GPU Spectrometer：電波天文学のための実時間広帯域信号解析システム

研究ノート

置　田　真　生*

GPU Spectrometer:
A Real-time Wide Bandwidth Signal Analysis System for Radio Astronomy

Key Words：GPU, Radio Astronomy, Signal Processing, FFT

1. はじめに

近年，電波天文学の分野では高速な信号解析の需要が高まっている．その理由は，観測できる帯域幅の広範化や観測装置の高性能化に伴って観測データ量が増大しているためである．絶え間なく観測される電波データを全て記録すると膨大なデータ量となり，全データの記録は現実的でない．そこで，観測と並行して解析を実施し，結果のみを記録する手法が用いられている．例えば，地球外知的生命体を探索するSERENDIP Vプロジェクト[1]では最大で毎秒約11.2 GBの電波データを観測する．このような大規模な電波データを実時間で解析する技術が求められている．

信号解析を高速化する手段として，近年の電波天文観測ではアクセラレータの利用が一般的である．例えば，SERENDIP V プロジェクトでは5台のFPGA（Field Programmable Gate Array）で構成されるSETI Spectrometer[2]を利用している．SETI Spectrometerは128M点の離散信号（1GBの電波データに相当）を1秒以内で解析できる．このようにFPGAを用いたシステムは高い性能を持つが，一方で高いハードウェアコストおよび開発コストという問題点があり，より大規模な電波データを解析するための拡張に必要なコストが大きい．

*Masao OKITA
1977年12月生
大阪大学大学院情報科学研究科博士後期課程修了（2006年）
現在、大阪大学　大学院情報科学研究科コンピュータサイエンス専攻　並列処理工学講座　助教　博士（情報科学）
並列処理
TEL：06-6879-4352
FAX：06-6879-4354
E-mail：okita@ist.osaka-u.ac.jp

そこで，本研究では電波天文観測のための信号解析システムをGPU（Graphics Processing Unit）を用いて実装する．GPUを利用する主な利点は3つある．まず，GPUは高い浮動小数点演算性能を持ち，1 TFlopsを超える性能をもつGPUも存在する．次に，大量生産されるため，FPGAと比較してハードウェアのコストが低い．最後に，CUDA（Compute Unified Device Architecture）[3]を用いることで，柔軟性の高いGPUプログラムの開発が可能である．

特にSERENDIP Vプロジェクトで使用されているシステムを対象とし，SETI Spectrometerの機能をGPUを用いて実現するGPU Spectrometerを開発した．4台のGPUを利用し，SETI Spectrometerが扱える以上の大規模な電波データを実時間で解析することを目指す．なお本研究では，Compute Capability1.3[4]以前に準拠するGPUを対象とし，1台のPCのPCI-Expressに複数のGPUを接続した環境を想定している．

2. GPU Spectrometer

2.1 概要

SERENDIPプロジェクトでは，地球外知的生命体が特定の周波数帯にメッセージを載せて地球に電波を発している，と考えている．SETI Spectrometerは観測した電波信号に対して周波数解析を行い，強いエネルギーを持つ周波数帯を探索する．例えば，特定の周波数帯に強いエネルギーを観測し続ければ，その周波数帯にメッセージが込められている可能性がある．

GPU Spectrometerの処理は6つのステージで構成されており，観測中は，バッファされた入力データごとに，この処理を繰り返し実行する．詳細な処理に関しては文献2)5)に記述されている．

(1)　ホストPCからGPUへのデータ転送

入力データはホストPC 上で一定時間ごとにバッファされ，バッファ単位でGPU に転送される．入力は標本化された電波信号であり，信号の1要素は16ビットの複素数（実部，虚部がそれぞれ8ビット）で表現される．たとえば，128 MHz の帯域幅を1 Hz の分解能となるように標本化すれば，入力データサイズは256 MB となる．

（2）　FFT の実行
入力データに対して，1次元FFT（Fast Fourier Transform）を計算する．計算精度を高めるために1要素を64ビットの複素数（実部，虚部がそれぞれ32ビット）に変換してから計算する．

（3）　パワースペクトルの計算
FFT の結果からパワースペクトルを計算する．FFT の結果 $x_n + y_n i$ に対して，パワースペクトルは $x_n^2 + y_n^2$ で計算できる．

（4）　強いエネルギーを持つ周波数スペクトルの検出
パワースペクトルの値が閾値を超える周波数スペクトルを検出する．

（5）　GPU からホストPC への結果転送
検出した周波数スペクトルをホストPC のメモリへ転送する．通常の観測においては，出力されるスペクトル数は少なく，入力データ長の10万分の1程度である．

（6）　結果のファイル出力
そのスペクトルの周波数帯，エネルギー値，閾値の値をファイルに出力する．

2.2 大規模な1 次元FFT の実現

GPU Spectrometer の主要な処理は，ステージ（2）の1次元FFT である．128M 点を超える大規模な1次元FFT が必要となるが，既存のGPU 向けFFT ライブラリ[6]では最大8M 点までの1次元FFT しか実行できない．信号を周波数領域で分割し，各部分に対して小規模なFFT を計算したのち結果を統合することもできるが，最終的なFFT の結果にノイズが発生する．

本研究では，ノイズの発生を抑制するために，Four-Step FFT アルゴリズム[7]をGPU 上に実装した．このアルゴリズムを用いることで，FFT の数学上の意味を変えずに，大規模なFFT をいくつかの小規模なFFT に分割できる．

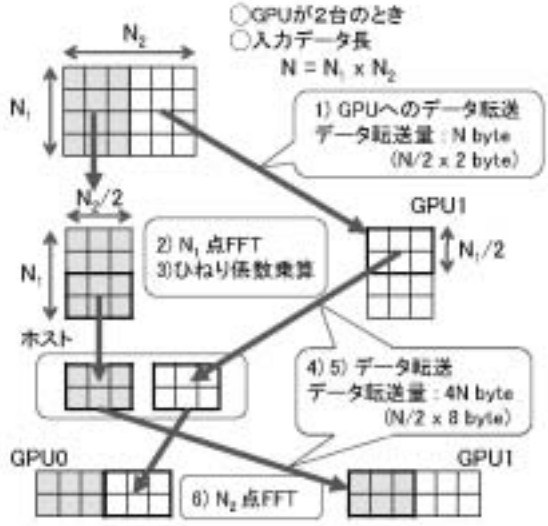

図1　複数GPU 上を用いたFour-Step FFT の単純な実装

Four-Step FFT はその名前の通り，次の4ステップで構成される．入力データ長 N が $N = N_1 \times N_2$ と因数分解できるとき，長さ N の入力データ列を N_1 行 N_2 列の行列に配置して計算する．

（1）　行列の行に沿って，N_2 個の N_1 点FFT を計算（列方向FFT）

（2）　全要素に対してひねり係数を乗算（ひねり係数は A_{jk} 要素に対して $e^{-2\pi jki/N}$）

（3）　行列の列に沿って，N_1 個の N_2 点FFT を計算（行方向FFT）

（4）　行列を転置し，N_2 行 N_1 列の行列に再配置

2.3 複数のGPU を用いた実装

単純に，複数GPU 上でFour-Step FFT アルゴリズムをそのまま適用した場合を考えると，図1のようになる．使用するGPU は2台を想定しており，図中のホストはGPU を接続しているホストPC を示す．はじめに入力データを2つのGPU に分割し，各GPU で列方向FFT を計算する．列方向FFT の終了後，次の行方向FFT のためにGPU 間でデータを交換する必要がある．

実際には，このGPU 間のデータ交換が大きなボトルネックとなる．この原因はアーキテクチャの2つの特徴にある．

まず，現行のアーキテクチャでは，GPU 間で直接データを転送できない．あるGPU A から異なるGPU B へデータを転送する際には，いったんA からホストにデータを転送し，その後ホストからB

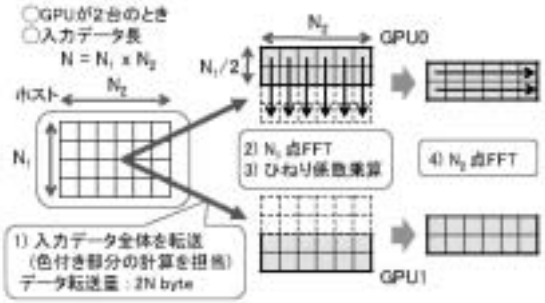

図2　複数GPU上での提案実装

へ転送する必要がある．この結果，全体のデータ転送量が増大する．例えば，図1の総データ転送量は少なくとも$10N$バイト以上である．GPUの台数をPとして一般化すると，少なくとも$2N+16N(P-1)/P$バイトのデータ転送が必要となる．

次に，多くのアーキテクチャでは，複数のGPUが1本のPCI-Expressバスを共有している．バスを共有するGPUが同時にデータを転送すると，1台のGPUが使用できる帯域幅が減少し，結果としてデータ転送に要する時間が増大する．

以降では，これらの問題点を解決するために，GPU Spectrometerで採用している提案実装について述べる．

2.3.1 GPU間データ転送の削減

ボトルネックとなるGPU間のデータ交換を削減するため，アルゴリズムを改良する．

改良したアルゴリズムの特徴は，最初に全入力データを複製して各GPUに転送することにある．GPUの台数が2台の場合の処理の流れを図2に示す．入力データをN_1行N_2列の行列として，列方向をPで分割し，各GPUはN_1/P行N_2列の計算を担当する．まず，各GPUに全入力データを転送する．次に，列方向のN_1点FFTを計算する．この際，各GPUは担当領域（N_1/P要素）の結果のみを保持し，それ以外の結果は破棄する．最後に行方向のN_2点FFTを計算する際，各GPUは必要なデータを全て保持しているので，GPU間でデータを交換せずに計算可能となる．

このアルゴリズムの総データ転送量を一般化すると$2NP$バイトである．Nによらず$P<8$であれば，提案手法のデータ転送量は単純実装（$2N+16N(P-1)/P$バイト）よりも少ない．これは，GPUの台数が少ない場合には，GPU間のデータ交換の削減による転送量の減少が，入力データの複製による転送量の増大よりも大きいことを意味する．なお，GPU Spectrometerの想定環境ではGPUの台数が8台を超えることは稀であり，このアルゴリズムが有用な環境は多い．

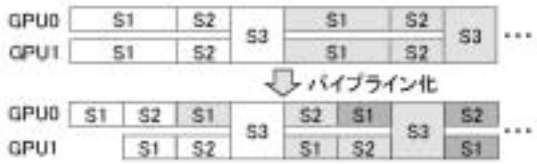

図3　2台のGPUを用いたパイプライン実行の例

2.3.2 冗長計算の隠蔽

前章のアルゴリズムでは各GPUが列方向のN_1点FFTをN_2回計算するため，GPU全体では冗長な計算が発生していることになる．この冗長計算によるオーバヘッドを隠蔽するため，GPUの機能を利用して，図2の1）データ転送と2）列方向FFTをオーバラップしている．列方向FFTよりもデータ転送に要する時間が長ければ，オーバヘッドを完全に隠蔽できる．

2.3.3 パイプライン実行

複数のGPUがバスを共有する際に発生するデータ通信の集中を避けるため，パイプライン実行を導入する．

パイプライン実行のために，信号の解析処理の過程を3つのステージに分割した．

S1：　入力データの転送（オーバラップする列方向FFTを含む）

S2：　行方向FFTの実行

S3：　周波数スペクトルの解析やファイルへの出力

なお，複数のGPUで実行する場合，S1およびS2は各GPUで独立に実行できるが，S3は分割した全FFTの結果を必要とするためGPU間の同期が必要である．

簡単のため，2台のGPUを使用する場合を考える．共有するバス上でパイプライン実行を導入しない場合の実行状況を模式化すると，図3上のようになる．この図では，2つのGPUでデータ転送（S1）を同時に実行するため，PCI-Expressバスの帯域幅が低下しS1の実行時間が増大する．

一方で，パイプライン実行を導入した場合の模式図を図3下に示す．データ転送（S1）を同時に実行

しないため，PCI-Expressバスの帯域幅が低下せず，S1の実行時間がほぼ半減することを期待できる．またGPU1がS2を実行する間は，GPU0は次の時刻のデータをあらかじめ転送する．そのため，S3以外では常にバスを使用しており，図3上と比較してバスの利用効率が向上している．

一般にGPU数に対してバス数が少ない場合は，1つのバスを共有するGPUごとにグループを作り，グループ内でパイプライン実行を導入する．

表1　実験環境

CPU	Intel(R) Xeon(R) CPU E5450
OS	Cent OS release 5.3
Memory	16 GB
GPU	NVIDIA Tesla S1070
Software	CUDA Toolkit 2.3, GCC 4.1.2
PCI-Express	PCI-Express 2.0 x8 のバスが2本

3. 性能評価

実験環境を表1に示す．この環境では最大4台のGPUを利用できる一方で，PCI-Expressのバスは2本である．したがって，4台のGPUを用いる場合は，必ずバスを共有することになる．2台のGPUを用いる場合には，バスを共有する場合と共有しない場合の両方を実験可能である．

また，評価実験に用いる入力は，人工的に作成した電波信号データである．地球外知的生命体からの信号を再現するために，作成した信号の各所にパルス波を加えている．

図1の単純実装と，提案実装の実行時間を比較した結果を図4に示す．ここで実行時間とは，ある時刻の観測データに対する解析結果出力の終わりから次の時刻の結果出力の終わりまでを計測している．

提案実装では，GPU間データ転送の削減を基本とし，他の2つの手法を適用する組合せを変えながら実行時間を計測した．凡例の括弧内のRおよびO，Pは，それぞれデータ転送の削減および冗長計算の隠蔽，パイプライン実行の適用を表す．例えば，提案実装（RO）の記述は，パイプライン実行を適用せずに，データ転送の削減および冗長計算の隠蔽を適用した場合の結果を示す．なお，バスを共有しない2台のGPUを用いる場合は，パイプライン実行を適用する必要がないため計測していない．

4台のGPUを用いる場合，3つの提案手法全てを

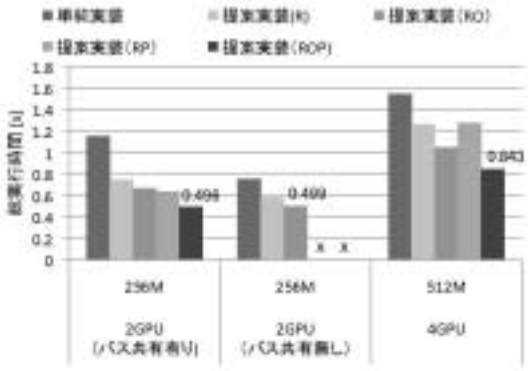

図4　手法別の総実行時間

適用することで，単純実装と比較して実行時間をおよそ半減できている．実行時間は1秒以内であり，実時間の電波解析に十分な性能が得られている．さらに，いずれかの手法が欠落すると実行時間が1秒を超えることから，3つの手法全てが有用であると言える．

また，2台のGPUを用いる場合，バスの共有に関わらず実行時間は約500ミリ秒である．これは，バスを共有する場合でもパイプライン実行を適用することで，バスを占有する場合とほぼ同等の帯域幅でデータを転送できることを示す．

なお，今回の実験で実時間で解析できる入力データ長は最大512M点である．この原因は，実験環境のPCI-EXpressバスの実効帯域幅にある．例えば1G点の入力データを4台のGPUで解析する場合には，8Gバイトのデータ転送が必要となる．実験環境の物理的な実効帯域幅は5.72 GB/sであり，データ転送時間だけで1秒間を超えるため，実時間解析は出来ない．

4. おわりに

大規模な実時間電波信号解析システムGPU Spectrometerについて述べた．GPU Spectrometerは Four-Step FFTアルゴリズムを用いて大規模なFFTを実時間で計算する．複数のGPUを用いる場合に問題となるデータ転送のボトルネックについて，解決手法を紹介した．

評価実験の結果，4台のGPUを用いて，512M点の入力データに対して実用上十分な性能が得られた．これは，FPGAを用いる既存のSETI Spectrometerと比較して，4倍の観測データを実時間で処理できることを意味する．

参　考　文　献

1) SETI Project : SERENDIP, SETI project (online), Available from 〈http://seti.berkeley.edu/serendip/〉 accessed (2010-9-27).

2) CASPER project : SETI Spectrometer, CASPER project (online), Available from 〈http://casper.berkeley.edu/wiki/SETI_Spectrometer〉 accessed (2010-9-27).

3) NVIDIA Corporation : NVIDIA CUDA Programming Guide version 2.3 (2009).

4) NVIDIA Corporation : CUDA Toolkit and SDK Version 2.3 (2009).

5) Hirofumi, K., Eric, H., Masao, O., Dan, W. and Kenichi, H.: A Multi-GPU Spectrometer System for Real-time Wide Bandwidth Radio Signal Analysis, *Proc. 8th Int'l Symposium on Parallel and Distributed Processing with Application (ISPA 2010)*, pp.594 − 604 (2010).

6) NVIDIA Corporation : NVIDIA CUDA CUFFT Library Version 2.3 (2009).

7) Bailey, D. H.: FFTs in external of hierarchical memory, *Proc. 1989 ACM/IEEE conference on Supercomputing*, pp.234 − 242 (1989).