

DOI: 10.15514/ISPRAS-2026-38(3)-10



## Разработка виртуальной инфраструктуры для удаленного доступа к лабораторному оборудованию

*В.М. Винарский, ORCID: 0009-0005-5319-6304 <vmvinarskiy@edu.hse.ru>*

*А.А. Американов, ORCID: 0000-0002-5970-2125 <aamerikanov@hse.ru>*

*Л.Г. Евтушенко, ORCID: 0000-0003-1261-9735 <levtushenko@hse.ru>*

*И.И. Романова, ORCID: 0000-0002-2047-4225 <iromanova@hse.ru>*

*Е.В. Лежнев, ORCID: 0000-0001-9990-6467 <elezhnev@hse.ru>*

*Национальный исследовательский университет «Высшая школа экономики»,  
Россия, 101000, г. Москва, ул. Мясницкая, д. 20.*

**Аннотация.** Статья посвящена разработке инфраструктуры для удаленного доступа к лабораторному оборудованию с использованием средств виртуализации. Создана система, позволяющая удаленно взаимодействовать с установленными на платах программируемых логических интегральными схемами (ПЛИС) и выполнять научные исследования и практические работы. Виртуализация лабораторных стендов, состоящих из персональных компьютеров и подключенных к ним плат, позволяет централизовать управление оборудованием и уменьшить количество физических устройств, необходимых для организации работы и проведения экспериментов. Проведен анализ существующих решений, разработана архитектура удаленного доступа, реализованы методы виртуализации существующих стендов, организована сетевая инфраструктура для удаленного доступа к платам ПЛИС. В рамках апробации разработанной системы проведена косимуляция сетей на кристалле.

**Ключевые слова:** удаленный доступ, виртуализация, платы ПЛИС, удаленные стенды.

**Для цитирования:** Винарский В.М., Американов А.А., Евтушенко Л.Г., Романова И.И., Лежнев Е.В. Разработка виртуальной инфраструктуры для удаленного доступа к лабораторному оборудованию. Труды ИСП РАН, том 38, вып. 3, часть 1, 2026 г., стр. 171–186. DOI: 10.15514/ISPRAS-2026-38(3)-10.

**Благодарности:** Публикация подготовлена в ходе проведения исследования (№ 24-00-012 «Удаленная лаборатория для работы со встраиваемыми системами») в рамках Программы «Научный фонд Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ)».

## Development of a virtual infrastructure for remote access to laboratory equipment

*V.M. Vinarskii, ORCID: 0009-0005-5319-6304 <vmvinarskiy@edu.hse.ru>*

*A.A. Amerikanov, ORCID: 0000-0002-5970-2125 <aamerikanov@hse.ru>*

*L.G. Evtushenko, ORCID: 0000-0003-1261-9735 <levtushenko@hse.ru>*

*I.I. Romanova, ORCID: 0000-0002-2047-4225 <iromanova@hse.ru>*

*E.V. Lezhnev, ORCID: 0000-0001-9990-6467 <elezhnev@hse.ru>*

*HSE University,  
20, Myasnitckaya st., Moscow, 101000, Russia.*

**Abstract.** This article focuses on the development of an infrastructure for remote access to laboratory equipment using virtualization tools. A system has been created that enables remote interaction with FPGA boards and enables scientific research and practical work. Virtualization of laboratory setups consisting of PCs and connected boards allows centralized equipment management and reduces the number of physical devices required for workflow and experiments. An analysis of existing solutions was conducted, a remote access architecture was developed, methods for virtualizing existing setups were implemented, and a network infrastructure for remote access to FPGA boards was established. Cosimulation of networks on a chip was conducted to validate the developed system.

**Keywords:** remote access, virtualization, FPGA boards, remote testbed

**For citation:** Vinarskii V.M., Amerikanov A.A., Evtushenko L.G., Romanova I.I., Lezhnev E.V. Development of a virtual infrastructure for remote access to laboratory equipment. Trudy ISP RAN/Proc. ISP RAS, vol. 38, issue 3, part 1, 2026, pp. 171-186. DOI: 10.15514/ISPRAS-2026-38(3)-10.

**Acknowledgements.** The publication was prepared within the framework of the Academic Fund Program at HSE University (grant № 24-00-012 Remote laboratory for working with embedded systems).

### 1. Введение

Одним из этапов жизненного цикла разработки цифровых устройств является их прототипирование на специализированном оборудовании, таком как отладочные платы ПЛИС (программируемые логические интегральные схемы). Исследования показывают, что использование физического оборудования при подготовке специалистов в области микроэлектроники углубляет понимание концепций и улучшает навыки решения задач по сравнению с подходами, основанными исключительно на моделировании [1]. При этом доступ к реальному оборудованию не всегда возможен, что может быть обусловлено ограниченностью бюджета, недостатком необходимого оборудования и организационными сложностями, особенно в условиях массового применения.

Одним из способов решения указанных проблем является внедрение удаленных лабораторий, которые позволяют инженерам взаимодействовать с реальным оборудованием через сеть Интернет независимо от их географического положения [2]. В отличие от программ эмуляторов, удаленные лаборатории предоставляют доступ к реальному оборудованию и настоящим экспериментальным данным, что существенно приближает рабочие процессы к условиям реальной инженерной практики. Инструменты эмуляции, несмотря на свою доступность и удобство, обладают рядом недостатков, таких как упрощение реальных условий и отсутствие влияния внешних факторов и аппаратных ограничений [3]. Именно опыт работы с реальным оборудованием и связанными с ним сложностями является ключевым для формирования полноценного инженерного мышления и практических навыков.

Разработка инфраструктуры для удаленного доступа к стендам с платами ПЛИС и их виртуализация являются ключевой темой данной статьи. Результатом работы является

спроектированная архитектура и результаты ее реализации и апробации на промышленном оборудовании.

## 2. Способы реализации удаленных лабораторий

Существующие подходы к созданию удаленных лабораторий предлагают различные методы взаимодействия с оборудованием, а также способы развертывания программного обеспечения. Рассмотрим основные подходы к реализации удаленных лабораторий.

### 2.1 По типу лаборатории

Удаленные лаборатории могут быть реализованы двумя способами:

- программная эмуляция;
- взаимодействие с физическим устройством.

В первом случае для создания удаленной лаборатории не требуется наличия физического оборудования, а достаточно создания программного обеспечения, которое будет имитировать работу оборудования [4-5]. Такой подход удобно применять в той области, где оборудование достаточно дорого купить или сложно найти. В том же случае, когда необходимо провести более серьезное исследование, такого подхода может быть недостаточно.

Во втором случае реализация проекта предполагает создание физической лаборатории, к которой необходимо подключать серверную часть, оборудование, средства визуального наблюдения и взаимодействия с оборудованием, разработку архитектуры виртуальной инфраструктуры для удаленного доступа к лабораторному оборудованию, а также обеспечение безопасности, чтобы пользователи не смогли при неправильных действиях вывести из строя оборудование [3-6]. Дополнительно требуется разработать систему менеджмента, которая позволит пользователям резервировать время для работы с оборудованием, для избежания конфликтов одновременного доступа. Удаленная физическая лаборатория значительно выигрывает в сравнении с реализацией в виде эмулятора, так как позволяет выполнять те же эксперименты, что и в обычной лаборатории, а также получать обратную связь с помощью средств визуализации.

### 2.2 По способу взаимодействия с оборудованием

Взаимодействие пользователя с удаленным оборудованием может происходить:

- в синхронном формате;
- в асинхронном формате.

В первом случае пользователи работают с оборудованием в режиме реального времени. Пользователь на своем компьютере удаленно подключается к оборудованию лаборатории через специально разработанный сервис. Синхронный доступ менее эффективен в плане распределения ресурсов, а также может иметь проблемы, связанные с безопасностью оборудования. Дополнительно необходимо организовать очередь, в которую можно будет заранее записываться, чтобы подключаться в специально выделенное именно для этого пользователя время. При этом такой способ наиболее приближен к реальному взаимодействию с оборудованием.

В случае асинхронного способа работы [4, 7] пользователь в отличие от синхронного варианта не работает напрямую с оборудованием, в режиме онлайн, а отправляет набор команд для выполнения эксперимента, а затем получает отчет о его результатах. Отчет может быть оформлен как в текстовом формате, так и в виде видеозаписи выполнения требуемого сценария. В данном режиме у пользователя нет возможности пользоваться периферией, изменять какие-либо настройки в режиме реального времени. Все настройки и

взаимодействие с периферией необходимо прописывать заранее в файле со сценарием выполнения эксперимента. При реализации асинхронного режима взаимодействия отсутствует проблема конфликтов доступа к оборудованию.

### 2.3 По способу развертывания программного обеспечения

Существуют три основных способа развертывания программного обеспечения (ПО):

- физическое развертывание (baremetal);
- виртуализация;
- контейнеризация.

Стенды лаборатории, реализованные с помощью физического развертывания, где каждому стенду соответствует отдельный персональный компьютер (ПК) требуют значительных ресурсов в виде занимаемого физического места, потребляемого электропитания, дополнительного оборудования в виде сетевых коммутаторов, источников бесперебойного питания. Кроме того, выделение отдельного ПК под каждую плату ПЛИС является финансово неэффективным и усложняет дальнейшее масштабирование лаборатории. Физически развернутые стенды позволяют полноценно работать с платами ПЛИС и наблюдать различные физические процессы, происходящие с отладочными платами во время работы (нагрев, дрейбег контактов и пр.).

Виртуализация позволяет запускать несколько независимых виртуальных машин (ВМ) на одном физическом сервере с помощью специального программного обеспечения, называемого гипервизором. Виртуализация существенно экономит аппаратные ресурсы, позволяя централизованно управлять инфраструктурой, создавать и применять шаблоны виртуальных машин, делать их снимки (снэпшоты), а также использовать подход «инфраструктура как код» (Infrastructure as Code, IaC) [8] для автоматизации многих задач. Виртуализация обеспечивает удобное управление сетевой инфраструктурой с помощью программно-определяемых сетей (Software-Defined Networks, SDN) [9], при этом реализация коммутации физического оборудования внутрь виртуальных машин имеет технические ограничения и требует специализированных знаний в области виртуализации.

Контейнеризация представляет собой подход, при котором приложения запускаются в контейнерах – программных средах, которые позволяют запускать ПО в изолированных окружениях. Наиболее распространенной платформой для оркестрации контейнеров является Kubernetes [10], позволяющий автоматизировать развертывание, масштабирование и управление приложениями. Использование контейнеров и Kubernetes существенно упрощает добавление новых лабораторных стендов, уменьшает расход ресурсов и обеспечивает более эффективную реализацию IaC [11]. Тем не менее контейнеры обеспечивают меньшую степень изоляции по сравнению с виртуальными машинами – все контейнеры на одной системе делят единое ядро системы. Это может нести риски для безопасности и стабильности всей системы, а также ограничивает возможности непривилегированных контейнеров – запуск контейнера внутри контейнера зачастую не поддерживается. Кроме того, платформа Kubernetes является гораздо более сложной системой для управления и поддержки, требующей персонала с более высокими навыками и компетенциями. Исходя из данного исследования, было принято решение выполнить развертывание ПО способом виртуализации.

## 3. Разработка инфраструктуры

В настоящее время на базе лаборатории систем автоматизированного проектирования (САПР) Московского института электроники и математики им. А.Н. Тихонова Национального исследовательского университета «Высшая школа экономики» (УИ САПР МИЭМ НИУ ВШЭ) разработано базовое решение для удаленного доступа к удаленным

стендам с отладочными платами ПЛИС [12] на основе программы удаленного доступа к компьютеру AnyDesk [13].

Хотя этот вариант реализации предоставляет минимально необходимый функционал, его использование выявило несколько значительных проблем. Во-первых, AnyDesk является проприетарным программным обеспечением, что приводит к зависимости от внешнего сервиса и ограничивает адаптацию текущего решения под специфические потребности лаборатории. Во-вторых, у студентов могут возникать проблемы из-за конфликтов доступа, когда несколько человек пытаются одновременно подключиться к одному и тому же оборудованию, что создает неудобства и снижает возможности эксплуатации стенда.

Было проведено обзорное исследование и сравнение различных вариантов реализации программно-аппаратного комплекса, приведенное далее.

### 3.1 Выбор системы виртуализации

Виртуализация лабораторных стендов позволяет эффективно использовать ресурсы, размещая несколько изолированных сред на одном физическом сервере. Это обеспечивает гибкость в развертывании различных конфигураций стендов, упрощает управление и снижает затраты на оборудование. При выборе платформы виртуализации были рассмотрены три популярных решения: Proxmox VE [14], OpenStack [14] и XCP-ng (Xen) [15]. Сравнение данных решений выполнялось по пунктам представленным в таблице 1.

Табл. 1. Сравнение систем виртуализации.

Table 1. Comparison of virtualization systems.

	Proxmox VE	OpenStack	XCP-ng
<b>Гипервизор</b>	KVM	KVM	Xen
<b>Поддержка версии ядра Linux</b>	Актуальная (обновляется с Debian)	Зависит от дистрибутива	Устаревшая (CentOS 7-based, ядро 4.x)
<b>Поддержка контейнеров в системе виртуализации</b>	Да (LXC)	Да (LXC, Docker)	Ограниченная (через VM)
<b>Резервное копирование</b>	Встроенное, интеграция с Proxmox Backup Server	Требует сторонних решений	Xen Orchestra (требует компиляции или лицензии ХОА)
<b>Масштабируемость</b>	Подходит для малых и средних развертываний	Высокая, подходит для крупных облачных развертываний	Средняя, подходит для средних и крупных развертываний
<b>Сложность развертывания</b>	Низкая	Высокая	Средняя

Система виртуализации Proxmox VE имеет ряд преимуществ:

- 1) Простота развертывания и управления. Интуитивно понятный веб-интерфейс позволяющий быстро настраивать виртуальные машины и контейнеры;
- 2) Интеграция с Proxmox Backup Server. PBS обеспечивающая резервное копирование

с дедупликацией и сжатием, не требующая отдельной установки или лицензирования дополнительных компонентов;

- 3) Поддержка контейнеров. Встроенная поддержка LXC [16] позволяющая запускать легковесные контейнеры для различных задач, включая CI/CD, тестирование и сетевые сервисы;
- 4) Актуальное ядро Linux. Proxmox VE регулярно обеспечивает регулярную поддержку актуальных ядер Linux, что позволяет использовать современные технологии, включая поддержку новых аппаратных платформ и функций виртуализации.

В отличие от Proxmox, платформа OpenStack обладает высокой степенью масштабируемости, но требует значительных усилий на развертывание и поддержку множества компонентов. Это делает его избыточным для задач удаленной лаборатории.

Платформа XCP-ng, в свою очередь, использует гипервизор Xen с устаревшим ядром, что ограничивает возможности по интеграции современных технологий (например, vGPU). Кроме того, резервное копирование в XCP-ng требует ПО Xen Orchestra, которое либо компилируется из исходников вручную, либо приобретается по платной лицензии, что также усложняет обновления.

Платформа Proxmox также обладает активным форумом и широким сообществом пользователей, регулярно публикующих решения, статьи и рекомендации. Это значительно упрощает поиск информации и решение возникающих вопросов. Благодаря отсутствию значимых лицензионных ограничений, в отличие от XCP-ng, Proxmox VE может быть скачан и установлен без необходимости собирать гипервизор из исходных файлов. Обновления также распространяются бесплатно на тестовой ветке репозитория. Таким образом, Proxmox VE является наиболее подходящим для реализации лабораторной инфраструктуры.

### 3.2 Виртуализация стендов

После выбора платформы виртуализации следующим шагом стало развертывание базовой инфраструктуры. Был установлен и настроен сервер Proxmox VE, на котором подготовлен шаблон виртуальной машины на базе операционной системы Linux с установленной средой Quartus (основная система автоматизации проектирования для работы с ПЛИС компании Altera) [17].

Для обеспечения полноценной работы с аппаратным обеспечением, подключаемым по стандарту USB (в частности, с устройствами USB Blaster для прошивки ПЛИС), потребовалась настройка соединений USB-устройств в виртуальной машине, что было реализовано с использованием механизма «resources mapped» (отображение ресурсов) и позволило виртуальным машинам полноценно взаимодействовать с физическими USB-устройствами.

Применение данного механизма обеспечило автоматическое назначение необходимых прав доступа для устройств с идентификатором Intel/Altera USB-Blaster. Пользователь виртуальной машины был добавлен в группу «plugdev» (с возможностью монтирования внешних устройств), что гарантировало корректное определение и использование устройств внутри виртуальной среды. Такой подход позволил добиться прозрачного взаимодействия программного обеспечения Quartus с подключенными ПЛИС, обеспечив полноценную поддержку функций прошивки и отладки.

Дополнительно, для обеспечения надежности работы стендов, был развернут сервер Proxmox Backup Server (PBS). PBS осуществляет регулярное резервное копирование всех виртуальных машин на внешнее сетевое хранилище с использованием NFS (Network File System) [18], что обеспечивает быстрый откат в случае необходимости и минимизацию риска потери данных и минимизацию времени на восстановление работоспособности системы.

### 3.3 Выбор подхода к реализации удаленного доступа

Для организации удаленного доступа к виртуализированным стендам необходимо проанализировать несколько решений, каждое из которых обладает своими достоинствами и ограничениями. Наиболее простыми в реализации являются средства, встроенные в Proxmox VE, такие как SPICE Proxu и VNC [19]. SPICE Proxu обеспечивает высокую производительность удаленного рабочего стола, поддерживает низкие задержки и полноценную работу специальных сочетаний клавиш (например, Alt+Tab), что делает его удобным для постоянной работы. Недостатком SPICE Proxu является отсутствие гибкого управления пользовательскими сессиями: при подключении пользователя, разрывать сессию возможно только вручную, а для доступа требуется открывать TCP-порт наружу, что усложняет настройку сетевой безопасности. Также для доступа требуется устанавливать клиентское приложение на компьютер пользователя, что не всегда возможно.

Аналогичная ситуация наблюдается с использованием средств VNC. Этот протокол предполагает необходимость открывать отдельный порт для каждой сессии, что делает его неудобным для масштабируемой инфраструктуры с большим количеством стендов, либо в окружении с наличием строгого межсетевого экрана (firewall). Кроме того, VNC, как и SPICE, не предоставляет развитых механизмов управления пользователями и их правами доступа. В остальном, решения SPICE и VNC похожи.

Более продвинутым решением является Apache Guacamole [20] – платформа, обеспечивающая доступ к удаленным рабочим столам через сетевой браузер. Guacamole не требует установки агентов на управляемые машины (agentless). При этом она предоставляет базовое управление пользователями и их сессиями. Но у Guacamole отсутствует возможность предоставлять доступ к конкретным виртуальным стендам заранее заданные интервалы времени. Кроме того, настройка подключения для нового стенда требует конфигурации SPICE или VNC на нем, что затрудняет автоматизацию процесса.

Схожим решением является MeshCentral [21] – система, сочетающая в себе удобство управления сессиями и пользователей с простотой настройки. В отличие от других решений, MeshCentral позволяет централизованно управлять агентами, установленными на виртуальных машинах, обеспечивает возможность предоставления гостевых ссылок с ограничением по времени действия и позволяет объединить систему с внешними порталами бронирования ресурсов. Это важно для задач, связанных с временным доступом пользователей к стендам. Дополнительно MeshCentral, как и Apache Guacamole, поддерживает возможность записи экранных сессий на стороне сервера. Эта функция особенно полезна в образовательной среде, где важно контролировать активность студентов. Также MeshCentral предоставляет гибкое управление правами пользователей, позволяя выделять гранулярные уровни доступа: только просмотр, доступ к файлам, к консоли или полное управление виртуальной машиной. Это обеспечивает дополнительный уровень безопасности и удобства в администрировании. Наличие полноценного API упрощает интеграцию с другими системами. Также стоит отметить высокий уровень безопасности, достигаемый за счет работы через HTTPS (аналогично Apache Guacamole), с возможностью дополнительного ограничения IP-адресов.

В работе [22] представлено несколько архитектур удаленного доступа к стендам. Одна из них, приведенная на рис. 1, использует виртуализацию вместе с системой бронирования, которые объединяются с MeshCentral через API.

Архитектура имеет недостаток: при подключении внешних пользователей нет изоляции от общей сети организации, что может влиять на безопасность сетевой инфраструктуры.

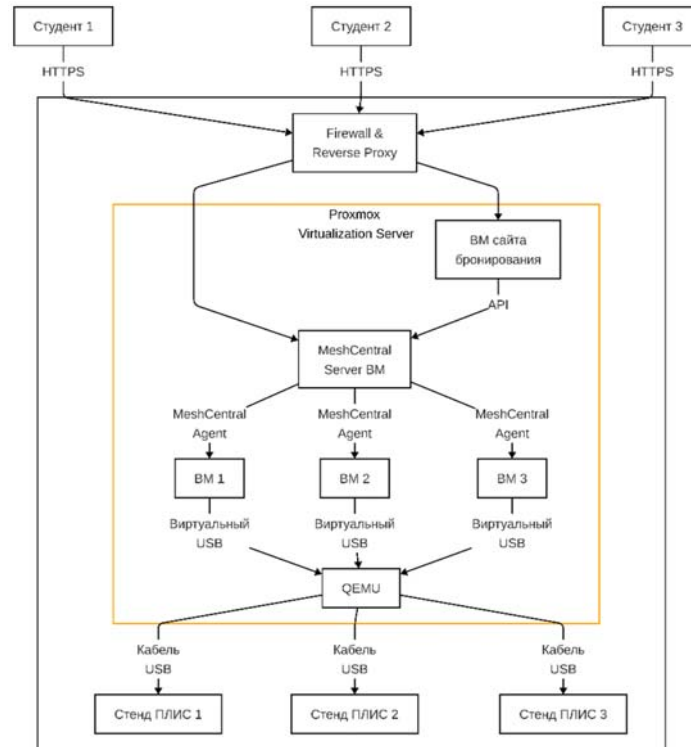


Рис. 1. Архитектура удаленных стендов [22].  
Fig. 1. Remote testbeds architecture [22].

### 3.4 Развертывание системы удаленного доступа

Для того, чтобы исправить недостаток, который имеется в архитектуре, представленной в работе [22], сервер виртуализации должен быть развернут в выделенной “демилитаризованной зоне” (DMZ сети), изолируя виртуальные машины студентов от сети университета. С помощью изолированной подсети обеспечивается безопасность сетевой инфраструктуры организации от несанкционированного доступа, через систему удаленных стендов. Рассмотрим более подробно реализацию предложенной архитектуры.

Для удобства управления инфраструктурой, развертывание системы MeshCentral осуществлялось в рамках единого стека с разработанной системой бронирования ресурсов, что позволило унифицировать подход к управлению всей платформой. Основой системы служит подсистема управления групповыми ресурсами Docker Compose, которая обеспечивает удобство развертывания, масштабируемость и изоляцию.

В состав конфигурации входят несколько ключевых компонентов: MeshCentral, система бронирования с разделением внешнего представления (front end) и внутренней реализации (back end), база данных PostgreSQL [23] и кеширующий сервер Redis [24]. Такое разделение позволяет гибко масштабировать отдельные элементы системы в зависимости от нагрузки. Сама система MeshCentral запускается в собственном контейнере с выделенными томами для хранения конфигурации, пользовательских файлов и резервных копий. Это обеспечивает как безопасность, так и сохранность данных при обновлениях или сбоях.

Сервисы взаимодействуют с другими компонентами через внутреннюю Docker-сеть. Ключевые параметры конфигурации вынесены в отдельный файл (.env), что обеспечивает удобное управление настройками без необходимости модифицировать код.

Комбинация архитектура подходит под все требования данной работы, так что было решено реализовать эту архитектуру. Данная архитектура позволяет реализовать процесс работы с помощью одного сервера, встроенного в серверную ячейку и настроенного на работу с большим количеством плат.

Было принято решение реализовать именно такую архитектуру стенов и сетевого взаимодействия ввиду того, что она позволяет объединить в рамках одного компактного сервера большое количество удаленных стенов (с различными платами) с помощью виртуализации, а также обеспечить к ним распределенный и разграниченный доступ с помощью системы бронирования, разработанной на базе системы MeshCentral.

### 3.5 Организация сетевой инфраструктуры для удаленного подключения

Сетевая архитектура виртуализированной инфраструктуры была разработана с учетом необходимости обеспечения надежного и безопасного удаленного доступа к лабораторным стендам из внешней сети. Основные задачи состояли в организации защищенного канала связи между внешними пользователями и внутренними сервисами, а также в изоляции трафика различных компонентов инфраструктуры. Общая схема архитектуры приведена на рис. 2.

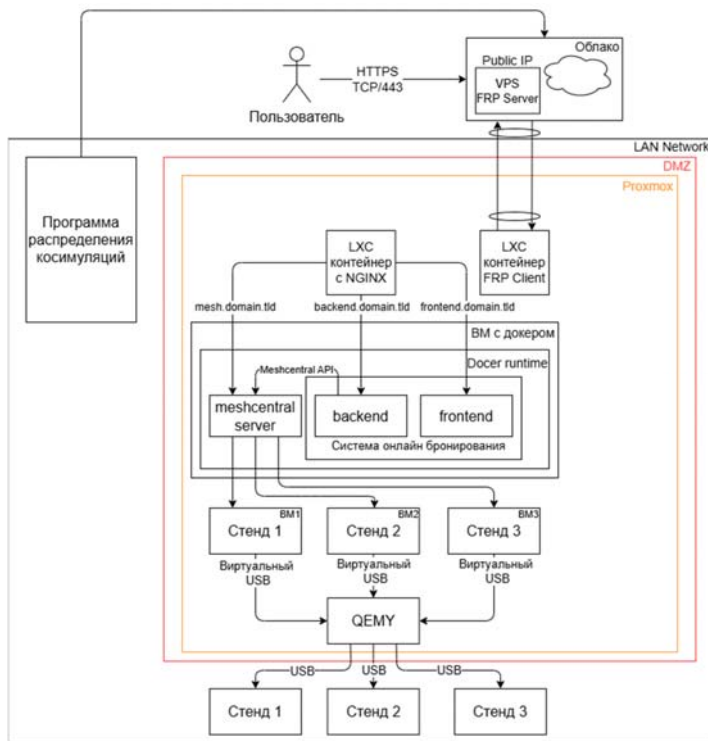


Рис. 2. Архитектура сетевого взаимодействия.  
Fig. 2. Network interaction architecture.

На внутреннем уровне используется сеть Proxmox VE с виртуальным мостом, который связывает виртуальные машины между собой и с управляющими сервисами. Это обеспечивает необходимый уровень связности между компонентами системы и позволяет управлять их взаимодействием, используя правила межсетевого экрана, и сегментированием через виртуальную локальную сеть (Virtual Local Area Network, VLAN).

Для организации удаленного доступа к внутренним сервисам, таким как MeshCentral, используется сервер Nginx [25], выполняющий роль обратного прокси-сервера. Его основная функция заключается в том, чтобы принимать все входящие запросы из сети Интернет, а затем пересылать их к нужному внутреннему сервису, скрывая внутреннюю структуру сети от внешнего пользователя. Это повышает удобство управления доступом и упрощает настройку безопасности.

Кроме того, Nginx обрабатывает все зашифрованные соединения с пользователями с использованием протокола HTTPS (SSL/TLS). Это означает, что Nginx расшифровывает входящий зашифрованный трафик и передает его дальше по внутренней сети уже в открытом виде (внутри сети гипервизора). Такой подход позволяет централизованно управлять сертификатами безопасности и снижает нагрузку на внутренние сервисы, избавляя их от необходимости самостоятельной обработки шифрования.

Для получения SSL-сертификатов используется сервис Let's Encrypt [26]. Обновление сертификатов автоматизировано с помощью инструмента Certbot [26], который регулярно проверяет срок действия сертификатов и продлевает их при необходимости, обеспечивая постоянную защищенность соединений.

Nginx настроен на прослушивание стандартных портов 80 (HTTP) и 443 (HTTPS), принимая все входящие запросы от пользователей. В зависимости от доменного имени, указанного в запросе с использованием механизма индикации имени сервера (Server Name Indication, SNI), сервер Nginx перенаправляет трафик к соответствующему внутреннему сервису. Это позволяет обслуживать несколько различных сервисов через один публичный IP-адрес и единые порты, обеспечивая гибкость в управлении доступом и возможность дальнейшего расширения инфраструктуры. В текущей конфигурации Nginx используется для маршрутизации запросов к серверу MeshCentral и сервису для бронирования, но этот же механизм позволит в будущем подключать и другие HTTP/HTTPS-сервисы лаборатории, используя уже существующую инфраструктуру и схему проксирования.

Для обеспечения доступа из внешней сети, в условиях ограничений сетевой инфраструктуры, используется промежуточный сервер, создающий обратные туннели (Fast Reverse Proxy, FRP) [27]. FRP-сервер размещается на внешнем виртуальном выделенном сервере (Virtual Private Server, VPS) с публичным IP-адресом и туннелирует трафик через защищенное соединение к внутреннему Nginx-серверу. Таким образом, обеспечивается безопасный доступ к внутренним сервисам, даже если сервер находится за преобразователем сетевых адресов (Network Address Translation, NAT) или межсетевым экраном.

Внутренняя инфраструктура организована с акцентом на простоту и гибкость развертывания новых компонентов. Такая архитектура позволяет добавлять новые стенды и сервисы без значительных изменений в конфигурации сети, что облегчает масштабирование системы и ее адаптацию под изменяющиеся требования.

### 4. Результаты тестирования

После развертывания всей инфраструктуры было проведено первичное тестирование с целью проверки работоспособности основных компонентов системы. Тестирование состояло из следующих этапов:

- 1) Проверка работы виртуальных машин с САПР Quartus и корректного соединения USB-устройств (USB Blaster);

- 2) Подключение к виртуальным стендам через MeshCentral с различных клиентских устройств;
- 3) Проверка работы обратного прокси Nginx и корректного завершения SSL-соединений, а также процедуры обновления сертификатов;
- 4) Тестирование туннелирования через FRP для обеспечения доступа из внешней сети.

Тестирование показало стабильную работу виртуальных машин и корректное функционирование проброса плат ПЛИС. Подключение через MeshCentral также прошло успешно, обеспечив возможность удаленного взаимодействия с виртуальными стендами. Все компоненты взаимодействовали корректно, что подтвердило работоспособность разработанной архитектуры.

Тем не менее, в ходе тестирования были определены направления для дальнейшего улучшения системы:

- 1) Добавление опции монтирования персональных файловых ресурсов студентов в виртуальные машины и сохранение результатов работы на стенде на сетевой диск при окончании бронирования;
- 2) Аутентификация пользователей сайта системы бронирования с университетской базой данных студентов и сотрудников;
- 3) Внедрение дополнительных протоколов удаленного доступа (например, SPICE с прокси), чтобы предоставить альтернативные методы взаимодействия с виртуальными стендами не через браузер.

## 5. Апробация виртуальной инфраструктуры для удаленного доступа на примере проведении косимуляции с использованием отладочных плат ПЛИС

В рамках работы [22] была предложена архитектура САПР для удаленного высокоуровневого моделирования сетей на кристалле (СтнК). Также в статье была представлена типовая схема проектирования СтнК, которая состоит из следующих этапов: составления технического задания, предварительного проектирования, высокоуровневого моделирования, низкоуровневого моделирования, прототипирования и косимуляции, производства.

Низкоуровневое моделирование систем на кристалле является важным этапом жизненного цикла проектирования, который занимает значительное количество времени. Также для проведения анализа характеристик систем на кристалле необходима многократная симуляция сети с разными параметрами. Косимуляция позволяет уменьшить требуемое количество времени в несколько раз [28], но она не решает проблемы многократной симуляции. Использование виртуальных стендов позволяет решить данную задачу за счет одновременного запуска нескольких симуляций, так как разработанная система удаленного доступа может управлять несколькими удаленными стендами одновременно.

Все этапы типовой схемы проектирования СтнК, кроме производства и косимуляции, можно реализовать удаленно, используя специализированное программное обеспечение. Этап косимуляции требует доступа к специализированному оборудованию, который может быть осуществлен с использованием разработанной архитектуры.

Для решения этой задачи в инфраструктуру системы удаленного доступа был добавлен сервер формирования заданий для симуляции. Одна его часть реализована в составе виртуальных машин и обеспечивает накопление заданий для моделирования и результатов моделирования. Вторая часть реализуется на пользовательских компьютерах и регулирует общий сбор результатов, а также распределением задач на виртуальных машинах. Получив IP адреса удаленных стендов, сервер связывает их в единую вычислительную систему и отправляет задания для симуляции. На удаленных стендах дополнительно была установлена

САПР низкоуровневого моделирования сетей на кристалле HDLNoCGen [29], которая, получая задания, подготавливает проект для САПР Quartus, который создает прошивку для ПЛИС компании Altera. Далее плата ПЛИС прошивается и выполняется совместная с HDLNoCGen косимуляция. Результаты работы отправляются обратно на сервер задач, для сохранения в базу данных. Схема работы системы показана на рис. 3.



Рис. 3. Схема системы косимуляции для удаленного стенда.  
Fig. 3. Schematic diagram of the co-simulation system for a remote testbed.

Тестирование работы системы косимуляции проводилось на 15 удаленных стендах. Было выполнено 4400 симуляций. На симуляции затрачивалось по 12 часов в день, а общее количество дней на получение данных составило 16. При выполнении аналогичной задачи получения данных на одном компьютере с подключенной платой ПЛИС, было бы затрачено более 120 дней.

Используемая в инфраструктуре удаленного доступа система бронирования позволяет гибко организовывать загрузку рабочего времени удаленного стенда для минимизации времени его простоя. В основное рабочее время для сервера задач симуляции предоставляется возможность создавать список задач, требующих малого времени для косимуляции, а все долгие задачи планировать на более продолжительное время простоя стендов. За счет внешнего доступа к стендам появляется возможность децентрализованного их размещения с возможностью организации в единый вычислительный комплекс.

## 6. Заключение

Изучение существующих решений позволило разработать новую архитектуру удаленного доступа к инфраструктуре на базе системы Proxmox VE, благодаря внедрению которой стало возможным оптимальное решение задач виртуализации лабораторных стендов. Использование данной платформы обеспечило простоту управления, надежность работы и возможность масштабирования системы. Для удаленного доступа к виртуальным окружениям был использован MeshCentral. Использование MeshCentral позволило обеспечить доступ пользователя к стенду через браузер, а возможность создание гостевых ссылок с ограниченным сроком действия упростила интеграцию в систему бронирования. Практическая апробация системы на примере косимуляции сетей на кристалле продемонстрировала высокую эффективность разработанного решения. Проведение 4400 симуляций на 15 удаленных стендах позволило сократить время получения результатов с 120 до 16 дней, что подтверждает значительное повышение производительности при использовании распределенных вычислений.

Созданная инфраструктура обладает существенным потенциалом для дальнейшего развития. Перспективными направлениями являются: интеграция с университетской системой аутентификации, расширение возможностей работы с пользовательскими файлами и масштабирование системы для обслуживания большего количества стендов.

## Список литературы / References

- [1]. Ma. J., Nickerson J. Hands-on, simulated, and remote laboratories: A comparative literature review. *ACM Comput Surv*, 2006, vol. 38, issue 3, p. 1, DOI: 10.1145/1132960.1132961.
- [2]. Gomes L., Bogosyan S. Current trends in remote laboratories. *IEEE Transactions on Industrial Electronics*, 2009, vol. 56, issue 12, pp. 4744-4756, DOI: 10.1109/TIE.2009.2033293.
- [3]. Gustavsson I. et al. On objectives of instructional laboratories, individual assessment, and use of collaborative remote laboratories, *IEEE Transactions on Learning Technologies*, 2009, vol. 2, issue 4, pp. 263-274, DOI: 10.1109/TLT.2009.42.
- [4]. Hardison J., DeLong K., Bailey P., Harward V. Deploying interactive remote labs using the iLab Shared Architecture. *Proceedings. Frontiers in Education Conference, FIE*, 2008, DOI: 10.1109/FIE.2008.4720536.
- [5]. Kazmierkowski M., Liserre M. Advances on Remote Laboratories and e-Learning Experiences (Gomes, L. and Garcia-Zubia, J., Eds.) [Book News]. *IEEE Industrial Electronics Magazine*, 2008, vol. 2, issue 2, pp. 45-46, DOI: 10.1109/MIE.2008.924851.
- [6]. Winzker M., Kiessling R., Schwandt A., Paez C., Shanab S. Teaching across the ocean with video lectures and remote-lab. *EDUNINE 2018. 2nd IEEE World Engineering Education Conference: The Role of Professional Associations in Contemporaneous Engineer Careers*, *Proceedings*, 2018, DOI: 10.1109/EDUNINE.2018.8451002.
- [7]. Harward V. et al. The iLab shared architecture: A web services infrastructure to build communities of internet accessible laboratories. *Proceedings of the IEEE*, 2008, vol. 96, issue 6, pp. 931-950, DOI: 10.1109/JPROC.2008.921607.
- [8]. Bessghaier N., Ouni A., Sayagh M., Chouchen M., Mkaouer M. Towards understanding code review practices for infrastructure-as-code: An empirical study on OpenStack projects», *Empir Softw Eng*, 2025, vol. 30, issue 3, pp. 1-52, May 2025, DOI: 10.1007/s10664-025-10654-w.
- [9]. Tkachova O., Salim M., Yahya A. An analysis of SDN-OpenStack integration. *2015 2nd International Scientific-Practical Conference Problems of Infocommunications Science and Technology, PIC S and T 2015. Conference Proceedings*, 2015, pp. 60-62, DOI: 10.1109/INFOCOMMST.2015.7357269.
- [10]. Ascensão P., Neto L., Velasquez K., Abreu D. Assessing Kubernetes Distributions: A Comparative Study. *2024 IEEE 22nd Mediterranean Electrotechnical Conference, MELECON*, 2024, pp. 832-837, DOI: 10.1109/MELECON56669.2024.10608706.
- [11]. Ochuba N., Kisina D., Adanigbo O., Uzoka A., Akpe O., Gbenle T. Systematic Review of Infrastructure as Code (IaC) and GitOps for Cloud Automation and Governance. *International Journal of Multidisciplinary Research and Growth Evaluation*, vol. 4, issue 2, pp. 664-670, 2023, DOI: 10.54660/IJMRGE.2023.3.2.664-670.
- [12]. Наваби З. Проектирование встраиваемых систем на ПЛИС. *Litres*, 2022. / Nawabi Z. Design of Embedded Systems on FPGAs. *Litres*, 2022. (in Russian).
- [13]. Application for fast remote – AnyDesk. Available at: <https://anydesk.com/en>, accessed 29.10.2025.
- [14]. Oleksiuk V., Oleksiuk O., Spirin O. Comparative Study of the Support of Academic Clouds Based on Apache CloudStack and Proxmox VE Platforms, 2023, pp. 349-361, DOI: 10.5220/0012064300003431.
- [15]. XCP new generation GitHub. Available at: <https://github.com/xcp-ng/>, accessed 29.10.2025.
- [16]. LXC - Linux Containers. Available at: <https://github.com/lxc/>, accessed 20.12.2025.
- [17]. Quartus Prime Design Software. Available at: <https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/overview.html>, accessed 20.12.2025.
- [18]. Network File System Version 4 (nfsv4). Available at: <https://datatracker.ietf.org/group/nfsv4/documents/>, accessed 29.10.2025.
- [19]. Hagström M. Remote desktop protocols: A comparison of Spice, NX and VNC, 2012.

- [20]. Hassan I. Leveraging Apache Guacamole, Linux LXD and Docker Containers to Deliver a Secure Online Lab for a Large Cybersecurity Course. *Proceedings. Frontiers in Education Conference, FIE*, vol. 2022-October, 2022, DOI: 10.1109/FIE56618.2022.9962510.
- [21]. GitHub - Ylianst/MeshCentral: A complete web-based remote monitoring and management web site. Once setup you can install agents and perform remote desktop session to devices on the local network or over the Internet. Available at: <https://github.com/Ylianst/MeshCentral>, accessed 29.01.2025.
- [22]. Американов А.А., Винарский В.М., Евтушенко Л.Г., Зунин В.В., Романова И.И. Разработка архитектур удаленных лабораторий для обучения проектированию на ПЛИС. *Информационные технологии*, т. 31, вып. 3, с. 124-130, мар. 2025, doi: 10.17587/IT.31.124-130. / Amerikanov A.A., Vinarskii V.M., Evtushenko L.G., Zunin V.V., Romanova I.I., CAD for remote high-level modeling of NoC. *Proc. ISP RAS*, vol. 37, issue 1, 2025, pp. 133-144. (in Russian).
- [23]. Official PostgreSQL GIT. Available at: <https://github.com/postgres/postgres>, accessed 29.10.2025.
- [24]. Redis GitHub. Available at: <https://github.com/redis/redis>, accessed 29.10.2025.
- [25]. The official NGINX Open Source repository. Available at: <https://github.com/nginx/nginx>, accessed 29.10.2025.
- [26]. Tiefenau C., Von Zezschwitz E., Häring M., Krombholz K., Smith M. A usability evaluation of let's encrypt and CertBot: Usable security done right, *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 1971-1988, 2019, DOI: 10.1145/3319535.3363220.
- [27]. Frp GitHub. Available at: <https://github.com/fatedier/frp>, accessed 29.10.2025.
- [28]. Ou J. и Prasanna V. MATLAB/simulink based hardware/software co-simulation for designing using FPGA configured soft processors. *Proceedings. 19th IEEE International Parallel and Distributed Processing Symposium, IPDPS 2005*, vol. 2005, DOI: 10.1109/IPDPS.2005.275.
- [29]. Lezhnev E., Zunin V., Amerikanov A., Romanov A. Electronic Computer-Aided Design for Low-Level Modeling of Networks-on-Chip, *IEEE Access*, 2024, vol. 12, pp. 48750-48763, DOI: 10.1109/ACCESS.2024.3382710.

## Информация об авторах / Information about authors

Владимир Максимович ВИНАРСКИЙ – студент национального исследовательского университета «Высшая школа экономики». Сфера научных интересов: системы на кристалле, ПЛИС, машинное обучение.

Vladimir Maksimovich VINARSKII – Student at the HSE University. Research interests: SoC, FPGA, machine learning.

Александр Александрович АМЕРИКАНОВ – кандидат технических наук, доцент национального исследовательского университета «Высшая школа экономики». Сфера научных интересов: разработка САПР, разработка логических устройств, ПЛИС, сети на кристалле.

Aleksandr Aleksandrovich AMERIKANOV – Cand. Sci (Tech.), Associate Professor at the HSE University. Research interests: CAD development, development of logic devices, FPGA, networks-on-chip.

Лариса Геннадьевна ЕВТУШЕНКО – старший преподаватель национального исследовательского университета «Высшая школа экономики». Сфера научных интересов: разработка логических устройств, ПЛИС.

Larisa Genadevna EVTUSHENKO – Lecturer at the HSE University. Research interests: development of logic devices, FPGA.

Ирина Ивановна РОМАНОВА – старший преподаватель национального исследовательского университета «Высшая школа экономики». Сфера научных интересов: Искусственный интеллект, разработка логических устройств, ПЛИС, сети на кристалле.

Irina Ivanovna ROMANOVA – Lecturer at the HSE University. Research interests: Artificial intelligence, development of logic devices, FPGA, networks-on-chip.

Евгений Владимирович ЛЕЖНЕВ – кандидат технических наук, доцент национального исследовательского университета «Высшая школа экономики». Сфера научных интересов: Разработка логических устройств, ПЛИС, сети на кристалле.

Evgeny Vladimirovich Lezhnev – Cand. Sci (Tech.), Associate Professor of the National Research University Higher School of Economics. Research interests: Development of logic devices, FPGA, networks-on-chip.