

FPGA の高速シリアル通信を用いた クラスタコンピューティング環境の一検討

高木 大智^{1,a)} 趙 謙^{2,b)} 久我 守弘^{2,c)} 尼崎 太樹^{2,d)} 飯田 全広^{2,e)} 末吉 敏則^{2,f)}

概要: ビッグデータ処理のために、クラスタコンピューティングがデータセンタで用いられている。クラスタコンピューティングとは、複数のコンピュータをネットワークで接続し、単一のコンピュータでは得られない高い性能を得ることができるシステムである。しかしながら、ビッグデータ処理では大規模なデータ転送が発生するため通信がボトルネックになり、クラスタ全体のスループットの低下を引き起こす要因になる。そこで、本研究ではサーバに FPGA を搭載し高速シリアル通信を用いたネットワークを構築することで通信高速化を実現する。サーバ間の通信のために FPGA 上に Host PC のメモリと FPGA 間の通信および FPGA 間の高速シリアル通信を行う Communication module を設計する。2 台のサーバ間の通信速度を評価し、10Gbps のイーサネットと比較してソーティングの通信にかかる時間を 54%削減できる可能性を示す。

A Case Study of Cluster Computing Environment using High Speed Serial Communication of FPGA

DAICHI TAKAGI^{1,a)} KEN CHO^{2,b)} MORIHIRO KUGA^{2,c)} TAIKI AMAGASAKI^{2,d)}
MASAHIRO IIDA^{2,e)} TOSHINORI SUEYOSHI^{2,f)}

Abstract: Cluster computing is used for big data processing in data centers. Cluster computing is a system that connects multiple computers via a network and can obtain high performance that can not be obtained by a single computer. However, large data exchange occurs in big data processing, communication becomes a bottleneck, which is a factor that causes a decrease in the throughput of the entire cluster. Therefore, in this research, communication speed is improved by installing the FPGA in the server and constructing a network using high speed serial communication. For communication between servers, design a communication module that communicates with the memory of the Host PC and the FPGA, and performs high-speed serial communication between the FPGAs. In this paper, we evaluate the communication speed between two servers and show the possibility of reducing 54% of the sorting time compared with 10 Gbps Ethernet.

Keywords: Cluster Computing, High-speed serial communication, FPGA

1. はじめに

様々な産業や分野で、ビッグデータが活用されている。ビッグデータとは一般的に従来のデータベースシステムでは処理できないような、多様で大規模なデータを指す。例えば、センサデータ、顧客データ、ウェブサーバのログ、ソーシャルネットワークサービスのコメントなどがある。これらのデータサイズは、数十テラバイトから数ペタバイトに及ぶといわれている [1]。このような大規模なデータ

¹ 熊本大学 大学院自然科学研究科
2-39-1 Kurokami, Chuo, Kumamoto 860-8555, Japan
² 熊本大学大学院 先端科学研究部
2-39-1 Kurokami, Chuo, Kumamoto 860-8555, Japan
^{a)} takagi@arch.cs.kumamoto-u.ac.jp
^{b)} cho@arch.cs.kumamoto-u.ac.jp
^{c)} kuga@cs.kumamoto-u.ac.jp
^{d)} amagasaki@cs.kumamoto-u.ac.jp
^{e)} iida@cs.kumamoto-u.ac.jp
^{f)} sueyoshi@cs.kumamoto-u.ac.jp

の処理では高い演算能力、大規模なストレージが必要になるため、クラスタコンピューティングが用いられるようになった。

クラスタコンピューティングとは、複数のコンピュータを組み合わせてひとつの計算機システムとして扱う手法である。それぞれのコンピュータ間をネットワークで接続し、データのやり取りを行いつつデータ処理を実行していく。このクラスタ内のコンピュータをノード、相互接続されたネットワークをインターコネクトと呼ぶ。クラスタコンピューティングでは各ノードに処理を分配し、負荷を分散させることで、単一のコンピュータでは得られない高いスループットを得ることができるため、HPC (High-Performance Computing) や科学技術計算などの分野で多く用いられることが多かった。

ビッグデータが幅広い分野で活用され膨大な量のデータを処理、管理する必要性がでてきた。そこで、高い性能や膨大なストレージが必要なクラウドコンピューティングやデータセンタでクラスタコンピューティングが活用されている。現在、多くのビッグデータ処理向けクラスタコンピューティングフレームワークが開発されている。有名なフレームワークとして Apache Spark[2] があり、様々なプロジェクトで採用されている。Spark はレイテンシ重視であり、以前からあるスループット重視のフレームワーク Apache Hadoop[3] とよく比較される。データをメモリに配置することで処理速度を向上させている。しかしながら、Spark はデータ処理時にノード間で大量のデータ通信が発生するため、帯域幅が低い場合は通信がボトルネックになる可能性がある。その結果、データ処理のスループットの低下を招く恐れがある。

近年、ビッグデータ処理の高速化に対するアプローチとして、FPGA (Field-Programmable Gate Array) が注目されている。FPGA は、プログラマブル・ロジック・デバイス (PLD) の一種であり、回路構成をユーザが自由にプログラミングできる集積回路である。特定用途向けに作られた ASIC (Application Specific Integrated Circuit) とは違い、FPGA は製造後に目的や要求に合わせて回路構成を変更することが可能である。ASIC と比べて、製造工程が短く容易に回路の動作検証が可能のため、設計期間は短くなり、製造コストは安くなる。先行研究 [4] では、クラスタスケジューラ YARN を拡張し、FPGA アクセラレータに対応した Blaze を開発した。また、先行研究 [5] では Apache Hadoop[3] のデータ処理を FPGA にオフロードし、データ分析で最大 100 倍の高速化を達成した。これらの先行研究では、FPGA を用いたデータ処理演算の高速化に焦点を当てデータ通信は考慮されていない。

そこで、本研究ではクラスタコンピューティングのデータ通信の高速化について着目し、アプローチとして FPGA に搭載されている高速シリアルリンクを使用した通信を行

う。従来のクラスタコンピューティングでは通信にイーサネットが用いられるが、本研究で採用する高速シリアル通信は、イーサネットと比較し高い転送レートと低レイテンシを実現することができる。提案する通信システムは、ホスト PC と FPGA を PCI Express で接続し、ノード間は FPGA の高速シリアル通信を介してデータの転送を行う。ノード間の通信を高速化するハードウェアを設計し、その通信性能の見積りを行う。

以下、第 2 章では Apache Spark を用いたプロファイリング結果について述べ、3 章で提案システムについて述べる。4 章においてアーキテクチャの評価考察をする。最後に 6 章でまとめを述べる。

2. Apache Spark のプロファイリング

本章では、ビッグデータ処理フレームワーク Spark の概要を説明し、次に Spark のプロファイリングの結果について述べる。プロファイリングの目的は、実行時間中の演算時間、通信にかかる時間の調査である。

2.1 Apache Spark 概要

Apache Spark は 2009 年にカリフォルニア大学バークレイ校 AMPLab で開発されたビッグデータ処理向け分散処理エンジンである [2]。データをメモリに配置することで、データ処理中のディスクアクセスを減らし高速化を図っている。Spark は shuffle と呼ばれる操作を繰り返しデータの処理を実現していくが、この shuffle 時に大量のデータ通信がノード間で発生する。

2.2 プロファイリング環境

2 台のサーバによる Spark クラスタを構築し、terasort と呼ばれるソーティングのベンチマークを用いて 20GB の Key-Value 型配列のソーティングを実行した [6]。評価項目はクラスタ全体の通信速度、CPU 使用率、実行時間であり、通信帯域幅 1Gbps および 10Gbps 時の比較を行った。また、通信の調査であるため、ソーティングはすでにメモリにデータが配置されている状態からソート完了までを評価する。プロファイリングに用いた評価環境を表 1 に示す。

本稿で用いた Spark のバージョンは 2.0.0 であり、各ノードの 40GB を Spark のエグゼキュータメモリに割り当てた。図 1 および図 2 にネットワーク回線 1Gbps および 10Gbps のときの CPU 使用率と通信帯域幅の時間変化をそれぞれ示す。

図 1 および図 2 から実行時間中の演算時間を見積る。1 秒あたりの CPU 使用率を 1 秒あたりの演算時間と仮定すると演算時間とその他の処理は表 2 のようになる。また、この実行時間の内訳をグラフにしたものを図 3 に示す。

図 3 より 1Gbps および 10Gbps 両者の演算時間は、ほぼ

表 1 評価環境
server1

Model	HP Z800
CPU	Intel Xeon(R) X5680@3.33-3.6GHz
Memory	48GB
NIC	Broadcom Corporation NetXtreme
1Gbps	BCM5764M Gigabit Ethernet PCIe
NIC	Intel Corporation Ethernet
10Gbps	Controller 10-Gigabit X540-AT2

server2

Model	Super Micro model 747-14
CPU	Intel Xeon(R) X5690@3.47-3.73GHz
Memory	96GB
NIC 1Gbps	Intel82574L Gigabit Ethernet Controller
NIC	Intel Corporation Ethernet
10Gbps	Controller 10-Gigabit X540-AT2

Network Switch

1Gbps	Cisco Catalyst 2960G-24TC-L
10Gbps	Netgear XS708

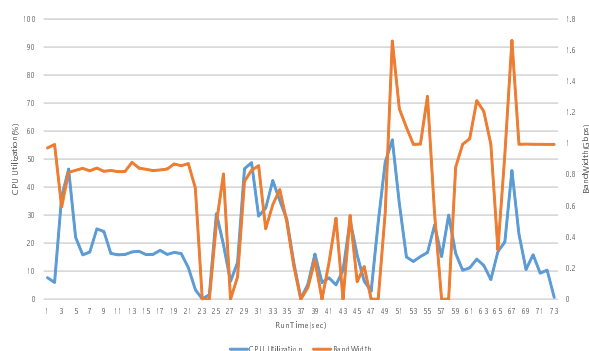


図 1 1Gbps 時の CPU 使用率と通信帯域幅

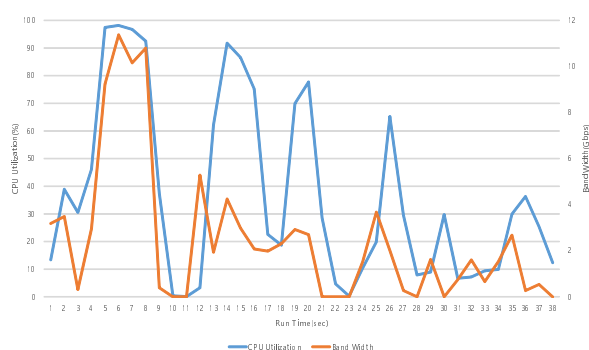


図 2 10Gbps 時の CPU 使用率と通信帯域幅

表 2 処理時間内訳の見積り

	1Gbps	10Gbps
Run Time(sec)	73.0	38.0
CPU Time(sec)	14.0	13.7
Other(sec)	59	24.3

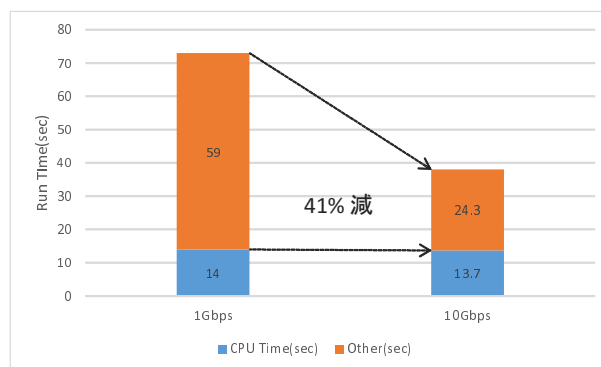


図 3 実行時間の内訳

同じであることが確認できた。これはベンチマークとして同じ terasort を行っているため、CPU の演算時間はほぼ同じ結果になったと考える。また、Other の処理時間は主に通信にかかる時間であり約 41%程度削減されている。この結果から、更なる高速ネットワークを用いることで通信にかかる時間を削減できる可能性があることがわかる。

3. 提案システム

本章では提案する通信ハードウェアについて説明する。まず、FPGA における高速シリアル通信について説明し、提案ハードウェアについて述べる。

3.1 FPAG における高速シリアル通信

本稿では GTH トランシーバを使用する。GTH トランシーバは Xilinx 製 FPGA で利用できる通信モジュールであり、最大 16.375Gbps の通信速度を実現する。このトランシーバにはトランスミッタとレシーバが存在し、データの送受信を行う。このトランシーバは通信速度やエンコード方法などの設定が可能で、多くの通信規格に対応可能である。図 4 にトランスミッタブロック図を示す。データ送信の流れは以下のとおりである。

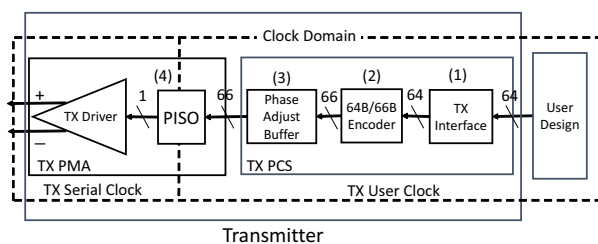


図 4 トランスミッタのブロック図

- (1) ユーザ回路から送信された 64bit のデータを受信する
- (2) 送られてきたデータに対して 66B/66B エンコードを行う。このエンコードにより、データとクロック転送を同一の通信露で行うことが可能になる
- (3) 位相調整バッファによりシリアル通信クロックとの同期を行う

表 3 ADM-PCIE-KU3 の仕様

FPGA	KCU060-2FFVA1156E
Board Format	Half-Length, low profile x16 PCIe form factor
Host I/F	PCI Express Gen3 x8
ECC-SODIMM	Two 8GB DDR3 ECC-SODIMM for memory speeds up to 1600MT/s
FLASH	1Gbit of BPI x 16 configuration flash
Interface	Two QSFP sites capable Two SATA3 capable

(4) 送信データを PISP (Parallel-IN Serial-Out) を用いてシリアルデータに変換し、差動バッファを介してデータを転送する

図 5 にレシーバのブロック図を示す。データ受信の流れは以下のとおりである。

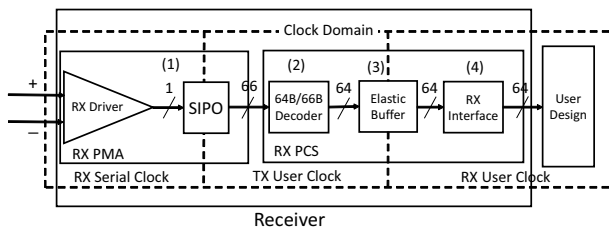


図 5 レシーバのブロック図

- (1) 差動信号として受信したデータからクロックを再生し、送信元と同期を取り、受信データを SIPO (Serial-In Parallel-Out) を用いてパラレルデータに変換する
- (2) パラレルデータを変換し、デコードを行う
- (3) Elastec バッファによりシリアル通信クロックとレシーバ用クロックの差異を吸収する
- (4) RX インタフェースを用いて、受信した 64bit データをユーザ回路に送信する

3.2 FPGA ボード

ADM-PCIE-KU3 は Alpha Data 社製の FPGA ボードであり、Xilinx 社の Kintex UltraScale の XCKU060-2FFVA1156E FPGA チップを搭載している。ADM-PCIE-KU3 はデータセンタアプリケーション向けに開発されたボードであり、Xilinx 社のデータセンタ向けアクセラレーションツール SDAccel にも対応している [7]。ADM-PCIE-KU3 の仕様を表 5 に示す [9]。

ADM-PCIE-KU3 は外部とのインタフェースとして PCIe, QSFP+ソケット, SATA などを搭載している。本稿では、通信システムに PCIe, QSFP+ケーブルを用いる。QSFP+を用いた通信の最大通信レートは 40Gbps である。

3.3 提案システム概略

提案システムの概略を図 6 に示す。図 6 は 2 つのノード

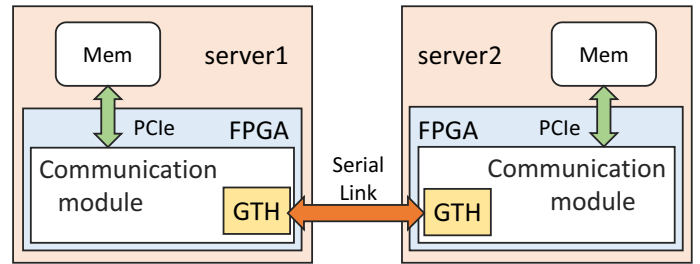


図 6 提案システムの概要

間の通信を表したものである。本システムでは FPGA 上に Host PC のメモリ-FPGA 間の通信, FPGA 間の通信を行う Communication Module を実装する。

図 6 内の Communication Module はデータ転送, データ制御を実行するモジュールである。Communication Module のブロック図を 7 に示す。Communication Module は Host PC と FPGA の通信部, 内部のデータ FIFO (First In First Out), 高速シリアルリンクから構成される。Communication Module は高速なデータ転送のために以下の手法を取り入れる。

- DMA (Direct Memory Access)
- データのストリーミング転送

DMA は Host PC と FPGA 間の通信に使用する。DMA は通信時に CPU を介さず直接メモリにアクセスしデータを転送する方法である。Communication Module に DMA Engine を組み込み PCIe を通して FPGA と Memory のデータ転送を実現する。DMA Engine から出力するデータ幅は 256bit である。本システムでは 1 つの QSFP+ポートに対して 2 つの DMA Engine を割り当てている。2 つの DMA Engine をそれぞれデータ送信用, データ受信用にすることで, データの送受信を個別に制御する。2 つの DMA Engine は BAR (Base Address Register) を通して制御を行う。また, 通信時に CPU を介さないためデータ通信時の CPU の負担を軽減することができる。

GTH トランシーバを用いてデータ転送を行う場合はインタフェースにデータをストリーミングで送る必要がある。本稿では GTH トランシーバを容易に使用可能な IP である Aurora を用いるが, 同様にデータをストリーミングで送らなければならない。Aurora はデータの入出力インタフェースに AXI-stream をサポートしているため, GTH トランシーバをそのまま用いるよりハードウェア開発及びハードウェア制御が容易になる。DMA Engine で用いるクロック周波数は 250MHz であるが, Aurora で用いるクロック周波数は 125MHz である。そのため, DMA Engine と Aurora の間に FIFO を挿入し, 転送処理に要する動作周波数の差異を吸収する。データ転送量のスループットを最大にするには, FIFO を満杯にしないような深さを設定する必要がある。適切な FIFO の深さはアプリケーションによって異なるため, 実装後に評価を行った上でユーザが

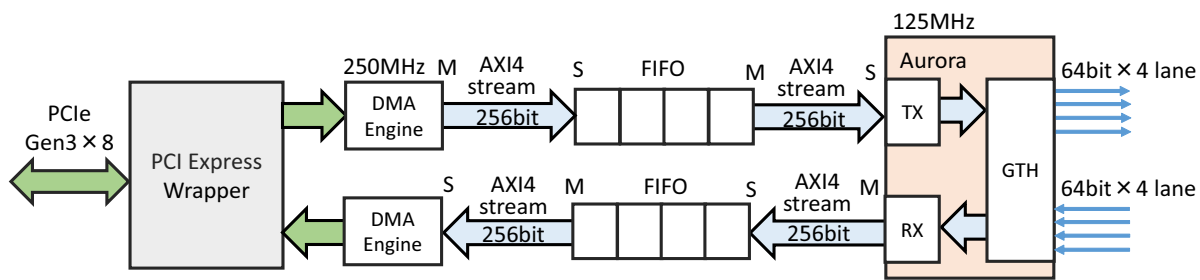


図 7 Communication Module の詳細

表 4 Host PC の仕様

CPU	Intel Core i7-7800X
Motherboard	ASUS TUF X299 MARK 2
Memory	128GB
OS	Ubuntu 16.04 LTS
Host I/F	PCI Express Gen3 x8

表 5 ADM-PCIE-KU3 の仕様

FPGA	Xilinx 社製 KCU060-2FFVA1156E
DMA Clock	250 MHz
GTH	Line Rate 8Gbps × 4 lanes, total 32Gbps
	clock 125MHz

指定する必要がある。

Aurora には 8B/10B, 64B/66B の 2 種類のエンコード方式があるが、64B/66B の方がスループットが高いためこちらを採用する。また、64B/66B エンコーダを用いて、GTH のレーン数を 4 にすると FIFO からのデータ幅である 256bit を変更することなく Aurora の入出力として接続することができる。

4. 評価

4.1 評価目的

3 章で提案した通信システムの評価について述べる。本稿では、複数のノード間での通信ではなく、1 台の Host PC を用いて通信露をループバックさせたときの性能を見積り、および単方向のデータ転送のスループットを算出する。その結果から 2 つのノード間の通信速度を見積る。評価に用いた Host PC の仕様を表 4 に示す。また、評価ボード ADM-PCIE-KU3 の仕様を表 5 に示す。ハードウェアの開発には Xilinx Vivado 2017.1 を使用した。

4.2 評価と考察

図 8 に本稿で評価するデータパスを示す。本稿では、データパスを 3 つに分けて評価する。

- (1) Host PC のメモリ-FIFO 間
- (2) FIFO のデータの送受信
- (3) Aurora を用いた高速シリアル通信による送受信

仮定する転送データのサイズを Spark のデフォルトのブ

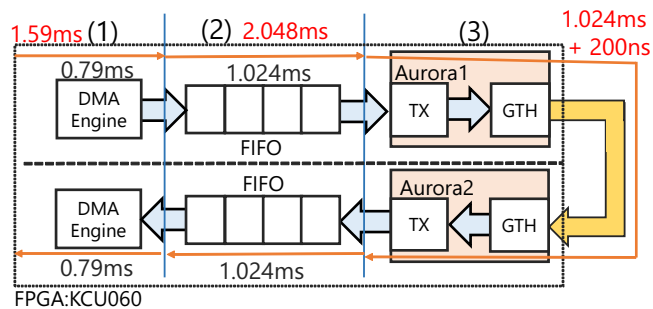


図 8 評価のためのブロック図

表 6 データパスのレイテンシ

Data Path1	Data Path2	Data Path3	ALL
1.59 ms	2.048 ms	1.024 ms	4.662 ms

ロックサイズ 4MB にする。表 6 にデータパス 2 およびデータパス 3 のレイテンシを示す。図 8 中の (1) は Host PC のメモリ-DMA Engine までのデータパスであり、Alpha data 社製の DMA example デザインを用いてデータ通信のループバックを行い測定した。その結果、Host-FPGA 間の通信スループットは 5.133GB/s であった。4MB のデータ転送を行うには送受信で 2 度 DMA 転送を行うため 1.59ms である。

図 8 中の (2) は FIFO を介してデータの送受信に対するレイテンシである。FIFO は動作するまでにデータシートから書き込み読み込み 1clock の時間を要するため、1.024ms かかる。送受信で 2.048ms のレイテンシになる。

図 8 中の (3) は Aurora を使用した高速シリアル通信処理に要するレイテンシであり、FPGA 外部の通信ケーブルのレイテンシも含む。Xilinx 社製の IP である Integrated logic Analyzer (ILA) を用いてレイテンシを測定したところ、送信から受信まで 25clock かかっていた。シリアル通信は 125MHz で動作しているためレイテンシは 200ns である。4MB のデータの送受信に 1.024ms かかる。

以上から 4MB のデータの送受信に総計 4.662ms かかる。この結果を基に Spark の 20GB のソートがどれほど高速化できるかを考察する。図 9 に実行時間を見積り示す。CPU の演算時間はイーサネットを用いた通信と変わらないと考え 13.7s とした。図 2 より 1 秒あたりの通信速度が

らソーティング時の総データ通信量は 12.25GB であると見積る。よって、高速シリアル通信を使用したときの通信にかかる時間は 13.6s であると見積れる。通信帯域幅 10Gbps の場合に対して 54%通信時間を削減できると予測できる。

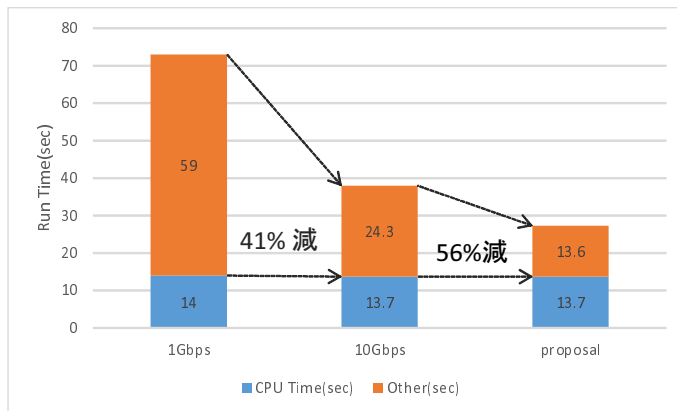


図 9 提案手法の実行時間見積り

5. まとめ

FPGA の高速シリアル通信を用いた通信システムを提案した。提案システムでは FPGA 上に Host PC のメモリ-FPGA 間の通信および FPGA 間の高速シリアル通信を行う Communication module を設計した。本稿では 2 台のサーバ間の通信を設計し、その性能を評価した。その結果、10Gbps のイーサネットと比較し 20GB のソーティングの通信時間を 54%の実行時間の削減できる可能性を示した。今後は Communication module の実装を行い、実機上での評価を行うとともに、Spark 上での本システムを利用できるように改変を行う予定である。

参考文献

- [1] 総務省 平成 24 年版 情報通信白書, <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h24/html/nc121410.html> [2017-7-17 参照].
- [2] Apache SparkTM - Lightning-Fast Cluster Computing, <https://spark.apache.org/>. [2016-5 参照].
- [3] Welcome to ApacheTM Hadoop!, <http://hadoop.apache.org/>. [2016-5 参照].
- [4] Muhuan Huang, Di Wu, Cody Hao Yu, Zhenman Fang, Matteo Interlandi, Tyson Condie, Jason Cong, “Programming and Runtime Support to Blaze FPGA Accelerator Deployment at Datacenter Scale,” SoCC ’16 Proceedings of the Seventh ACM Symposium on Cloud Computing, pp. 456-469, 2016.
- [5] 日立製作所, ニュースリリース : 2017 年 11 月 14 日, <http://www.hitachi.co.jp/New/cnews/month/2017/11/1114.html>. [2017-12 参照].
- [6] Github-ehiggs/spark-terasort:Spark Terasort, <https://github.com/ehiggs/spark-terasort>. [2016-5 参照].
- [7] Xilinx, SDAccel 開発環境, <https://japan.xilinx.com/products/design-tools/software-zone/sdaccel.html>. [2017-11 参照].

- [8] ADM-PCIE-KU Product Page:Functionality and Applications, <https://www.alpha-data.com/dcp/products.php?product=adm-pcie-ku3>. [2017-11 参照].
- [9] ADM-PCIE-KU3 User Manual, <https://www.alpha-data.com/pdfs/adm-pcie-ku3>