

FPGAリモートラボラトリー：ハイブリッドクラウド内におけるセミオートマチック環境の実装手法の提案

豊田, 雄一 / TOYODA, Yuichi

(出版者 / Publisher)

法政大学大学院情報科学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 情報科学研究科編

(巻 / Volume)

11

(開始ページ / Start Page)

1

(終了ページ / End Page)

6

(発行年 / Year)

2016-03-24

(URL)

<https://doi.org/10.15002/00012883>

FPGA リモートラボラトリー：ハイブリッドクラウド内における セミオートマチック環境の実装手法の提案

An FPGA-based Remote Laboratory: Implementing Semi-Automatic Experiments in the Hybrid Cloud

豊田 雄一*

Yuichi Toyoda

法政大学大学院情報科学研究科情報科学専攻

Email: 10k1027@stu.hosei.ac.jp

Abstract—An FPGA-based remote laboratory implemented in a hybrid cloud for semi-automatic FPGA-run experiments is presented. The design file and the FPGA test flow file including the FPGA-run parameters are sent to the hybrid cloud to trigger the sequence of the following tasks automatically: the logic-synthesis, and the FPGA-run, with the video recording. In this way, the amount of the users works can be reduced. The user only prepares the high-level Verilog HDL description and a few FPGA device setup parameters. The remote laboratory manager which provides semi-automatic experiment service handles the rest of the works, and the FPGA-run results together with the live recorded video are returned to the user for further inspection/debugging. This semi-automatic experiment service eliminates many cumbersome user works. Each user can have an access to the remote laboratory through a web browser and communicate with the experiment task manager via the Web Services and the WebSockets. As the manager acts like an agent for the user and realizes the semi-automatic experiment, the user can concentrate on the high-level designs. The user can get the feeling of the exclusive and interactive use of an FPGA-based experiment platform, while allowing efficient sharing of the platforms at the same time. In case of the device usage conflicts, the task manager resolves the contentions by reallocating the platforms in both space- and time-division fashions.

I. 序論

近年、様々な FPGA (Filed-Programmable Gate Array) をベースとしたリモートラボラトリーシステムが提案されている。その多くのシステムでは、実験環境を擬似的にシミュレーションするか、ネットワークを通じて実験機器をリモートで使用方法が取られている。そのような FPGA ベースのリモートラボラトリーの中でも主にリアルタイムとノンリアルタイムの2つのプラットフォームが提案されている。

ノンリアルタイムの FPGA リモートラボラトリーの場合、FPGA デバイスのようなリソースを意識せずに、FPGA による開発や実験環境の効率的なサービス提供を可能にしている [1],[2],[3]。そして、リアルタイムのリモートラボラトリーの場合、Web カメラによるリアルライブや FPGA ボードと直接通信できるインターフェイスを提供するサービスを実現している [4],[5],[6]。しかし、デザインファイルに何かしらのバグがある場合、FPGA 実行するまでの手順に時間がかかる場合があり、ユーザーは面倒なデバイスのテストをやり直すことになる。その結果、テスト失敗の度にユーザーの

負担がかかることになる。その問題に対して、ベクターファイルを利用した簡易ロジックアナライザーを FPGA に実装しシミュレーションファイルとして吐き出す方法を取ることで、FPGA のテスト自動化を可能とし、ユーザーの負担を軽減する方法も提案されている [7]。その他にも、多人数がリモートラボラトリーを利用するためにあたり、ネットワーク帯域やマシンのスペック不足の問題があったが、これはクラウドコンピューティングサービスを利用することで解決している [8],[9]。しかし、スペックの問題は解決できていても、ユーザーがテストするためのデバイスのリソースは限られているため、多人数が同時に利用する場合には競合がおり、ユーザーは他のユーザーがデバイスをリリースするまで待たなければならない状態が発生するほか、仮にテストをやり直す場合、ユーザーは再度コンパイルしたあとで面倒なテスト（スイッチを押すなど）を繰り返さなければならない。

本論文では、学生が授業で使用する教育用として、セミオートマチックな FPGA 実行環境を導入したノンリアルタイムの FPGA リモートラボラトリーのプロトタイプを実装を提案する。このシステムは、ユーザーの実行命令をトリガーに、コンパイルから FPGA テスト実行を一括に行う環境を提供する。不確定多数のユーザー（学生）を想定し、実験機器と共存しながら、スケーラビリティな環境の構築を行うため、ハイブリッドクラウド環境での構築を考える。また、ユーザーのデバイス依存のテストを平等に捌くためスケジューリングを行う。このジョブスケジューリングは、FPGA 実行のテストのリクエストに対して、到着順に空いているラボサーバーに割り当てる。そして、FPGA のテストを、リモートラボラトリー上で記録・処理するため、テストを行う際には、予め、ユーザーがテストの手順（今回はスイッチとキー）を決め、そのジョブを、提案する FPGA リモートラボラトリーが処理を行うことで、デバイスのテストを軽減し他のユーザーと共有しながら並行に実行できる。プロトタイプとして実際のラボラトリーのデバイスリソースとして、FPGA ボードである Altera DE2-115 board [10] をベースに構成され、1 つにつき 1 台のラボサーバーとつながっている。テストを割り当てるスケジューリングや FPGA の実行はプライベートクラウド上の Web サーバーとして構築され、ユーザーが利用するスケーラビリティ可能なサービスはパブリッククラウド上に移行する。システムのサーバー間ではパブリッククラウド内での効果的なリ

* Supervisor: Prof. Yamin Li

モートラボラトリーを提供できると考えられる。

II. システム・コンセプト

図 1 は提案リモートラボラトリーの全体概要である。これは、FPGA のセミオートマチックテストを実現しており、各ユーザーは実験環境を専有しているように思える。

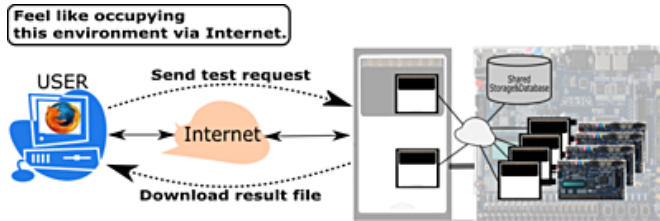


図 1. FPGA リモートラボラトリーの基本的コンセプト

ユーザー（学生）は、多数の共有されたデバイス依存のラボマシンのテストの割り当てや実行をリモートラボラトリーに任せることで、ユーザーの負担を軽減しつつ、他のユーザーと並行に利用できる。デバイスに依存するサービスは Web サービス・アプリケーションという形で運営され、オンプレミス・プライベートクラウドとして、複数ラボマシンがプライベートクラウドと結合する形で構築される。ライセンスのあるソフトウェアサービス（論理合成・配置配線）などの、スケーラブルが必要な他のサービスはパブリッククラウドで構築する。ユーザーのテストを平等に実行するため、セミオートマチック FPGA 実行環境は、空間的にも、時間的に分割され、ユーザーのテスト自動化を実現し、結果、スケーラブルかつ、ユーザーにとって効率的なリモートラボラトリーを実現できる。

ユーザーは Web ブラウザと WebSocket プロトコルを利用して提供された Web アプリケーションにアクセスできる。この WebSocket プロトコルは、一度、接続が設立される場合、ユーザーとサーバーの互いに同一と確立され、サーバーとユーザー側に相互通信するために使用し、プッシュ機能を使用してリモートラボラトリーの状況をユーザーに送る。

サービス一連の流れは、Verilog HDL の論理合成・配置配線、FPGA テスト実行である。ユーザーは、多くのラボ・プラットフォームを利用し、異なる設定された実験を並行に実行できる。またデバイスの空きが十分に利用できる場合、時間分割のプラットフォームにおいて、多数のユーザーのテストは 1 つずつ処理できると考えられる。実験結果は、共有されたデータベースに自動的に保存され、ユーザーは後に実験結果をダウンロードしチェックできる。仮に、実行中に、ユーザーと接続が切断されても、結果はデータベースにあるので、ユーザーは後で結果を見ることが可能である。

図 2 はこのセミオートマチック実験環境の概要を示す。一人のユーザーは、プラットフォーム上の多くの FPGA ボードを別々の FPGA の実験を並行に実行している。ユーザーは、リモートラボラトリーに HDL デザインファイル、ピンアサインファイル、そして、FPGA テスト手順のファイルをリモートラボラトリーにアップロードし、単一のジョブとして、セミオートマチック環境の実行を依頼する。複数のジョブをリモートラボラトリーに投げることも可能で、リモートラボラトリーは、ジョブの情報から論理合成などの

コンパイルから、FPGA テスト実行までを半自動的に行うことができる。

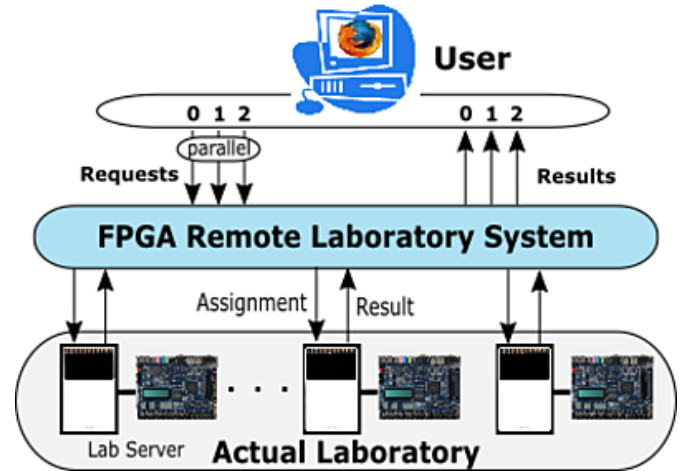


図 2. セミオートマチック環境の概要

提案するセミオートマチック FPGA 実行環境は、コンパイルから、テストのスケジュールと実行をリモートラボに任せるので、HDL 文法エラーが発生、あるいは、実験結果（ムービー）に間違いがある場合でも、訂正したデザインファイルをアップロードしなおし、再度ジョブを投げて再実行することができ、ユーザーは繰り返されるテストに露呈することなく、結果を待つだけで良く、コーディングに集中できる。

III. FPGA リモートラボラトリーの概要

図 3 で示されているように、FPGA リモートラボラトリーはハイブリッドクラウドで構成されている。FPGA リモートラボラトリーは、なるべく同じ動作環境をユーザーに与えるため、デバイスに依存している実際のラボラトリーとつながっている。ハイブリッドクラウドは VPN (virtual private network) によってパブリッククラウドとオンプレミス・プライベートクラウドの異なるネットワーク環境を結んでいる。オンプレミス・プライベートクラウドは Web サービスおよびファイル転送（デザインや結果など）をするためパブリッククラウドとユーザーコンピュータにおいてラボサーバーの間と密接に動作し、セミオートマチックな環境下での FPGA テストを実行する。FPGA テスト実行以外の残りサービスはパブリッククラウドで処理が可能であり、スケールアウト可能である。

また、一般的な仮想化の代わりに、ポータビリティかつホスト OS から独立しているコンテナを使用する。将来的に、他マシンからクラウドにシステムを移行性が高く、クラウド間での移行やスケールアウトと収縮時間は短縮できると考えられる。本論文でのプロトタイプでは、コンテナ管理ソフトウェアとして Docker を使用する [11]。この、Docker コンテナは、ホスト以外の外部コンピュータと接続するための IP アドレスを持つことができないため、ホストマシンの IP アドレスを共有する。図 4 は複数のコンテナが外部と接続している方法を示している。コンテナは、外部のネットワークと接続するために、ポートフォワーディングを用

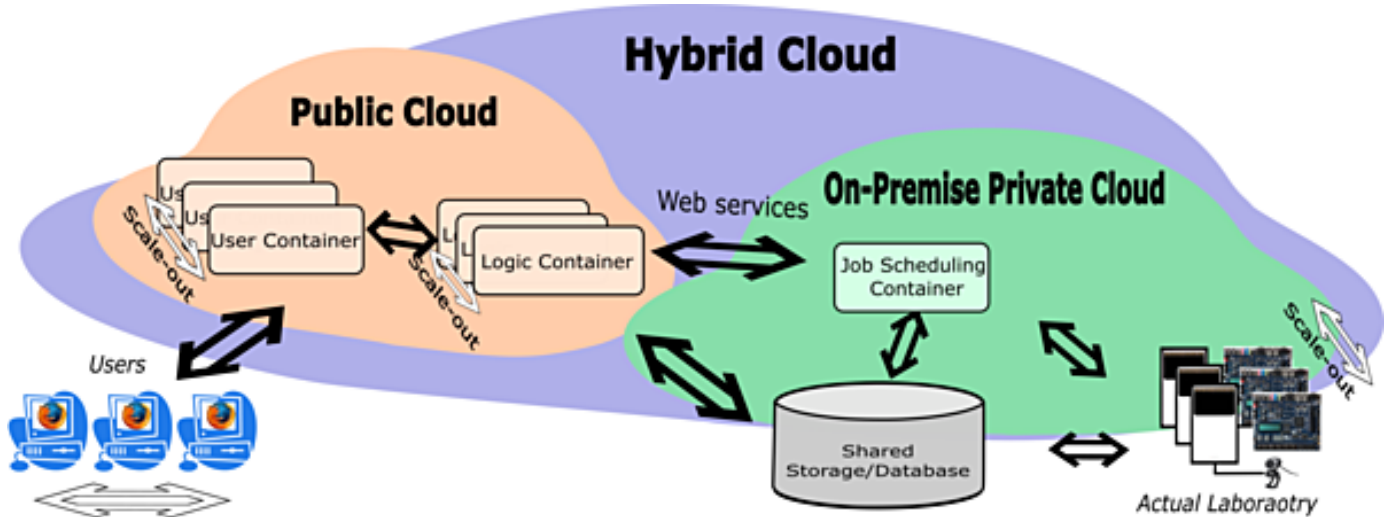


図 3. ハイブリッドクラウド内 FPGA リモートラボラトリーの全体図

いている。そして、パブリックやプライベートクラウド内で Web サービスとして提供され、ユーザーは、Web ブラウザから、提供された Web アプリケーションを利用して提供されたサービスを利用する。

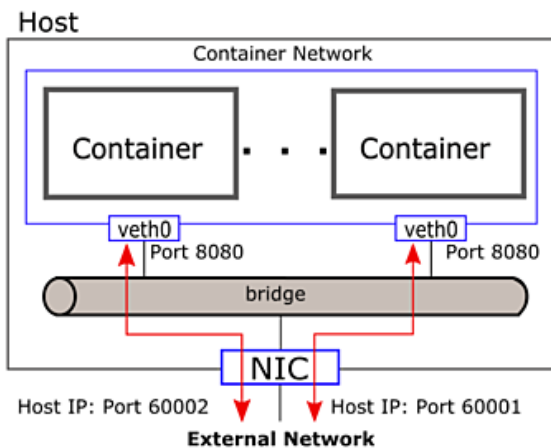


図 4. コンテナにおける通信

A. スケーラビリティなパブリッククラウドの概要

パブリッククラウドではユーザーの HDL デザインファイルをコンパイルできるライセンスソフトウェアサービスや、サービスをコントロールする Web サービス・アプリケーションが構築されている。ユーザーは、必要なファイルをプライベートクラウドに存在する共有データベースにアップロードとダウンロードすることができ、デザインファイルのコンパイルからデバイスに依存している Web サービス (FPGA テスト実行) を利用することができる。これらサービスは、プライベートクラウド側のラボサーバーで構築されており、パブリッククラウド側の Web サーバーから Web ブラウザを通じて利用することが可能である。また、WebSocket プロトコルによりユーザーコンテナとユーザーは双方向に接続されているため、サーバーの状況を確認しながら、リモート

ラボラトリーの Web サービスを Web ブラウザ上で利用することができる。そして、ユーザーコンテナは、Web ページを動的に更新を行うため、ユーザーは Web ページを遷移することなく、与えられたサービスに利用する。

1) デザインファイルをコンパイルするためのロジックコンテナ: ライセンスが紐付けられたソフトウェアが、このコンテナに存在しており、今回は、Altera Quartus II ソフトウェア [7] によって、ユーザーが設計した HDL デザイン (Verilog HDL) をコンパイル (論理合成からコンフィグレーションファイル作成) を行う。コンフィグレーションファイル (SOF ファイル) とは、FPGA テスト実行のために必要なファイルである。最初に、このコンテナがリクエストを受け取り、ピンアサインの情報とユーザーの HDL デザインファイルを取得する。そして、予め用意している HDL モジュールである、FPGA と通信するために必要な通信用のモジュールと、スイッチとキーの仮想コントロールモジュール、そして、ユーザーのデザインファイルを入れた、一つの HDL トップモジュールを作成しコンパイルする。このトップモジュールは、ピンアサインの情報から、ユーザーが定義したスイッチとキーのピンのアサインを除外し、仮想コントロールのモジュールからアウトプットされている仮想的なスイッチとキーにアサインする。そして、Verilog HDL に文法エラーや潜在的なエラーを含んでいる場合、ユーザーコンテナにエラーを返し、成功している場合は、ユーザーコンテナにその旨を通知する。

2) リモートラボラトリーサービス利用のためのユーザーコンテナ: 図 5 の通り、このコンテナには、Web サーバと、それに関連する Web アプリケーションがあり、FPGA リモートラボラトリーのサービスを実行する。サービスは、ファイルマネジメントや、Verilog-HDL のエラーチェック実行し、エラーのリターンや、リクエストを Web ブラウザを通してプライベートクラウド内のラボサーバーに送りこむ。ユーザーはコンテナの Web アプリケーションによって、共有データベースから必要なファイルをアップロード・ダウンロードでき、実行のリクエストをこのリモートラボラトリーへ送ることができる。また、このコンテナは、

共有データベースにユーザーのデザインファイル（Verilog HDL）をアップロードする前にデザインファイルの、簡単な文法チェックを実行する。もしデザインファイルに何かしらの文法エラーがある場合、ユーザーコンテナは共有データベースにアップデートを行わない。代わりに、ユーザーに対してエラーメッセージを返す。この Verilog HDL の文法エラーチェックには、Open Source Software (OSS) である Icarus Verilog[12] を使用している。

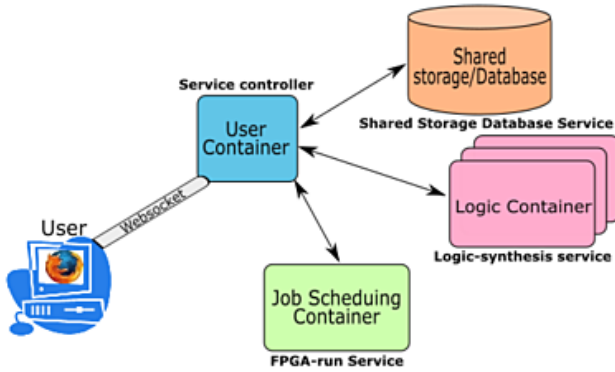


図 5. ユーザーコンテナの概要

B. FPGA テスト実行のためのオンプレミス・プライベートクラウド

オンプレミス・プライベートクラウドでは、FPGA 実行サービスのようなデバイス依存するサービスを構築するため、実際のラボラトリーに存在する多数のラボサーバーによりそれらの処理を行う。そのため、オンプレミス・プライベートクラウドは、実際のラボラトリーと結合され、コンテナは、FPGA のテスト実行を行うことができる。デバイス依存のサービスでは、利用できるデバイスの数は限られているため、競争を避け、テストを平等に分配するため、ジョブスケジューリングコンテナによって、多数存在するラボサーバーを管理し、テストの割り当てを行う。

1) **FPGA テスト実行のためのジョブスケジューリングコンテナ**: 図 6 の通り、このコンテナが、FPGA テスト実行サービスを平等にスケジューリングを行うため、複数のラボサーバーと通信を行う。FPGA 実行サービスのリクエストが来た際、ジョブスケジューリングコンテナは、多数の FPGA 実行のサービスのリクエストを捌き、テストの衝突を解決するため、テストを割り当てられていないラボサーバーにテストを到着順に割り当てる。そして、すべてのラボサーバーにテストが割り振られている場合は、ジョブスケジューリングコンテナはリクエストを待機キューに入れ、ラボサーバーがタスクを終了するまで待機する。同じサーバー内で、FPGA 実行サービスは同時に実行されないため、ジョブスケジューリングコンテナは、空間分割および時間分割の手法において、FPGA 実行のサービスを空いているラボマシンに割り当てることになる。ラボサーバーがテストを完了した際には、終了したことをジョブスケジューリングコンテナに知らせ、待機キューにリクエストがある場合は、次のテストを、空いたラボサーバーに割り当てる。そして、終了時にはユーザーコンテナにテストを完了したことを知らせ、ユーザーは結果の動画をダウンロードすることができる。

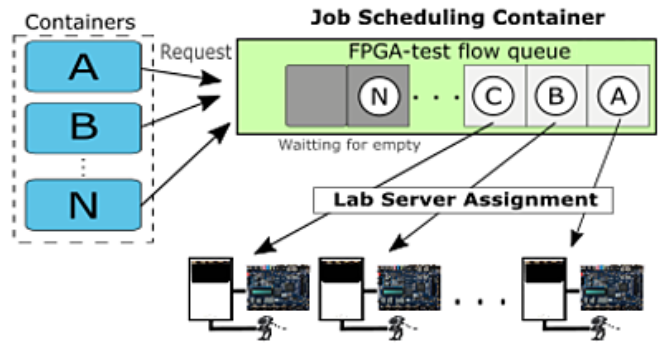


図 6. ジョブスケジューリング

C. 共有ストレージ・データベース

ユーザーの情報や、ジョブの情報、アップロードされた Verilog HDL ファイル、FPGA テスト実行の結果のビデオファイル（MP4 ファイル）などの、ほぼすべてのリソースは、ここに保存、そして外部から参照される。すべてのコンテナは共有ストレージ・データベースにアクセスし、必要に際して、情報を参照、ファイルをアップロード・ダウンロードできる。また、ユーザーデザインである Verilog HDL ファイルを共有ストレージ・データベースにアップロードする際には、パブリッククラウドに存在するユーザーコンテナの Verilog HDL コンパイラ（Icarus Verilog）が、ソースファイルを簡単な文法チェックを行うため、すべての Verilog HDL ファイルには、潜在的エラーが存在するものの、簡易なエラーにおいてはアップロード時に弾かれるので、ある程度の保証がある。共有ストレージ・データベースでリソースを共有することで、リモートラボラトリーはシームレスな環境をユーザーに提供し、仮に、ユーザーがセッションを閉じたとしても、テストをバックグラウンドで実行し結果を保存することができるため、ユーザーは後から結果である動画を見ることができる。

IV. ラボサーバーの概要

セミオートマチック環境下における、FPGA テスト実行サービスを実現するため、デバイスに依存したラボラトリーは、プライベートクラウドと結合され、実装されている。ラボラトリーにはテストの対象デバイスである FPGA と繋がれ、通信を行うことができる複数の同一なラボサーバーで構成され、それぞれのラボサーバーはジョブスケジューリングコンテナによって 1 ノードとしてテストが実行可能であるかどうか管理されている。FPGA テストの実行中はサーバー側からユーザーが予め決められた動作手順に従って FPGA にデータを送り込み、FPGA ボードの操作を行う。そうすることで、ユーザーはスイッチとキーの操作をサーバーに委ねることができ、自動的な FPGA のテストを実現する。そして、ラボサーバーがテストを完了した際には、結果は共有のデータベースに保存する。図 7 は、実際のラボサーバー構成を示す。なお、FPGA へのコンフィグレーションファイルのプログラミングは Quartus II ツールを使用し JTAG (Joint Test Action Group) 経由で行う。

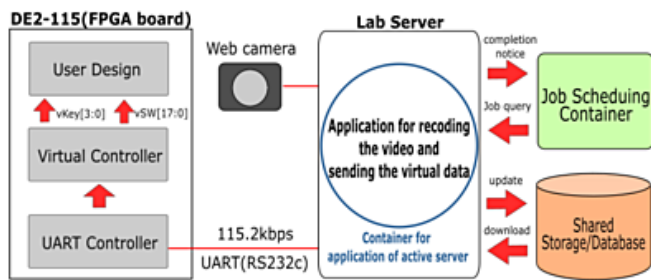


図 7. ラボラトリーサーバの概要

A. セミオートマチック環境における *FPGA* テスト実行

ユーザーは *FPGA* のデバイスをリモートで直接操作することや動作を確認することはできない。そのため、ラボサーバーは *FPGA* へ UART (Universal Asynchronous Receiver Transmitter) 経由でデータを送り込み、*FPGA* 内部で構築した仮想的なコントローラーを使用する。ラボサーバーがユーザーの代わりに、予めユーザーが決めた *FPGA* テストフローに沿ってテストを実行し、その動作確認として Web カメラを通じて、動作状況を動画として記録する。前もってユーザーが決めた手順で *FPGA* のテストをラボサーバーが行うため、リモートラボラトリーのシステムにユーザーは任せられた状態で、自動かつリモートでの実験環境を実現する。*FPGA* に送られるコントロールの情報である *FPGA* テストフローのファイルは、ユーザーが決めるテストの手順であり、キーやスイッチのプッシュ・プル の操作のタイミング情報である。ユーザーにより定義されたテストフローに従い、UART シリアル通信プロトコルに沿って、*FPGA* と通信し制御する。なお、通信量は毎秒 115.2 kbps なので、見た目ではほぼ遅延がない。そして、ラボサーバーは JTAG によって、ユーザーがデザインしたコンフィグレーションファイルを *FPGA* へプログラミングし、その後で、1 秒間隔で、キーやスイッチのシグナル情報を実行中に *FPGA* へ送り込む。そして、Web カメラによって決められた時間、*FPGA* 実行中の映像を録画する。*FPGA* テスト終了後、ラボサーバーは、ジョブスケジューリングコンテナに終了通知を行い、次のリクエストが来るまで、待機状態になる。そして、録画したムービーファイルは共有データベースに保存され、後で、ユーザーはダウンロードを行い、結果を確認することができる。

V. プロトタイプによるテスト分配の事前評価

提案されたシステムの示すために、デバイス依存に関係するプロトタイプシステムを構築した。図 8 は実際のラボラトリーのプロトタイプを示す。このラボラトリーは 2 つのラボサーバーによって構成されている。そして、図 9 は Web アプリケーションのメインとなる予定のテスト画面を示す。ユーザーは、この Web アプリケーションを通じて、ジョブの投下や、ファイルのアップロード・ダウンロードのマネージメント、そして、ジョブの状況のステータスを確認できる。ユーザーは、ピンやテストフローそして、HDL デザインソースをアップロード後、ジョブを作成し、システムにサービス要求を送る。もしすでにユーザーが各ファイルをアップロードしている場合は、ユーザーは *FPGA* ピンや *FPGA* テストフローのファイルなどを再利用できる予定である。

ンや *FPGA* テストフローのファイルなどを再利用できる予定である。

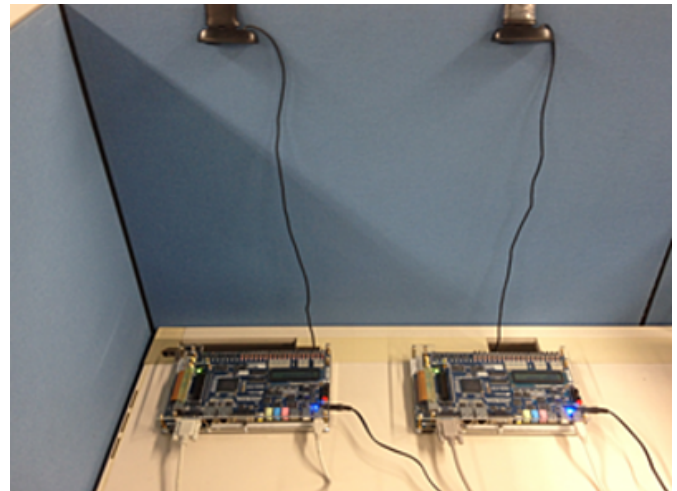


図 8. 2 つのラボマシンで構成されたラボラトリー

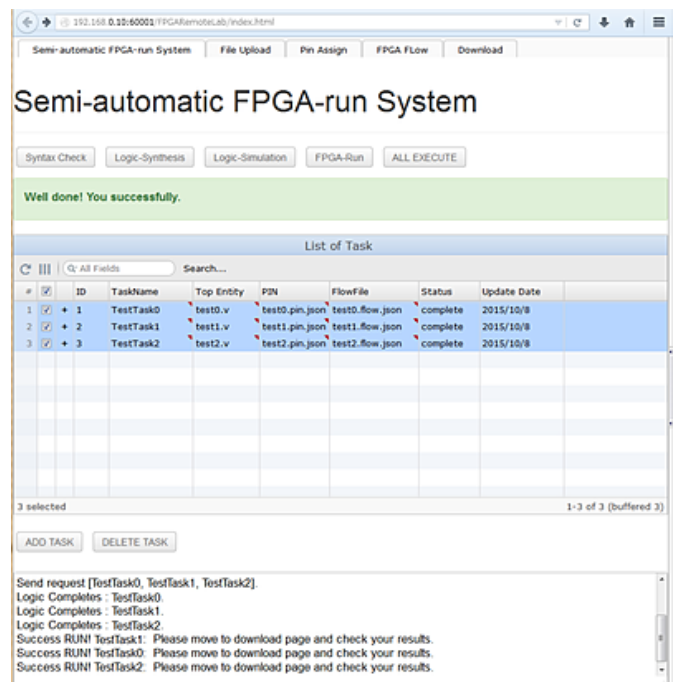


図 9. Web アプリケーション・プロトタイプ：テスト画面

このセミオートマチック環境において、提供されたサービスがユーザーに平等に分配されることを確認するため、図 10 にてジョブが投げられ、論理合成などのコンパイルから、ラボマシン 1 台あるいは 2 台を使用した自動的な *FPGA* テスト実行までのそれぞれの完了時間をグラフにして表している。評価の前提条件として、1 人のユーザーが 4 つのジョブを順番にリモートラボラトリーに送り、いずれも同じ HDL のデザインファイルを 1 分間テストする条件下のもとで行う。そして、ロジックコンテナからコンパイルされ、ジョブスケジューリングにテストのリクエストを行

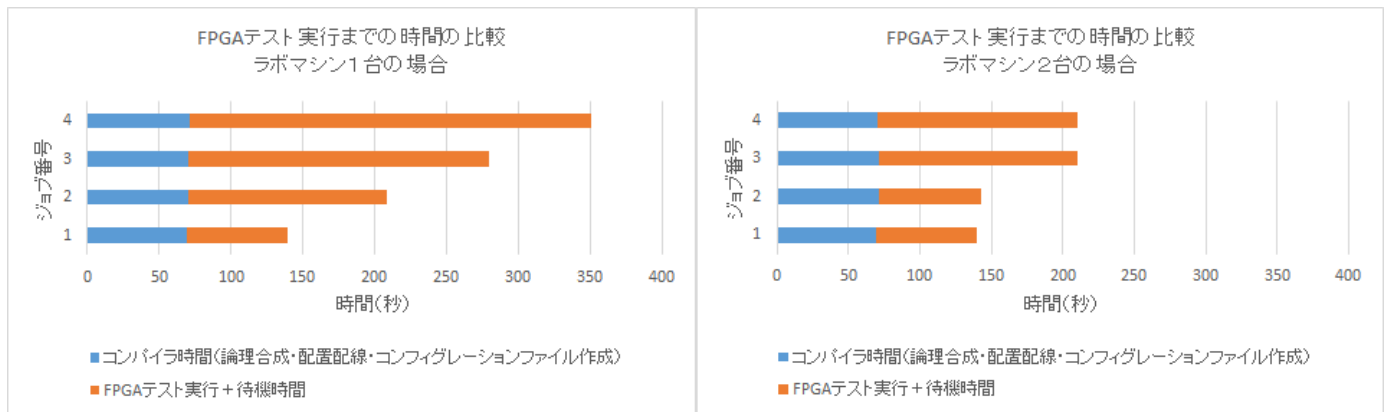


図 10. FPGA テスト割り当ての完了時間の比較

い、ラボマシンがテストを完了するまでの時間をそれぞれ割り出す。図 10 でわかるように、ラボサーバーが 1 台の場合、送られた順にテストが割り当てられるので、逐次的に後の方のジョブにつれて待機時間が長くなっている。2 台にラボサーバーが増えた場合、2 つまではそれぞれ割り当てられ、残りの 2 つが後に割り当てられていることがわかる。おそらく、4 台までラボサーバーが増えれば、ほぼ同じ時間に完了すると思われる。この結果から、ユーザーに対して提供されたサービスが分配されていることがわかる。そして、ユーザー（学生）にとって、実際の FPGA の実行時間は、空間分割および時間分割の手法において、多くのラボ・プラットフォームに割り当てられることができ、また、平均的な実行時間は同時に使用されるユーザー（学生）の数に従って、テスト完了までの時間は、増加することも推測できる。

VI. 締めと今後の課題

本論文は、ハイブリッドクラウドにおいて FPGA をベースとしたリモートラボラトリーにおけるセミオートマチック環境の実装について提案をし、平等にサービスを分配できたことを確認した。このセミオートマチックな環境をノンリアルタイムの FPGA リモートラボラトリーに取り入れることで、FPGA ベースのリモートラボラトリーがラボラトリーのデバイス・リソースを遊ばせることなく平等にユーザー（学生）たちにテストを割り当てることができ、また、デザイン失敗毎に繰り返される面倒なテスト（スイッチ・キーのプッシュ・プルなど）を避けられ、結果的にユーザー（学生）は HDL デザインのコーディングに集中できると考えられる。このプロトタイプは、現状、ハイブリッドクラウド上で構築する前段階として、それぞれ、FPGA と Web カメラに繋がれた、2 つのラボマシンと、パブリック用とプライベート用に分けられた複数の PC クラスタで構成されたラボサーバーを使用して FPGA テスト分配のテストを行った。また、使用したコンテナは移植性が高いので、他のクラウドからクラウドへ、将来的には容易に移行できると考えられる。今後の予定としては、クライアントサイドのウェブアプリケーション並びにこのプロトタイプのさらなる完成後、さらなる評価・テストを行いクラウドに移行する。予定としては、Amazon Web Service[13] をパブリッ

ククラウドとして利用し、プライベートクラウドについては、OpenStack[14] を使用して構築することで、ハイブリッド環境を構築する。

参考文献

- [1] Y. Toyoda, N. Koike, and Y. Li, "An FPGA-based remote laboratory: Implementing semi-automatic experiments in the hybrid cloud," in *Remote Engineering and Virtual Instrumentation (REV)*, 2016 13th International Conference on, Madrid, Spain, Feb. 24–26 2016.
- [2] N. Koike, "Cyber laboratory for hardware logic experiments: A seamless integration of actual laboratory and remote laboratory," in *Information Technology Based Higher Education and Training (ITHET)*, 2012 International Conference on, June 2012, pp. 1–5.
- [3] N. Fujii and N. Koike, "A time-sharing remote laboratory for hardware design and experiment with shared resources and service management," in *Information Technology Based Higher Education and Training, 2005. ITHET 2005. 6th International Conference on*, July 2005, pp. T2B/5–T2B/10.
- [4] J. Lobo, "A remote reconfigurable logic laboratory for basic digital design," in *Proceedings of the 1st Experiment International Conference - Remote Virtual Labs*, Nov 2011, p. 11.
- [5] M. Bascil, I. Yazici, and F. Temurtas, "A fpga based remote accessible digital system laboratory prototype," in *e-Learning and e-Technologies in Education (ICEEE)*, 2012 International Conference on, Sept 2012, pp. 49–53.
- [6] K. Ravanasa and R. Hashemian, "vlab, a high speed multi-accesses parallel processing remote laboratory access for fpga design technology," in *Electro/Information Technology (EIT)*, 2014 IEEE International Conference on, June 2014, pp. 377–381.
- [7] J. Jethra, S. Patkar, and S. Datta, "Remote triggered fpga based automated system," in *Remote Engineering and Virtual Instrumentation (REV)*, 2014 11th International Conference on, Feb 2014, pp. 309–314.
- [8] N. Koike, "Cyber laboratory: Migration to the hybrid cloud solution for device dependent hardware experiments," in *Information Technology Based Higher Education and Training (ITHET)*, 2014, Sept 2014, pp. 1–5.
- [9] F. Morgan, S. Cawley, M. Kane, A. Coffey, and F. Callaly, "Remote fpga lab applications, interactive timing diagrams and assessment," in *Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CICT 2014)*, 25th IET, June 2014, pp. 221–226.
- [10] Altera DE2-115 board webpage, <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=502>, [Online; accessed 12-August-2015].
- [11] Docker, <https://www.docker.com/>, [Online; accessed 12-August-2015].
- [12] Icarus Verilog, <http://iverilog.icarus.com/>, [Online; accessed 19-July-2015].
- [13] Amazon Web Serviced, <https://aws.amazon.com/>, [Online; accessed 19-July-2015].
- [14] OpenStack, <https://www.openstack.org/>, [Online; accessed 19-July-2015].