Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

**Факультет Информатика и системы Управления (ИУ)**

**Кафедра «Информационная безопасность» (ИУ8)**

**Отчет**

**по научно-исследовательской работе студента**

на тему Легковесные алгоритмы распознавания отпечатков пальца

ФИО студента: Синяков Артем Сергеевич Группа: ИУ8-104

Специальность: 10.05.01\_01 «Копьютерная безопасность»

Специализация: «Математические методы защиты информации»

Научный руководитель НИРС д.т.н., доцент Ключарев П.Г.

Работа выполнена 01.12.2015 А.С.Синяков

дата подпись студента И.О.Фамилия студента

Допуск к защите 01.12.2017 П.Г.Ключарев

дата подпись научного руководителя И.О.Фамилия научного руководителя

Дата защиты НИРС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Результаты защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва

2017

**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»**

# РЕФЕРАТ

Отчет 20с., 2ч., 4рис., 20 источников

~~ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ, СУПЕРКОМПЬЮТЕРЫ, ИМИТАТОРЫ, СИМУЛЯТОРЫ, ВИРТУАЛЬНЫЕ МАШИНЫ, АРХИТЕКТУРА, ПРОЦЕССОРЫ, ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ.~~

~~Объектом исследования является литература по теме и готовые решения для имитационного моделирования высокопроизводительных вычислительных комплексов.~~

~~Цель работы – обзор подходов, методов и существующих решений к имитационному моделированию высокопроизводительных вычислительных комплексов.~~

~~В процессе работы были рассмотрены имитаторы COTSon, Manifold, gem5 и другие.~~

~~Основными подходами к моделированию является функциональное моделирование и моделирование синхронизации, а также их сочетания. Основными задачами – нахождение баланса между точностью и скоростью, а также устранение ошибок моделирования, абстракции и спецификации.~~

# СОДЕРЖАНИЕ

[1. Вводная часть 7](#_Toc469327718)

[1.1. Обоснование актуальности темы 7](#_Toc469327719)

[1.2. Исходные данные НИРС 7](#_Toc469327720)

[1.3. Цели и задачи НИРС 7](#_Toc469327721)

[2. Основная часть 8](#_Toc469327722)

[2.1. Параллельные вычислительные системы. 8](#_Toc469327723)

[2.2. Зачем нужен параллелизм? 8](#_Toc469327724)

[2.3. Имитационное моделирование 8](#_Toc469327725)

[2.4. Ошибки и проблемы, возникающие при имитационном моделировании. 10](#_Toc469327726)

[2.5. Виртуальные машины: SimOS, QEMU и др. 11](#_Toc469327727)

[2.6. COTSon 11](#_Toc469327728)

[2.7. Manifold 12](#_Toc469327729)

[2.8. Gem5 12](#_Toc469327730)

[2.9. Другие имитаторы 13](#_Toc469327731)

[Заключение 14](#_Toc469327732)

[Список используемых источников 15](#_Toc469327733)

[Приложение 1. Пример выходных данных бенчмарка PARSEC 17](#_Toc469327734)

[Приложение 2. Схемы архитектуры эмуляторов gem5 и COTSon. 18](#_Toc469327735)

# Перечень используемых сокращений

|  |  |  |
| --- | --- | --- |
| ~~SIMD~~ | ~~-~~ | ~~Single Instruction Multiple Data~~ |
| ~~MIMD~~ | ~~-~~ | ~~Multiple Instruction Multiple Data~~ |
| ~~GPU~~ | ~~-~~ | ~~Graphics Processing Unit~~ |
| ~~CPU~~ | ~~-~~ | ~~Central Processing Unit~~ |
| ~~RAM~~ | ~~-~~ | ~~Random-access Memory~~ |
| ~~ОС~~ | ~~-~~ | ~~Операционная система~~ |
| ~~KVM~~ | ~~-~~ | ~~Kernel-based Virtual Machine~~ |
| ~~GPGPU~~ | ~~-~~ | ~~General-purpose computing on graphics processing units~~ |
| ~~ПО~~ | ~~-~~ | ~~Программное обеспечение~~ |
|  |  |  |
|  |  |  |

# Термины и определения

Дактилоскопия – способ опознания (идентификации) человека по отпечаткам пальцев (в том числе по следам пальцев и ладоней рук), основанный на неповторимости рисунка кожи.

Глобальный признак – признак отпечатка пальца в целом, такие как папиллярные узоры.

Локальный признак (минуция, minutiae) – признак одной конкретной папиллярной линии.

Окончание (termination) – локальный признак, при котором линия заканчивается.

Раздвоение (бифуркация, bifurcation) – локальный признак, при котором линия раздваивается на две отдельные линии.

Бинаризация – процесс преобразования изображения отпечатка пальца в двухцветный, черно-белый рисунок.

Утончение – процесс преобразования изображения отпечатка пальца таким образом, что ширина каждой линии становится равна одному пикселю.

# Вводная часть

## Обоснование актуальности темы

Дактилоскопия – один из наиболее известных, надежных, легких в использовании и безопасных методов биометрической идентификации. Бла бла бла при использовании со смарт-картами можно хранить отпечаток на карте, а сравнивать на терминале, но это открывает целый ряд уязвимостей, поэтому почему бы не посылать отпечаток на карту и сравнивать прямо там. Вычислительные мощности смарт-карт весьма ограничены, поэтому требуется легковесный алгоритм.

## Исходные данные НИРС

???.

## Цели и задачи НИРС

Целью данной работы является исследование существующих алгоритмов распознавания отпечатков пальца на предмет легковесности и применимости …

# Основная часть

## Принципы дактилоскопии.

Отпечаток пальца состоит из линий (темная часть рисунка) и промежутков (светлая, между линиями). Дактилоскопия – процесс сравнения известного отпечатка с другим отпечатком с целью определить, с одного ли пальца они получены.

…

## Виды алгоритмов распознавания отпечатка

Существуют разные подходы к распознаванию отпечатка, которые можно разделить на три группы:

* Основанные на паттернах

Эти алгоритмы сравнивают базовые паттерны отпечатка (дуги, завитки и петли) с ранее сохраненным шаблоном. Для этого необходимо, чтобы отпечатки были одинаково ориентированы. Алгоритм графически сопоставляет шаблон с отпечатком и определяет величину их совпадения.

* Основанные на минуциях

Это самый популярный и широко применяемый вид алгоритмов, основанный на методе, использовавшемся еще при сравнении отпечатков вручную. Из изображений извлекаются локальные признаки и хранятся как наборы точек на плоскости. Суть алгоритма в том, чтобы найти такое взаимное положение шаблона и входного набора минуций, что происходит наибольшее совпадение пар минуций.

* Основанные на корреляции

Два изображения накладываются друг на друга и находится корреляция между значениями пикселей при разных поворотах и смещениях.

## Предобработка

Во всех существующих алгоритмах распознавания отпечатка первым шагом является предобработка изображения для облегчения дальнейшего распознавания.

Первым шагом предобработки является бинаризация. Изображение, изначально представленное в виде оттенков серого, преобразуется в двухцветное согласно некоему пороговому значению.



Рис. 1 – Изначальное изображение (слева) и бинаризованное (справа)

Вторым шагом является утончение – приведение всех линий к ширине в один пиксель. Это преобразование сохраняет как локальные, так и глобальные признаки, и позволяет упростить алгоритмы их распознавания.



Рис 2 – Бинаризованное изображение (слева) и утонченное (справа)

В случае алгоритма, основанного на минуциях, следующим шагом предобработки является нахождение и определение локальных признаков и их параметров.

## FRMSM

FINGERPRINT RECOGNITION USING MINUTIA SCORE

MATCHING

## Ошибки и проблемы, возникающие при имитационном моделировании.

B. Black и J.P. Shen предлагают следующую классификацию ошибок имитационного моделирования:

* Ошибки моделирования - когда разработчик имитатора неправильно реализует требуемый функционал. Для широко используемых имитаторов такие ошибки постепенно сводятся на нет.
* Ошибки спецификации - когда разработчик не до конца понимает поведение моделируемого объекта.
* Ошибки абстракции - когда какие-то аспекты моделируемой системы не попадают в модель.

Эти ошибки могут вызвать неприемлемые экспериментальные отклонения, поэтому симулятор необходимо проверять на отсутствие этих ошибок во избежание ненадежных результатов.

Технология имитационного моделирования в последние годы сильно продвинулась вперед. Некоторые современные симуляторы способны на моделирование различных наборов инструкций, разные типы CPU и сложную топологию их взаимного соединения с такой точностью, что они способны запускать операционные системы и даже графические интерфейсы. Такое резкое увеличение возможностей аппаратного обеспечения приводит к увеличению ошибок спецификации и абстракции в современных имитаторах. Одна распространенная ошибка абстракции – отсутствие модели GPU. Такие имитаторы не могут с достаточной точностью эмулировать системы с графическими подсистемами.

Основными метриками работы имитатора считаются:

* Точность исполнения
* Масштабируемость многопоточности
* Точность системы памяти
* Точность микроархитектуры

Для оценки этих параметров используют специально созданные бенчмарки, такие как PARSEC. В приложении 1 можно видеть примеры выходных данных работы бенчмарка PARSEC для некоторых имитаторов.

## Виртуальные машины: SimOS, QEMU и др.

Для чисто функционального моделирования компьютерных систем существуют так называемые виртуальные машины. Они эмулировать аппаратное обеспечение целевой (target) платформы на хост-платформе (host). Виртуальная машина исполняет некоторый машинно-независимый код или машинный код реального процессора. Помимо процессора, виртуальная машина может эмулировать работу как отдельных компонентов аппаратного обеспечения, так и целого реального компьютера (включая BIOS, оперативную память, жёсткий диск и другие периферийные устройства). В последнем случае на нее, как и на реальный компьютер, можно устанавливать операционные системы и запускать пользовательские приложения.

На достаточно производительном аппаратном обеспечении скорость таких имитаторов приближается к скорости работы реального устройства. В связи с этим их часто используют с целью тестирования и отладки ПО, создания "песочниц" (специально выделенной и изолированной среды запуска приложений) или запуска приложений, написанных под другую платформу.

Важным недостатком таких систем является низкая функциональная точность (fidelity) и ограниченность доступных устройств.

## COTSon

COTSon - имитатор высокопроизводительных вычислительных комплексов, разработанный HP Labs и AMD. Целью разработки являлась быстрая и точная оценка производительности вычислительных систем, покрывая полный стек технологий и аппаратных моделей. Имитатор способен симулировать как отдельные многоядерные компьютеры, так и масштабируемые кластеры из сотен отдельных устройств, соединённых по сети. COTSon - гибкая платформа, поддерживающая подключение пользовательских модулей. Функциональный эмулятор основан на эмуляторе SimNow (от AMD) и поддерживает архитектуры x86 и x86\_64.

COTSon работает в парадигме functional-directed: быстрые функциональные симуляторы работают вместе с моделями синхронизации, обеспечивая высокую скорость и точность работы.

Разработчики COTSon преследовали следующие цели:

* Модульная архитектура с раздельными функциональным и синхронизационным модулями.
* Полная эмуляция системы, позволяющая запустить на эмуляторе полный стек программного обеспечения, включая пользовательские приложения.
* Высокая скорость работы.
* Возможность жертвовать скоростью работы ради точности, и наоборот.
* Постоянная коммуникация между функциональным эмулятором и эмулятором синхронизации.
* Масштабируемость и поддержка имитации кластеров.

Схема архитектуры COTSon представлена в Приложении 2.

## Manifold

Manifold - инфраструктура для построения имитационных моделей с варьирующимися точностью и скоростью из набора готовых компонентов. Вдобавок к функциональным и синхронизационным моделям, он также поддерживает моделирование физических свойств современных процессоров, таких как потребление энергии или температура.

Основными отличительными особенностями Manifold являются:

* Прозрачное параллельное исполнение
* Включение моделей потребления энергии, температуры, надежности и др.
* Полная эмуляция системы
* Архитектура, основанная на композиции независимых компонентов.

Пользуясь набором компонентов, пользователи могут самостоятельно строить модели требуемых устройств и проводить параллельные симуляции для получения любых типов данных. При необходимости пользователи могут писать собственные модули и компоненты.

Схема архитектуры Manifold представлена в Приложении 2.

## Gem5

Имитатор gem5 – модульная платформа для исследования архитектуры компьютерных систем, включающая как архитектуру уровня системы, так и микроархитектуру процессора.

Основные особенности:

* Взаимозаменяемые модели CPU. В gem5 есть четыре подели CPU: простая модель CPU с одной инструкцией за цикл; детализированная модель рабочего процессора; детализированная модель поврежденного процессора; и модель KVM-процессора, использующего виртуализацию для ускорения симуляции.
* Модель GPU NoMali. Эта модель обеспечивает репрезентативность результатов работы экспериментов, связанных с GPU, а также является полностью совместимой с драйверами GPU Linux и Android, убирая необходимость в программной отрисовке.
* Система памяти, основанная на событиях. В gem5 есть детализированная модель памяти, включающая в себя кэши, отслеживание их когерентности и точную модель контроллера DRAM. Эти компоненты можно гибко настраивать и комбинировать для моделирования сложных иерархий памяти.
* Trace-based модель CPU, позволяющая более эффективно моделировать систему памяти CPU
* Гомогенная и гетерогенная многоядерность. Модели CPU и кэшей можно сочетать в любом виде, с любой топологией, не теряя когерентности кешей.
* Поддержка разных наборов инструкций. Gem5 разделяет семантику набора инструкций от модели CPU, позволяя эмулировать множество архитектур.
* Поддержка нескольких систем. В одном процессе эмулирования можно запустить несколько систем, позволяя эмулировать клиент-серверные сети.
* Моделирование потребления питания.

## Другие имитаторы

* Flexus предназначен для быстрой и точной полной эмуляции системы при помощи статистического семплирования. С помощью сохранения состояний и запуска только необходимых частей модели, эмулятор позволяет достичь высокой скорости работы.
* GEMS был создан в первую очередь для эмуляции памяти – кэшей, топологии, взаимосвязанности, протоколов синхронизации – с высоким уровнем гибкости. Он использует SLICC, специализированный язык для реализации протоколов синхронизации.
* MARSS и PTLsim были созданы для быстрой имитации систем x86 путем тесной взаимосвязи с гипервизором, виртуальной машиной или имитатором. Виртуальная машина может переключаться между использованием CPU системы-хозяина и эмулированного CPU, позволяя эмулировать самые интересные части системы с полной детализацией, а остальные прогнать в почти реальном времени.
* OVPsim и Simics предназначены для моделирования компонентов операционных систем и для быстрой функциональной симуляции. Впрочем, для моделирования высокодетализированных систем, им все еще требуются дополнительные технологии симуляции.

# Заключение

Поставленная задача решена полностью. Были рассмотрены распространенные используемые имитаторы. Были сформулированы основные задачи и проблемы, возникающие при имитационном моделировании высокопроизводительных вычислительных комплексов. При необходимости принять решение о выборе того или иного фреймворка для имитационного моделирования, можно руководствоваться данной работой как обзором существующих решений.

# Список используемых источников

**Georg Hager, Gerhard Wellein**, Introduction to High Performance Computing for Scientists and Engineers [Текст] / July 2, 2010 by CRC Press : Reference – 356 Pages – 143 B/W Illustrations – ISBN 9781439811924 .

**Eduardo Argollo, Ayose Falcón, Paolo Faraboschi, Matteo Monchiero, Daniel Ortega**, COTSon: Infrastructure for Full System Simulation [Текст] / HP Laboratories HPL-2008-189.

**R. Bedicheck**. SimNow: Fast platform simulation purely in software [Текст] / Hot Chips – 16, Авг. 2004.

**F. Bellard**. QEMU, a fast and portable dynamic translator. [Текст] / USENIX 2005 Annual Technical Conf., – FREENIX Track – стр 41–46 – Апр. 2005.

**C. J. Mauer, M. D. Hill, D. A. Wood.** Full-system timing-first simulation. [Текст] / SIGMETRICS ’02: Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems – стр 108–116 – New York, NY, USA, 2002. ACM.

**M. Rosenblum, S. A. Herrod, E. Witchel, A. Gupta**. Complete computer system simulation: The SimOS approach [Текст] / IEEE Parallel Distrib. Technol..

**Jun Wang, Jesse Beu, Rishiraj Bheda, Tom Conte и др**. Manifold: A Parallel Simulation Framework for Multicore Systems [Текст] / Technical Report GIT-CERCS-14-07 : School of Electrical and Computer Engineering,Georgia Institute of Technology, Atlanta, USA : Май 30, 2014.

**Jakob Engblom**. Full-System Simulation [Текст] / Extended Abstract : ESSES 2003.

**Michael Flynn**. Some Computer Organizations and Their Effectiveness [Текст] / IEEE Transactions on Computers. – том c-21, No.9 – Сентябрь 1972.

**Antonio Alfonso-Faus**. Fundamental Principle of Information-to-Energy Conversion [Текст] / Department of Aerotecnia, Escuela de Ingeniería Aeronáutica y del Espacio : Madrid Technical University.

**Anthony Gutierrez Joseph Pusdesris, Ronald G. Dreslinski Trevor Mudge**. Sources of Error in Full-System Simulation [Текст] / Advanced Computer Architecture Laboratory University of Michigan : Ann Arbor, MI.

**R. Landauer**. Irreversibility and Heat Generation in the Computing Process [Текст].

**Krste Asanovic, Rastislav Bodik, и др**. A View of the Parallel Computing Landscape [Текст].

**Fabrice Bellard**. QEMU, a Fast and Portable Dynamic Translator [Текст] / USENIX Association FREENIX Track: 2005 USENIX Annual Technical Conference.

**Mendel Rosenblum, Stephen A Herrod, Emmett Witchel, Anoop Gupta**. Complete Computer System Simulation: The SimOSApproach [Текст] / Stanford University.

**Shivani Raghav, Martino Ruggiero, David Atienza**. Scalable Instruction Set Simulator for Thousand-core Architectures Running on GPGPUs [Текст] / DEIS - University of Bologna, Bologna, IT.

The gem5 Simulator: A modular platform for computer-system architecture research [Электронный ресурс] / [www.m5sim.org](http://www.m5sim.org) .

COTSon: infrastructure for full system simulation [Электронный ресурс] / cotson.sourceforge.net.

The end of Dennard scaling [Электронный ресурс] / [cartesianproduct.wordpress.com/2013/04/15/the-end-of-dennard-scaling/](https://cartesianproduct.wordpress.com/2013/04/15/the-end-of-dennard-scaling/).

PARSEC [Электронный ресурс] / parsec.cs.princeton.edu/.

# Приложение 1. Пример выходных данных бенчмарка PARSEC

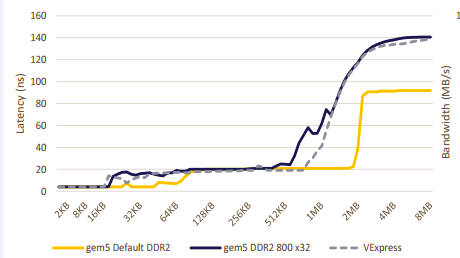


Рисунок 1 – Результаты измерения задержки доступа к памяти

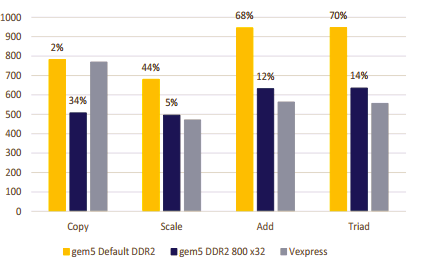


Рисунок 2 – Результаты измерения производительности памяти

# Приложение 2. Схемы архитектуры эмуляторов gem5 и COTSon.

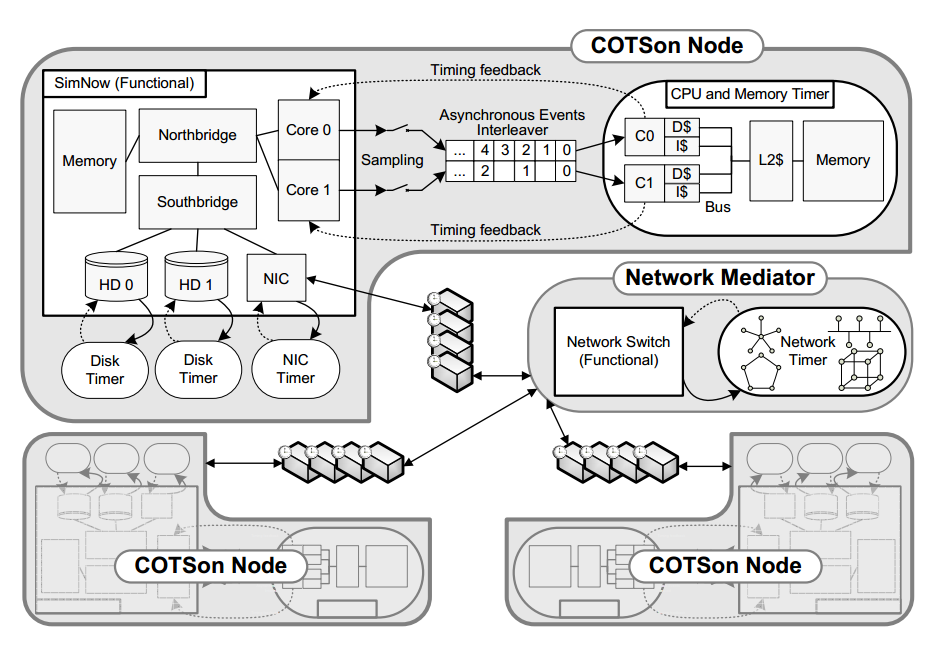


Рисунок 3 – Схема архитектуры COTSon

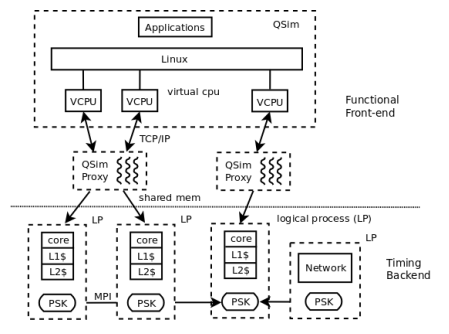


Рисунок 4 – Схема архитектуры Manifold