社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

ネットワーク結合型マルチFPGAボードを用いた集約演算クエリ処理

川原　尚人†　吉見　真聡††　策力　木格††　吉永　努††

† 電気通信大学大学院情報システム学研究科
†† 電気通信大学大学院情報理工学研究科
〒 182–8585 東京都調布市調布ケ丘 1–5–1
E-mail: †nkawahara@comp.is.uec.ac.jp, ††{yoshimi,clmg,yosinaga}@is.uec.ac.jp

あらまし　大量のデータが常時収集され集積されるようになった現在のデータセンタにおける重要な課題として，省エネルギーで高速な計算機システムの研究開発が挙げられる．ビッグデータの分析は，演算あたりのデータ量が多く，主記憶へのデータ転送がボトルネックになりやすい．そのため，メニーコアプロセッサなど従来型のアクセラレータを用いた高速計算の効果を挙げにくい状況にある．いくつかの研究組織では，プロセッサでの処理を行う前に，ネットワークやストレージのインタフェイスに接続したハードウェアで演算を行うハードウェアを開発している．著者らの研究グループでも，ネットワークとストレージを密に接続するFPGAボードを用いた，Interconnected-FPGAと呼ぶ計算機システムを開発中している．Interconnected-FPGAは，各FPGAボードに接続されたフラッシュストレージと，ボード間で構成するリングネットワークを活用し，データの読み込みやノード間データ転送を効率化するとともに，FPGAに構成する専用ハードウェアでデータ経路上での演算を実現する計算機システムである．本研究報告では，Interconnected-FPGAの応用例として，関係データベースの集約演算を行うハードウェアの実装と評価を示し，そのデモンストレーションを行う．

キーワード　密結合，FPGA，並列処理，クエリ処理，集約演算

Processing Aggregation Queries using Interconnected Multiple FPGA Boards

Naoto KAWAHARA†, Masato YOSHIMI††, Celimuge WU††, and Tsutomu YOSHINAGA††

†
†† The University of Electro-Communications　Chofugaoka 1–5–1, Chofu-shi, Tokyo, 182–8585 Japan
E-mail: †nkawahara@comp.is.uec.ac.jp, ††{yoshimi,clmg,yosinaga}@is.uec.ac.jp

Abstract　A huge amount of data is being generated and accumulated in datacenters, which leads to an important increase in the energy consumption required to analyze this data. Several research groups reported dedicated hardware which executes pre-process of application at the input of the network or the output of the storage. Our research group has also been developing a PC-cluster-based computer system called Interconnected-FPGA, which uses multiple FPGA boards which equips interfaces with flash storage and optical network. The interconnected-FPGA improves efficiency of data exchange by direct data transmission and performance by offloading pre-process to data-path in the FPGA. In this technical report, we demonstrate an availability of interconnected-FPGA by an aggregate operation which is used in relational database.

Key words　Tightly Coupled, FPGA, query processing, aggregation

1. はじめに

多様なWebサービスの普及とセンサ技術の発達にともない，データセンタに収集され，蓄積されるデータ量は急速な増大を続けている．そのような莫大なデータを解析し，重要データの抽出，意思決定，危機管理，サービスの改善など，データの量を情報の質に変換する，ビッグデータ分析が広く行われるようになった[1].

MapReduce[2]をはじめとする多数の計算機に対して良いスケーラビリティを示すプログラミングフレームワークは，ビッ

グデータ分析の普及に大きく寄与している．その一方で，蓄積されるデータの増加速度とアプリケーションの拡大は計算機の性能向上速度を超えているため，現実的な計算時間で解析処理を行うために，計算機システムの規模は拡大を続けている．

計算機システムの規模拡大による消費電力増大への対応のひとつとして，GPU などのメニーコアプロセッサによる高効率な計算機システムが利用されている．またその一方で，ビッグデータ分析は演算あたりのデータ量が多く，主記憶へのデータ転送がボトルネックになりやすい．そこで，いくつかの研究機関では，ネットワークやストレージに接続した FPGA で専用ハードウェアが演算の一部を行う仕組みを開発している．この専用ハードウェアは，通過するデータの抽出や分類を行い，データの供給源から主記憶に転送されるデータを削減してプロセッサの負荷を下げる機能を担う．これらは Near Data Processing (NDP) と呼ばれており，いくつかの研究では NDP の導入でアプリケーションの実行時間を数十倍高速化したと報告している．

著者らの研究グループは，ネットワークとストレージを密に接続する FPGA ボードを搭載した PC を複数集積して構築した PC-Cluster を，ビッグデータ解析の高速計算基盤とする Interconnected-FPGA と呼ぶ計算機システムの研究開発に取り組んでいる．

本研究報告では，Interconnected-FPGA を用いたデータ解析高速化の応用例として，関係データベースの集約演算を行うハードウェアを実装した計算機システムのデモンストレーションを行う．ストレージから読み出したテーブルを主記憶に格納する前に，データ経路上に存在する FPGA に実装した専用ハードウェアで集約演算を行い，アプリケーションの前処理をオフロードする．また，FPGA ボード間のネットワークを介してストレージデータの直接通信を行う機構を使用して，テーブルのパーティショニングを行い，より高速化する仕組みが組み込まれている．これらの高速化の仕組みにより，ベンチマーク TPC-H の Query1 を対象とした性能測定の結果，データベースソフトウェアと比べて単一 FPGA ボードで約 7 倍の性能向上を確認した．また，マルチ FPGA 環境においても，良いスケーラビリティが得られることを確認した．

2. 関 連 研 究

著者らの研究グループを含め，複数の研究機関がビッグデータ解析に FPGA による専用ハードウェアを導入して高速化を図る研究に取り組んでいる．

IBM 社は Netezza と呼ばれる，FPGA をストレージとプロセッサの間に組み込んだブレードサーバを開発している [3]. Netezza はストレージからのデータ読み出し時に FPGA で不要なデータを削除し必要なデータだけを抽出することで，高速データベース処理を実現している．また，計算生物学における応用例なども報告されている [4].

BlueDBM [5] は，FPGA ボード上のストレージと DRAM を高速ネットワークで接続することで，巨大な主記憶を持つ高価な計算機と同程度の性能を性能が得られることを報告している．

Microsoft 社の Catapult [6] は，検索システムで用いる PageRank を高速化するために導入された，相互接続された FPGA ボードを含む計算機システムである．

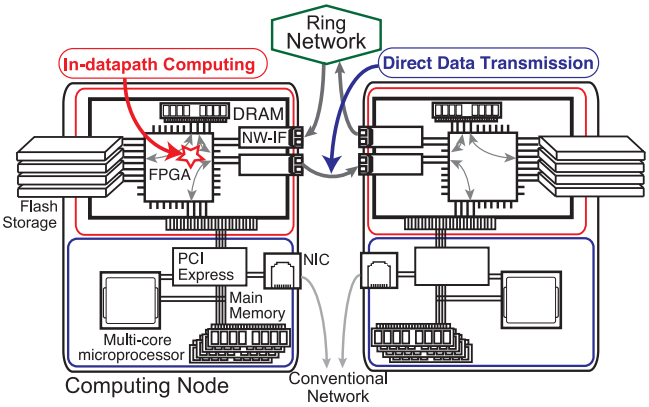

図 1: Interconnected-FPGA の概要

著者らの研究グループは，フラッシュストレージを接続可能な FPGA ボードを高速ネットワークで接続して並列分散処理を行う，Interconnected-FPGA と呼ぶ大規模データ解析の高速計算機システムの研究開発に取り組んでいる [7] [8]. Interconnected-FPGA は，マルチ FPGA 環境のフラッシュストレージに分散されたデータを，FPGA ボード間の直接通信 (Direct Data Transmission) と，FPGA 内部を通過するデータストリームに対する専用ハードウェアによる演算 (In-datapath Computing) を組み合わせて高速に処理する仕組みである．関係データベースを対象アプリケーションとして，マルチ FPGA 環境で取り組んでいることに，他の研究とは異なる特色がある．

3. Interconnected-FPGA

3.1 概　　要

著者らは，IO バウンドな計算の高速化を目的に，ストレージを持つ FPGA ボードをネットワークで接続して並列動作させる Interconnected-FPGA と呼ぶ機構を提案している．

図 1 に，計算機 2 ノードで構成する Interconnected-FPGA の例を示す．各ノードは，フラッシュストレージとネットワークのインタフェイスを持つ FPGA ボードが接続されており，相互に接続されている．Interconnected-FPGA は，計算がオフロードされる従来のメニーコアプロセッサや FPGA と比べ，2 つの利点が得られる．

Indatapath Computing ストレージやネットワークからホスト PC の主記憶にデータを読み込む途中で，FPGA に構成したハードウェアで計算の一部を実行できる．計算に必要なデータの抽出や加工など，従来は主記憶に読み込まれたデータに対して行われる前処理部分をオフロードすることで，プロセッサの負荷を下げ，より複雑な後処理に注力させることができる．

Direct Data Transmission FPGA ボード間のネットワークを使用して，ホスト PC の主記憶を介さずにストレージのデータを転送できる．OS のシステムコールなど，プロセッサの計算に割り込んで主記憶へアクセスするような処理の回数が

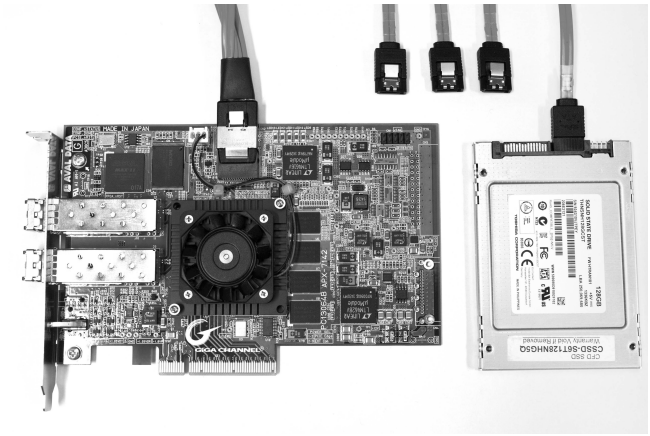

図 2: 第 2 世代 FPGA ボード APX-7142(改)

削減される．

Interconnected-FPGA の実装として，著者らは DSPE と呼ばれるハードウェア構造を提案している．

実験環境として特定の FPGA ボードといくつかのプロプライエタリなハードウェアモジュールを使用しているものの，Interconnected-FPGA は，ストレージとネットワークインタフェイスを持つ他の FPGA ボードへの移植することができる．

3.2 FPGA ボード APX-7142 改

著者らの研究グループでは，Interconnected-FPGA の実証実験を行うための FPGA ボードを開発している．表 1 に，著者らの研究グループが開発している FPGA ボードの仕様を示す．現在，開発に取り組んでいる APX-7142 改の外観を図 2 に示す．

Interconnected-FPGA では，以下の 4 つの主要な外部インタフェイスを持つ FPGA ボードが使用できる．

PCI-Express ホスト PC の主記憶と高速にデータ通信可能なホストインタフェイス．

DRAM データの一時的な保存や，ネットワークやストレージアクセスの際のバッファとして使用される DRAM.

(光) ネットワーク 高速な FPGA ボード間通信を実現するネットワークインタフェイス．Direct Data Transmission を可能にするために必要となる．

フラッシュストレージ データが格納されるストレージデバイス．読み書きするデータに対し，In-datapath Computing で演算が行われる．

著者らは，これらのインタフェイスを持つ FPGA ボードとして，アバールデータ社の APX-880A，APX-7142 を使用している [9]．当初，APX-880A を実験環境として利用して Interconnected-FPGA の研究開発に取り組んでいた．新世代の FPGA や SSD など新しいデバイスの利用や，転送単位の柔軟性の向上を目的に，実験対象を APX-7142 に移行した．APX-7142 ボードは，28Gbps の速度でボード間通信可能な光ネットワークインタフェイスを持つが FPGA ボードである．著者らは，APX-7142 ボードに実装されている SAS ポートを，フラッシュストレージとして SATA 接続 SSD を 4 ポート接続できるように改造して使用している．

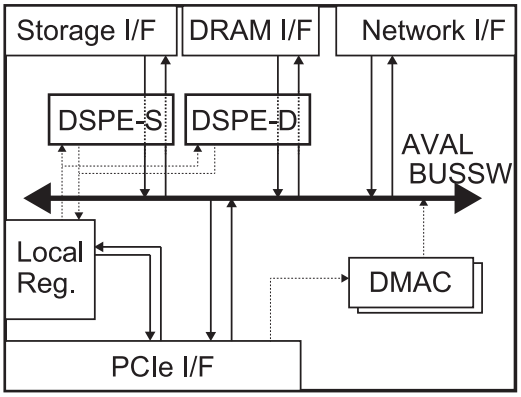

図 3: Interconnected-FPGA のハードウェア構造

3.3 DSPE : Data Stream Processing Engine

図 3 に, Inteconnected-FPGA において FPGA 上に構成されるハードウェア構造を示す．図 3 に示すように，FPGA ボードのインタフェイスは, AVAL-BUSSW と呼ばれる FPGA 内部に構成されるバスを介して相互接続されている．AVAL-BUSSW はアバールデータ社独自のバス規格であるが，他の FPGA ベンダによるバスと同様，ホスト PC から発行される DMA(Direct Memory Access) 命令によって，インタフェイス間のデータ転送を指示できる．

フラッシュストレージに格納されたデータは，ホスト PC から発行される DMA 命令に応じて読み出され，バスを介して光ネットワークへと直接送出される．受信側ホストも DMA 命令を発行し，光ネットワークから入力されるデータをフラッシュストレージに格納する．このデータ転送は送受信ホスト双方で主記憶を介さずに行われる．主記憶に転送データがバッファされることがないため，高速かつ低負荷なデータ通信 (Direct Data Transmission) が可能である．

In-datapath Computing を実現する専用ハードウェアを組み込む仕組みとして，著者らは DSPE (Data Stream Processing Engine) と呼ぶ機構を提案している [7]．図 3 に示すように，

表 1: FPGA ボード APX-880A, APX-7142 の仕様

Product	APX-880A(旧)	APX-7142(新)
FPGA Device	Stratix IV GX EP4SGX230KF40C2 runs at 125 MHz	Stratix V GX 5SGXMA3K1F40C2N runs at 125 MHz
DRAM (DDR3)	533 MHz, 512 MB	800 MHz, 2.0 GB
Flash Storage	18 SD Cards×1 or 2 1.5 GB/s×2ch two SDs for parity	SAS connector extends 4 SATA ports scratch build in this paper
unit to transfer	128[MB]	4[KB]
Network	Proprietary GiGA CHANNEL Optical token ring network unit to transfer 16 [KB]	
	8.5 Gbps×2ch	14 Gbps ×2ch
PCIe I/F	2.0 Gen2×4 Lane	2.0 Gen2×8 Lane
Internal Bus	Proprietary AVAL-bus	
	128 bits-width	256 bits-width

DSPE は各インタフェイスとバスの間に組み込まれ，バスに入出力されるデータに対して演算を行うハードウェアモジュールである．

3.4 DSPE を活用したアプリケーション

著者らは，ストレージインタフェイスの出力に接続される DSPE-S (Storage)，および，DRAM インタフェイスの入出力に接続される DSPE-D の 2 種類の DSPE を提案している．DSPE 内部は，アプリケーションに応じた専用ハードウェアが実装され，DSPE のデータ転送経路上を通過するデータストリームに対して演算を実行する．DSPE での処理は，PCI-Express 経由で書き換え可能なレジスタアレイ (Local Registers) を通じて制御される．

著者らは，DSPE を組み込んだ In-datapath Computing の有用性を示すため，2 種類の専用ハードウェアを実装し，評価している．

Simple Word Counting : ストレージに格納したデータ中に，指定した単語の出現数を数えるアプリケーション．さらにアプリケーションを単純化し，32 ビットのハッシュ値照合としてハードウェアを実装して評価した [7].

Group-by Aggregation : ストレージに格納された Key-Value テーブルから，Key 毎に値を集約して統計値を得る，関係データベースのオペレータのひとつ．その一例として蓄積型データベースのベンチマーク TPC-H の Query 1 を実行する専用ハードウェアを実装して評価した [8].

これら 2 種類のアプリケーションともに，ストレージからのデータ読み出し経路上での計算により，マイクロプロセッサでのソフトウェアに比べて 10 倍以上の高速化を確認している．

4. デモンストレーション

4.1 概　　要

本研究報告では，複数 FPGA ボードを用いた集約処理クエリ演算のデモンストレーションを行う．この集約処理クエリ演算は，先行研究である Group-by Aggregation 処理 [8] を拡張し，以下の 2 点を可能にしたハードウェアで実現する．

入力クエリにしたがう集約演算 先行研究 [8] では TPC-H Query 1 のみに特化した専用ハードウェアを実装したが，一定規模のクエリまでは任意の数の Aggretation オペレータを実行できるハードウェアを実装する．

複数の FPGA ボードを用いた並列分散処理 先行研究 [7] [8] では，DSPE-S のみで計算が完了するハードウェア構造を持っており，複数ノードで協調して高速実行する仕組みになっていなかった．そこで，複数の FPGA ボードが並列動作する分散処理構造を追加した．

4.2 集約処理クエリ演算

集約処理クエリ演算を，関係データベースのベンチマーク TPC-H の Query 1 として知られる図 4 に示すクエリを元に説明する．Query 1 は，lineitem と呼ばれるテーブルを対象に，(a) の条件にマッチしたタプルについて，(3)-(4) の算術演算を行った上で，(X) と (Y) の値ごとに (1)-(6) の値の総和，平均，出現数を集約して出力するクエリである．

```
SELECT
    l_returnflag, l_linestatus,
    sum(l_quantity) as sum_qty,
    sum(l_extendedprice) as sum_base_price,
    sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
    sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge,
    avg(l_quantity) as avg_qty,
    avg(l_extendedprice) as avg_price,
    avg(l_discount) as avg_disc,
    count(*) as count_order
FROM
    lineitem
WHERE
    l_shipdate <= date '1998-12-01' - interval '90' day
GROUP BY
    l_returnflag,
    l_linestatus
ORDER BY
    l_returnflag,  l_linestatus;
```

図 4: TPC-H Query 1

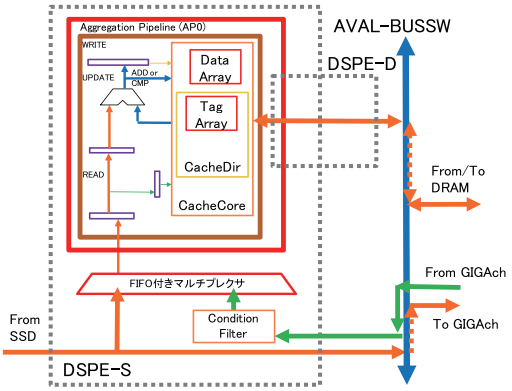

図 5: Aggregation-PIPE の構造

先行研究 [8] では，(X) と (Y) の取り得る値が合わせて最大 6 種類であることを利用して，(3)-(4) の算術演算を組み込んだ Query 1 専用ハードウェアを DSPE-S として実装し，フラッシュストレージから DRAM にテーブルが読み込まれる際に集約演算を行った．

4.3 Aggregation PIPE

図 5 に，実装した集約処理クエリ演算ハードウェアの構造を示す．図 5 は，4.1 節で述べた 2 つの拡張が行われたハードウェアモジュールである．

集約演算は，Aggregation PIPE で行われる．フラッシュストレージから読み出された lineitem テーブルは，DSPE-S 内で 2 つのデータストリームに複製される．一方の (第 1) ストリームは Aggregation PIPE 方向に向かい，集約演算が行われる．また，もう一方の (第 2) ストリームは，ネットワークを介して他の FPGA ボードに配送される．

Appregation PIPE に入力される前に，データストリームは FIFO 付きのマルチプレクサに入力され，集約するための key((X) および (Y) の組み合わせ) と value ((1)-(6) の値) が選択され，集約オペコードとして FIFO に入力される．またそのとき，集約オペレータ (Op.) が同時に生成される．これらの，タプル中の key や value の位置やオペレータの種類は，Local Register Array に書き込まれた設定値で決定される．

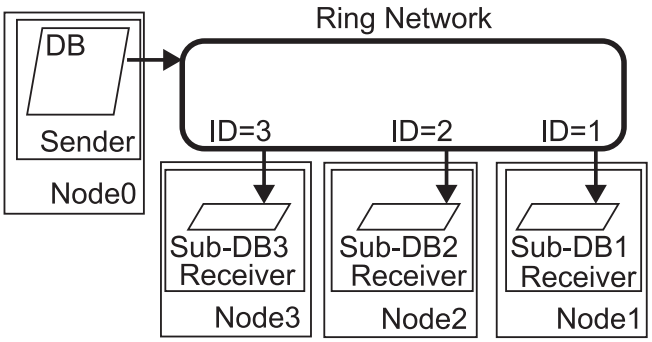

図 6: 集約演算クエリのマルチ FPGA への拡張

Aggregation PIPE は IBEX [10] と同様の，3 段のパイプラインを用いて，集約演算を行うハードウェアモジュールである．

第 1 ステージで，FIFO から取り出された集約オペレータと集約オペコードを用いて，key に基づいて集約演算の中間データの読み出しが行われる．中間データは Cache Core モジュール内の BlockRAM にキャッシュされており，キャッシュヒットした場合は 1 クロックサイクルで読み込まれるが，キャッシュミスした場合は DRAM からデータを読み込むまで待たされる．

第 2 ステージでは，集約オペレータにしたがって集約オペコードと中間データの演算を行う．集約オペレータは，MAX，MIN，SUM，Count の 4 種類がある．これらの演算は 1 クロックサイクルで実行される．平均を求めるオペレータ AVG は除算器のリソース量や時間の関係から，計算終了後に SUM/Count で求めるものとした．

第 3 ステージでは，演算結果を key の新たな中間データとして書き込みを行う．key の中間データは第 1 ステージで BRAM に格納されており，必ずキャッシュヒットするので，1 クロックサイクルで書き込みが終了する．

テーブルのタプルすべてをフラッシュストレージから読み込み終わったとき，Aggregation-PIPE と DRAM 内には，集約演算の結果が格納されていることになる．

この Aggregation-PIPE の構造により，Local Register Array に設定する値を PCI-Express 経由で書き換えることで，先行研究には無かった，様々な集約演算クエリに対応することが可能となる．また，DRAM が使用可能になったことで，key の種類の制限を大きく緩和できる．

4.4 マルチ FPGA ボードによる集約処理クエリ演算

Aggregation-PIPE によってストレージから読み込まれたテーブルの集約演算が可能となったが，いくつかの制限が存在する．Cache Core モジュール内の BRAM サイズが 16KB であるため，key の種類が多いクエリの場合，キャッシュミスが頻発することになる．

そのため，複数の FPGA ボードを用いて並列計算の効果を挙げる方法を提案する．集約演算を行うタプルを，図 5 下部に示す Condition Filter モジュールで選択する方法である．図 6 に，マルチ FPGA 環境での並列集約演算の概要を示す．

まず，Condition Filter モジュールで，Aggregation PIPE に入力するタプルを，FPGA ボードの ID とマッチする key を持つタプルのみに制限する．key の値は FPGA ボードの数より通常多いので，key の下位数ビットを分割のための判定ビットとする．

マッチしなかったタプルは，第 2 ストリームとしてネットワーク (GIGAch) を介して他の FPGA ボードに配送される．各 FPGA ボードは，ネットワークから入力される他ボードのデータストリームを受け付け，FPGA ボードの ID がマッチするタプルを Condition Filter モジュールで抽出し，Aggregation-PIPE に入力する．すなわち，Condition Filter モジュールで集約演算を行うテーブルをパーティショニングすることで，キャッシュサイズをボードの数だけ増加させ，キャッシュヒットの割合を上げる方法である．

5. 評　　価

4. 章で述べたハードウェアを実装し，表 2 で示した計算機環境でマルチ FPGA による集約演算の性能評価を行った．各計算機ノードのソフトウェア環境およびハードウェアの実装環境を表 3 に示す．

計算機ノード 1 台での MySQL での TPC-H Query1 の実行時間，1 枚の場合と 4 枚の場合に FPGA ボードを用いた集約演算の計算時間を，図 7 に示す．TPC-H Query1 の対象テーブルは，1GB と 4GB の 2 種類のデータセットで計測を行った．また，MySQL での計算時間と比べて，FPGA ボードでの集約演算による速度向上率を図 8 に示す．

Aggregation-PIPE を含めた FPGA のリソース使用量を表 4 に示す．Total は Agg-PIPE を含むインタフェイスやバス等で使用するリソース使用量，Agg-PIPE は集約演算を組み込んだ

表 2: 計算機環境

No. of Node	4	
Product Name	DELL Precision Workstation T5610 D01T	
CPU	Intel Xeon E5-2630	2.60GHz (6C12T)
Memory	four 4 GB DDR3	16.0 GB
Network	Broadcom NetXtreme on board Ethernet	10Gbps 1Gbps
SSD	Crucial CT480M500SSD1 ×2(RAID0)	960GB
HDD	Seagate ST2000DM001	2.0TB

表 3: ソフトウェア環境

OS	CentOS 7.2 kernel-3.10.0.327.22.2.el7.x86_64
SW Compiler	gcc-4.4.7 boost-1.57.0 OpenMPI-1.8.1 -m64 -O3 -fopenmp
HW CAD	Altera Quartus II (VHDL) 13.1 for APX-7142
Database	MySQL 14.14

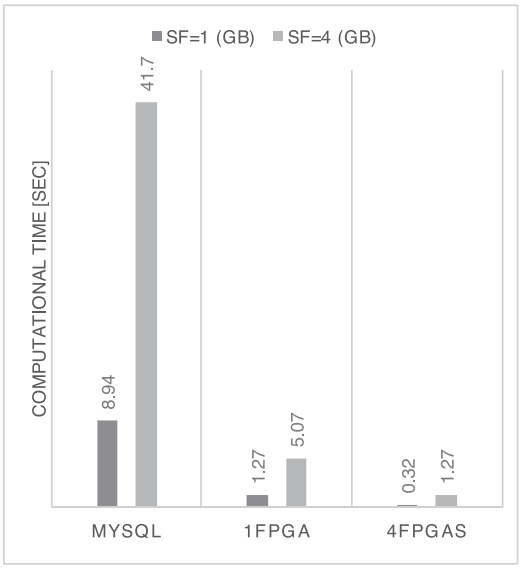

図 7: 計算時間の比較

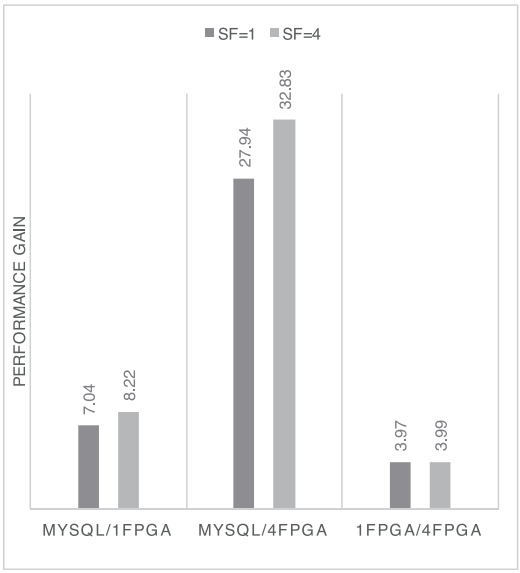

図 8: 性能向上率

DSPE-S のリソース使用量である．Agg-PIPE のリソース使用量は FPGA 全体に対して 1%程度と十分に小さく，FPGA の資源に余裕があるため，複数の Agg-PIPE を実装してキャッシュヒット率を上げ，さらなる高速化を図ることができる．

FPGA を使用した場合，ホスト PC は DMA 転送の指示と結果の読み込みだけを行えばよく，集約演算が FPGA 内部で完結するため，MySQL のようにマイクロプロセッサで集約演算を行う場合に比べて，約 7 倍の性能向上が確認された．

また，4 枚の FPGA を用いた並列集約演算の場合，Query1 では全て (6 種類) の key がキャッシュヒットするため，性能がほぼ 4 倍になることが確認された．

表 4: ロジック使用量

	Capacity	Total	Agg-PIPE
Logic [%] Utilization	100	50.3	—
Combinational ALUTs	128300	64558	1531
Dedicated Logic Reg.	128300	92858	789
Block RAM [Kb]	19140	8875	128

6. まとめと今後の課題

本研究報告では，ビッグデータ解析を高速化する計算機システムとして，ネットワークとストレージを密に結合する FPGA ボードを用いた Interconnected-FPGA の応用例のデモンストレーションを行った．関係データベースの集約演算を行うハードウェアと，それをマルチ FPGA 環境に拡張する実装を開発し，ベンチマーククエリでソフトウェア実行と比べて 7 倍程度の高速化が確認された．また，マルチ FPGA 環境においても，FPGA ボードの数にスケールする結果が得られたことを確認した．

今後の発展として，Query1 のような少数の key の集約だけでなく，キャッシュミスが生じるような多種の key を対象とするクエリでの集約演算を効率よく計算するハードウェアの実装と評価を行う予定である．

謝辞 本研究の一部は JSPS 科研費 26330061, 63003088 の助成を受けたものです．

文　献

[1] C.P. Chen and C.-Y. Zhang, “Data-intensive applications, challenges, techniques and technologies: A survey on big data,” Information Sciences, vol.275, pp.314–347, 2014.

[2] J. Dean and S. Ghemawat, “Mapreduce: Simplified data processing on large clusters,” Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation, pp.137–149, 2004.

[3] G.S. Davidson, et al., “Data-centric computing with the netezza architecture,” SANDIA REPORT, pp.1–24, Apr. 2006.

[4] J.A. Delmerico, et al., “Comparing the performance of clusters, hadoop, and active disks on microarray correlation computations,” HiPC, pp.378–387, 2009.

[5] S.-W. Jun, et al., “Bluedbm: An appliance for big data analytics,” ISCA, pp.1–13, 2015.

[6] A. Putnam, et al., “A reconfigurable fabric for accelerating large-scale datacenter services,” 41st Annual International Symposium on Computer Architecture (ISCA), pp.13–24, 2014.

[7] M. Yoshimi, R. Kudo, Y. Oge, Y. Terada, H. Irie, and T. Yoshinaga, “An FPGA-based Tightly Coupled Accelerator for Data-intensive Applications,” IEEE 8th International Symposium on Embedded Multicore/Many-core SoCs, pp.289–296, 2014.

[8] M. Yoshimi, R. Kudo, Y. Oge, Y. Terada, H. Irie, and T. Yoshinaga, “Accelerating OLAP workload on interconnected FPGAs with Flash storage,” Proceedings of 2nd International Workshop on Computer Systems and Architectures(CSA’14), pp.1–7, 2014.

[9] AVAL DATA, “APX880,” https://www.avaldata.co.jp/english_08/products/giga/tera_storage/apx880.html, 2011.

[10] L. Woods, Z. István, and G. Alonso, “Ibex – an intelligent storage engine with support for advanced sql off-loading,” Proceedings of the VLDB Endowment, vol.7, pp.963–974, 2014.