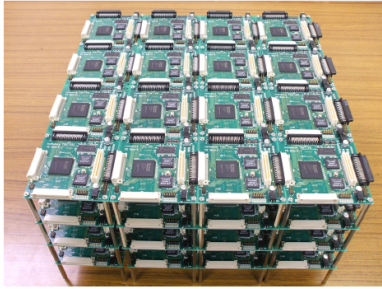


Fig. 5: 立体構成の FPGA アレイ外観.

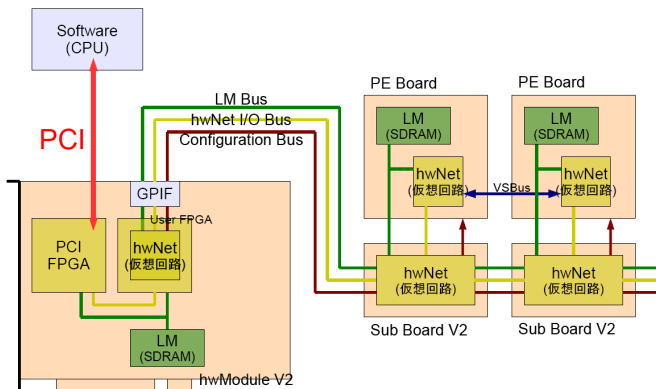


Fig. 6: hw/sw 複合体による FPGA アレイへのアクセス

PE10 個のとき約 85[%] である．なお，PE1 個あたりの回路規模は約 7[%] である．

今回の実験では，FPGA あたりの PE が 1 個の場合と 10 個の場合との 2 通りの場合について実験を行った．PE が 1 個の場合は  $20 \times 80$  の計算領域，PE が 10 個の場合は  $60 \times 60$  の計算領域について計算を行い，それぞれの場合の演算性能を見る．

PE10 個の場合は，hwModule VS の外部バス等について未検証の部分があるため，FPGA アレイには適用しない．しかし，単一の FPGA に PE10 個を搭載することで，FPGA アレイにした場合の性能を測定できるので，今回は hwModule V2 を用いて測定を行う．PE1 個の場合は，hwModule VS 1 枚あたり 1 個の PE を実装したものを 4 枚接続した FPGA アレイによって実験を行う．

Fig.9 に，PE10 個のときの演算における，計算格子点の走査順序についてのイメージを示した． $60 \times 60$  の計算領域については，6 回の横走査を終えたとき，1 回の計算が完了するイメージである．

#### 4. GPGPU

##### 4.1 使用する GPU の構成

本研究で用いたグラフィックスカード (NVIDIA 製の GeForce 9800GT) の GPU の構成を Fig.11 に示す．このグラフィックスカードの GPU には，G92 が使用されている [6] ．

HPC 用途において，ビデオメモリは変数の値を格納するために使用する．CUDA では Global Memory と呼ばれる．今回の実験に使用したグラフィックスカードでは，512[MB] の GDDR3 メモリである．

GPU 上には複数の Multi Processor が配置されている．ひとつの Multi Processor は，8 つの Streaming Processor によって構成される．Streaming Processor は，最小単位の演算ユニットであり，浮動小数点演算 (加算および乗算) を行わせることができる．

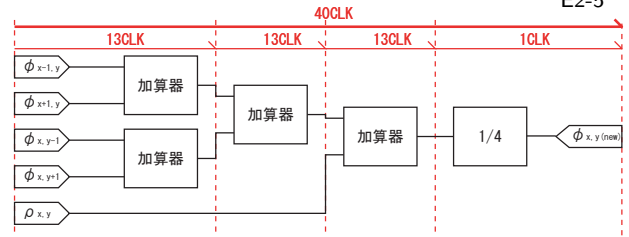


Fig. 7: FPGA 中のポアソン方程式演算回路 (コア)

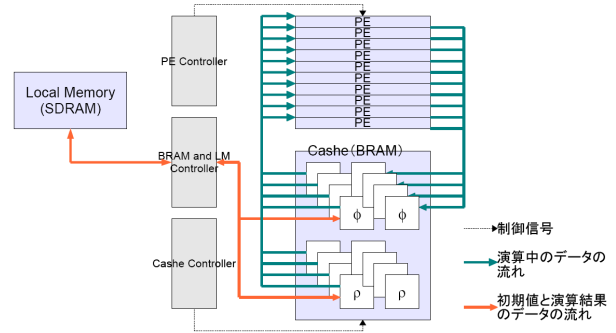


Fig. 8: 演算回路全体の構成

また，各 Multi Processor は，内部にひとつずつ Shared Memory を持つ．Shared Memory の容量は 16[KB] と小さいが，Global Memory よりも高速にアクセスできる (3) ．

今回使用したハードウェアは，全部で 14 個の Multi Processor を持つ．各 Multi Processor は 8 個ずつ Streaming Processor を持つので，合計 112 個の Streaming Processor があることになる．

##### 4.2 CUDA における grid,block,thread について

CUDA には grid,block,thread という概念があり，CUDA プログラミングにおいては，これらの概念を意識する必要がある．Fig.12 に構成例を示す．

CUDA では，GPU 上の Streaming Processor を効率的に使用するために，プログラムはスレッドとして実行される．例えば，計算格子を分割し各 block に割り当て，各 block は，担当する範囲の計算を thread を単位として行う，といった具合である．ハードウェア的には，1 つの Multi Processor には，一度に 1 つの block の処理が割り当てられる．Multi Processor が，一度に全ての block を処理するのに不足する場合は，結果的に 1 つの Multi Processor が複数 block の処理を担当することもある．block をまとめたものを grid という．

マルチコア CPU におけるマルチスレッド計算の場合，スレッド生成実行のオーバーヘッドが大きい．そのため，おおむねスレッド数は CPU のコア数に等しい場合に効率が良い．GPU の場合は，スレッド生成のコストが小さく，細粒度のスレッドが大量に生成される．効率的に処理が成されるよう，ジョブスケジューラによって，それら大量のスレッドに処理が割り振られる (6) ．

##### 4.3 CUDA によるポアソン方程式演算プログラムの実装

本研究では，CUDA を用いて差分法によるポアソン方程式演算を行い，提案 FPGA アレイの比較対象とする．計算には式 (2) を用いる．各点の計算を行う時，x 方向と y 方向のそれぞれの隣接点を参照しつつ，計算を行っていく．

計算格子の 1 行を 1block に割り当て，各 block の内部には，計算格子 1 行の要素数分の thread を生成する．1 つの thread が，1 つの計算格子点に対応する．

例えば， $64 \times 64$  の格子点を持つ平面の演算をする場合，block 数は格子点平面の行数と同じ 64 とし，毎 block に

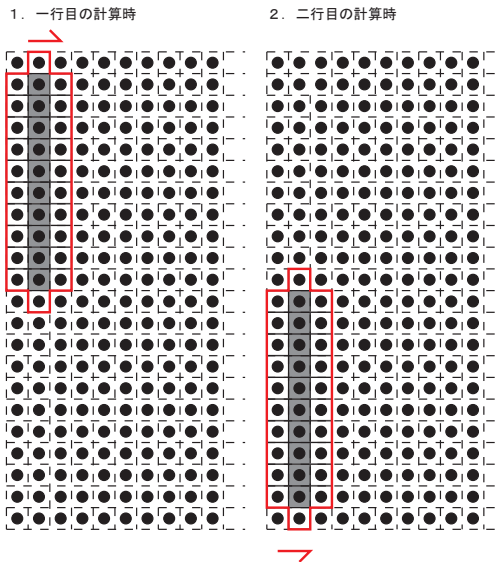


Fig. 9: PE10 個のときの演算方法のイメージ

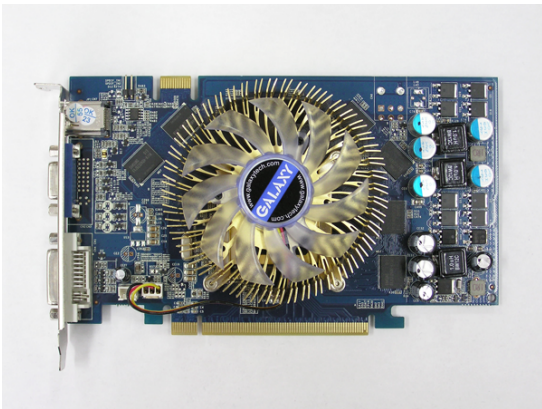


Fig. 10: 使用した GPU(GeForce 9800GT)

64 個の thread を持たせる．この場合，総スレッド数は  $64 \times 64 = 4,096$  となる．

CUDA プログラミングにおいて，演算アルゴリズムを考えると，GPU の特性を考えに入れるべきである．例えば，CUDA を用いる方法の高速性は，基本的には，GPU の多数の演算ユニットを並列的に使用できることに支えられている．つまり，並列的に演算ユニットを利用できない類の計算には，威力を発揮しにくい．また，個々のユニットの演算速度，浮動小数点演算回数，メモリの操作回数といった要素もある．

差分法によるポアソン方程式演算に関して，様々な CUDA の最適化方法が考えられる．本実装では，全てのデータを（大容量だが低速な）Global Memory に保存するというシンプルな仕様とした．このため，Global Memory からのロードが不連続になり伝送速度が落ちることが考えられる．Shared Memory をキャッシュとして効果的に使用することで，アクセス回数や伝送量を削減できる可能性がある．

## 5. 性能評価

### 5.1 提案 FPGA システムの演算性能

FPGA 中の回路ポアソン方程式の演算制御および演算結果の表示のため，ホスト PC にて動作するソフトウェアを製作した．Fig.13 に外観を示す．これは，回路の操作および演算結果の可視化に使用する．

Tab.1 に FPGA による演算の実験に用いた環境を示す．今回の実験では，FPGA 中の回路に PE を 1 個実装した場合と，10 個実装した場合について，それぞれの演算

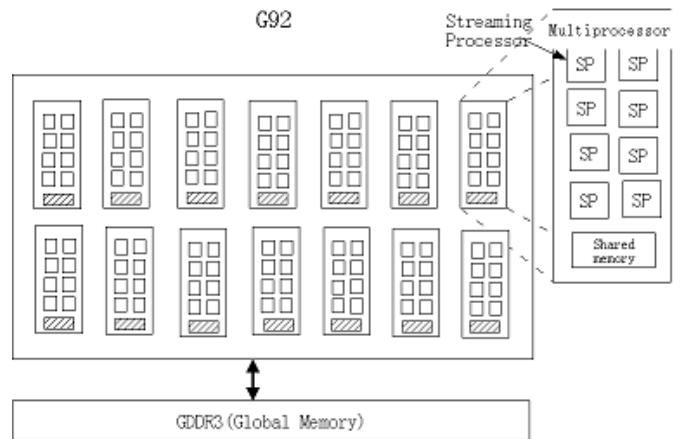


Fig. 11: GPU のハードウェア構成

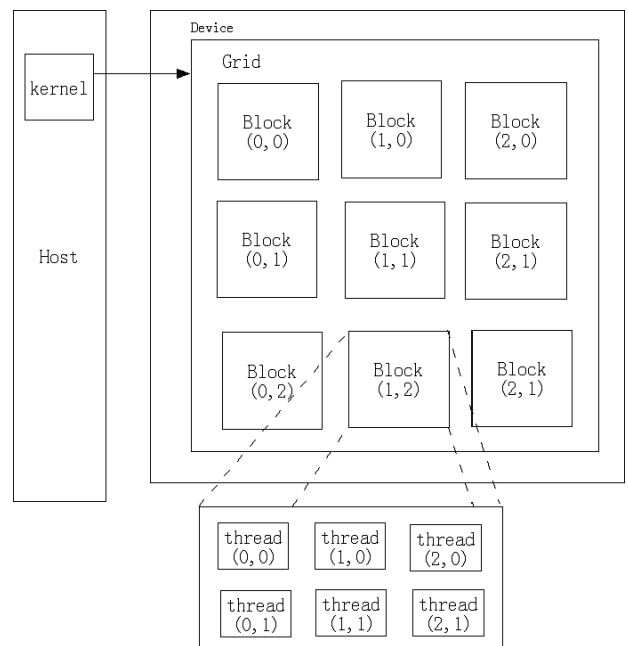


Fig. 12: grid, block, thread の構成例

性能を測定した．得られた結果は，後述の Fig.15 に示してある．おおむね，ソフトウェア (CPU) と比較して高い性能が得られた．この結果については後述する．

今回，PE10 個の回路は hwModule V2 に実装した．hwModule VS を用いなかった主な理由は，現在は hwModule VS 間での通信と演算が同時に行える仕様になっておらず，良い結果が期待できないからである．演算と通信を同時に行うことができる状態で，hwModule VS を 4 枚用いた FPGA アレイを構築すれば，10 倍の性能が得られるという試算がある<sup>(2)</sup>．

### 5.2 CUDA の演算性能

CUDA を用いて，ポアソン方程式演算を行った結果について述べる．

CUDA を用いる場合に使用した環境について Tab.2 に示す．

ここでは，GPU と CPU との演算性能比較について述べる．結果をグラフにしたものを Fig.14 に示す．

実験の条件について述べる．GPU の演算性能については，前述の条件で CUDA プログラムを作成して測定に用いた．CPU の方は，C++ によるポアソン方程式演算プログラムを作成して用いた．計算回数は 10,000 回で固定とした．Fig.14 は，1 回目の計算を始めてから 10,000 回目の計算が終了するまでの時間から計算した値である．

Tab. 1: FPGA システムの動作環境

CPU	Athlon64 X2 3800+ (2.0[GHz])
メインメモリ	DDR2 SDRAM PC2-6400 2[GB]
マザーボード	ASUS M2A-VM HDMI (AMD 690G 搭載)
OS	Windows XP Professional SP3
コンパイラ	Borland C++ Builder 2006

Tab. 2: CUDA の動作環境

CPU	Athlon64 X2 3800+ (2.0[GHz])
メインメモリ	DDR2 SDRAM PC2-6400 2[GB]
マザーボード	ASUS M2A-VM HDMI (AMD 690G 搭載)
OS	Windows XP Professional SP3
コンパイラ	MicroSoft Visual Studio 2008
グラフィックカード	GALAXY GeForce 9800GT (GDDR3 512[MB])

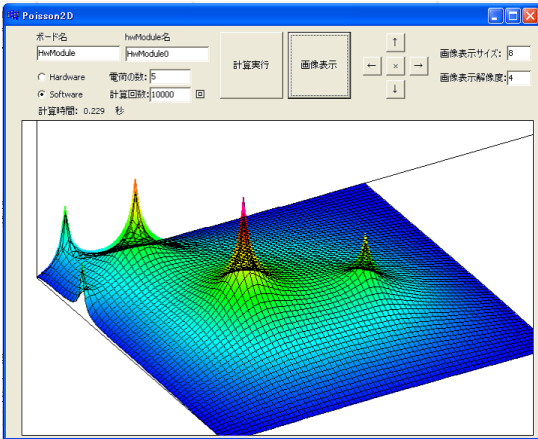


Fig. 13: ポアソン方程式演算アプリケーション

計算前に Global Memory に値をコピーするが、その時間は測定に含めない。

Fig.14 において、計算 matrix size が  $16 \times 16$  の時、GPU の演算性能は  $0.136[\text{GFlops}]$  であり、計算 matrix size が大きくなるほど GPU の演算性能は上昇していき、 $64 \times 64$  のとき、GPU の性能は CPU と同程度になる。 $128 \times 128$  の場合 GPU は CPU の性能の  $1.84$  倍、 $256 \times 256$  では  $2.86$  倍の性能となっている。

計算 matrix size が小さいときは、CUDA において生成される thread 数も少ない。例えば、 $16 \times 16$  の thread 数は  $256$  個である。これは、並列に行われる加算の量が少ないということにつながる。Multi Processor に値を渡すまではオーバーヘッドがあるが、thread 数が小さい状況では、オーバーヘッドをカバーできるほどの並列計算が成されていないと考えられる。

Fig.14 から、計算領域が大きくなると、GPU の方が CPU と比較して高い演算性能を示していることが見て取れる。 $128 \times 128$  のとき、GPU の性能は  $1.4[\text{GFlops}]$  を示している。

GPU の演算性能は、 $128 \times 128$  以下の計算 matrix size のときは、計算 matrix size が大きいほど高くなっている。しかし、それ以上の計算 matrix size での演算性能は同程度である。これは、GPU 内の  $14$  個の Multi Processor を使い切ってしまうからと考えられる。 $128 \times 128$  以上の計算 matrix size では、Multi Processor の使用効率は最大であり、それ以上の並列計算ができないことから、演算性能が頭打ちになっていると考えられる。 $128 \times 128$  のときの生成スレッド数は  $16,384$  個となるはずであるので、並列計算を一度に行うための thread 数限度はそれ以下であると推測される。

### 5.3 FPGA, GPU, CPU の演算性能についての考察

以上の結果より、FPGA, GPU, CPU の演算性能について考察をする。

FPGA, GPU, CPU の演算回数と Flops 数の関係を測定し、Fig.15 にまとめた。各グラフによって、計算 matrix size が異なるので注意されたい。GPU の場合は、 $128 \times 128$  と  $16 \times 16$  の 2 つの場合について示してある。他の場合、つまり、凡例ラベル CPU および FPGA(10PE) について

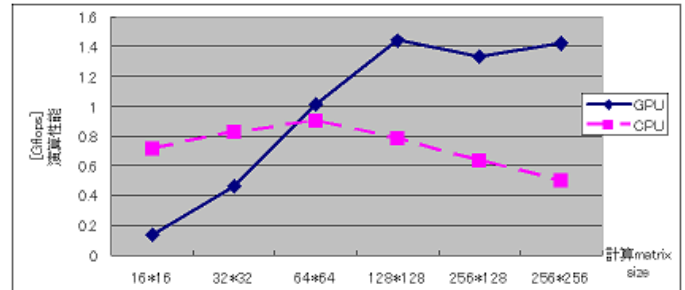


Fig. 14: GPU と CPU の演算性能比較

は、 $60 \times 60$  の計算 matrix size についての演算性能である。FPGA(10PE)の結果は、hwModule V2 に回路を実装して測定したものである。FPGA(1PE)の結果は、4 つの hwModule VS を接続した FPGA アレイを用いたものである。1 つの VS に 1 つずつ PE を実装したので、用いた FPGA アレイは合計 4 つの PE を持つ。4 つの PE は、全部で  $20 \times 80$  の領域を並列的に計算する。1 つの PE は  $20 \times 20$  の領域について計算を行う。

GPU の場合は、演算性能と計算回数との間に特に関係性は見られないグラフになっている。前節の実験結果より、 $128 \times 128$  の場合は、thread が大量に生成され、G92 の全体の  $14$  個 Multiprocessor と、その中の Streaming Processor を十分に利用していると考えられる。これより、計算回数を増加させても演算性能が高まるということと考えにくい。

FPGA の演算性能を見ると、計算回数が少ないときは、演算性能がよくない。しかし、PE が 1 つのときであっても、計算回数が多いときは CPU よりも高い性能を示している。PE が  $10$  個のときは、演算性能が高いときは CPU の  $3.4$  倍、GPU の  $2.2$  倍の性能を示している。本実装におけるポアソン方程式回路は、制御のために一定の時間がかかってしまう。そのため、計算回数が小さいときは、計算時間に占める制御時間の割合が高いため演算性能が低く出ることがあるが、演算回数が多いときは GPU よりも演算性能が高くなっている。前述のように、FPGA は対象問題に合わせて最適な演算器を用意できることが利点としてある。この結果は、あくまで製作した CUDA プログラムと比較した場合のものであるが、提案 FPGA アレイシステムの汎用性を示すものであると考えている。

また、先行研究において、FPGA アレイの演算性能はさらに大きく向上する可能性があることが示唆されている<sup>(2)</sup>。参考文献中では、 $150$  個の hwModule VS で FPGA アレイを構築したとすると、 $1[\text{TFlops}]$  の演算性能が達成できるとされている。

今回使用したグラフィックカード ( GeForce 9800GT ) の理論的な最大 Flops 数が  $462[\text{GFlops}]$  である。もちろん、理論値どおりの性能が発揮できる局面はそう多くはないと予想されるが、今回製作した CUDA プログラムについては、ポアソン方程式演算における性能向上の余地がある。製作した CUDA プログラムは基本的な並列計算のみを意識したものであり、ポアソン方程式演算において最適な手法が盛り込まれているとは言いがたい。全ての

Tab. 4: 消費電力の比較.

	性能 [GFlops]	消費電力 [W]	性能/消費電力 [MFlops/W]
FPGA アレイ	3.226	7.92	407.3
GPU(GeForce 9800GT)	1.442	66.0	21.85
CPU(Athlon64 X2)	0.954	62.0	15.39

いという問題はある．HPC 用途ということ考えたときに，GPU の場合は，CUDA のような開発環境があるが，FPGA については，汎用的な開発環境は普及していないのが現状である．本稿において，CUDA による GPU 利用システムと比較することで，hw/sw 複合体による FPGA 利用システムの汎用性のある程度示すことができたと考えている．

#### 6. おわりに

本稿では，多数の FPGA をアレイ上に集積したシステムを HPC 用途に適用することを提案し，その演算性能について検証するため，ポアソン方程式の演算という同一の課題について，CUDA を用いた GPGPU 手法との比較を行った．結果，計算回数の多い場合において，CPU および GPU と比較した提案 FPGA アレイの優位性を示すことができた．

本稿では，hwModule V2 に最大 10 個の PE を実装した場合の演算結果を用いた．今後は，hwModule VS を 2 次元，3 次元的に接続し，多数の PE を搭載して演算回路を大規模化することによって，更に高い演算性能の実現を目指していく．また，提案 FPGA アレイシステムと CUDA や他の HPC との演算性能比較も引き続き行っていく．

提案した FPGA アレイは，様々な大規模演算へと応用可能である．例えば脳機能の実時間シミュレーションは，自律ロボットの制御には不可欠であるが，大規模な強化学習や人工神経回路網等の実時間処理は現在の汎用計算機には不可能である．本稿での検討で，演算速度や消費電力の観点から，FPGA アレイは自律移動ロボットの脳として有望であることが考えられる．HPC としての FPGA アレイの方式検討と共に，これらの具体的な応用も平行して進め，アプリケーションレベルでの有用性や課題を検討する予定である．

#### 参考文献

- (1) nVIDIA CUDA, <http://www.nvidia.com/cuda/>
- (2) 佐藤一輝, パートルスレンパルス, 関根優年, “FPGA アレイを用いて TFlops を目指したポアソン方程式演算回路の実装と評価,” 電子情報通信学会技術研究報告, vol. 108, no. 414, pp. 19-24, 2009.
- (3) 飯島浩晃, 佐藤一輝, 関根優年, “FPGA アレイを用いたスケーラブルな Reconfigurable HPC,” 電子情報通信学会技術研究報告, vol. 107, no. 416, pp.13-18, 2008.
- (4) 関根優年, 佐藤季花, 工藤健慈, 田向権, “hw/sw 複合体,” 第 21 回 回路とシステム軽井沢ワークショップ, pp.207-212, 2008.
- (5) K.Kudo, et. al., “Hardware Object Model and Its Application to the Image Processing,” IEICE Trans. on Fund., vol. E87-A, no.3, pp.547-558, 2004.
- (6) 青木尊之, “フル GPU による CFD アプリケーション,” 情報処理学会誌, vol.50, no.2, pp. 107-115, 2009.
- (7) 小川慧, 青木尊之, “GPU を用いた CIP 法によるプラズマ 2 流体不安定性の高速シミュレーション,” 計算工学講演会論文集, vol. 13, no. 2, pp. 837-840, 2008.

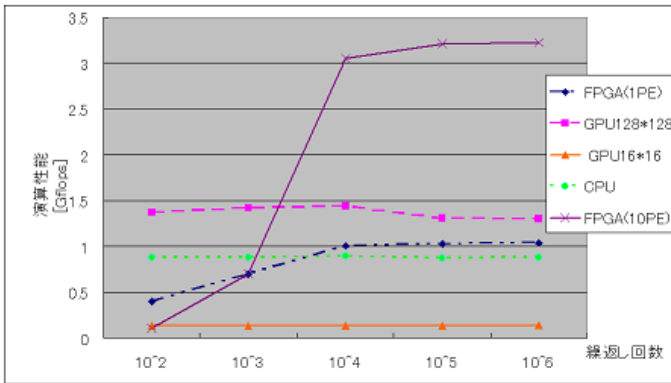


Fig. 15: FPGA , GPU , CPU の比較

Tab. 3: 演算性能の比較 (単位:[GFlops]).

計算回数 [回]	10 <sup>2</sup>	10 <sup>4</sup>	10 <sup>6</sup>
FPGA アレイ (10PE)	0.105	3.051	3.213
GPU(GeForce 9800GT)	1.370	1.442	1.299
CPU(Athlon64 X2)	0.900	0.957	0.952

演算のデータは Global Memory に保存するので，Global Memory へアクセスする回数が増えるほど，メモリ操作時間がかさんでしまう．ポアソン方程式の thread 間は完全な並列計算であるため，thread の良い分割方法を考え，高速な Shared Memory をうまく使用することができれば，演算性能は向上すると考えられる．

#### 5.4 消費電力について

先行研究において，提案 FPGA アレイシステムにおける PE が 1 個のときの，消費電力 1[W] あたりの演算性能は，131[MFlops/W] と計算されている<sup>(2)</sup>．PE が 10 個のとき，消費電力は大きく変化しないとして電力効率を計算すると，407[MFlops/W] と見積もれる．また，本実装における環境では各メーカーの仕様書によると，グラフィックカードと CPU の最大消費電力は約 65[W] である．これにより消費電力あたりの演算性能を求めた結果，FPGA の消費電力あたりの演算性能は GPU や CPU より非常に高いことが示された (Tab4).

#### 5.5 まとめ

GPU と FPGA のハードウェアの構造は大きく異なっている．GPU の内部には多数の演算ユニットがあり，CUDA においては，大規模の並列計算を適用する場合に威力を発揮しきることができる．並列計算量が多い場合には，処理 thread 数が多くなり，100 以上の演算ユニットを使用する．このようなときは，計算効率が高いといえる．一度の並列計算における演算量が多くない場合は，演算の効率は良くなく，内部の演算器の性能は，実効的に CPU より低くなってしまふ．

FPGA は対象問題にあわせて最適な演算器を利用することができる．専用の計算回路をその都度用意できるので，効率は非常に高い．また，前述の実験結果から，FPGA 中に PE を 1 個だけ実装する場合でも，使用方法によっては性能は CPU より高いと言える．さらに，10 個の PE を用意し計算を並列化した場合，演算性能が大きく向上する．

消費電力の面でも，FPGA は有利である．他のハードウェアでは，消費電力に比例して発熱が大きくなり，排熱の問題が発生する．しかし，FPGA の消費電力は低いので，提案 FPGA アレイシステムでは発熱の問題は発生しないと考えている．

しかしながら，FPGA 利用には，PC とは違い演算環境が整っておらず，利用者が独自にシステムを構築する必要があり，開発の利便性はソフトウェアに比べて良くない