УТВЕРЖДАЮ

Директор Федерального
государственного учреждения
«Федеральный исследовательский
центр «Информатика и управление»
Российской академии наук»,
академик РАН

Соколов И.А.

«29» 04 2021 г.

О Т З Ы В

ведущей организации на диссертационную работу
Андрианова Павла Сергеевича на тему
«Анализ корректности синхронизации компонентов ядра операционных систем»,
представленную на соискание ученой степени кандидата физико-математических наук по специальности 05.13.11 (математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей)

Представленная работа посвящена актуальной задаче — верификации многопоточных программ, в частности, компонентов ядра операционных систем.

Поиск ошибок, связанных с некорректной синхронизацией, является сложной и трудоемкой задачей. При верификации системного программного обеспечения, в частности, ядер операционных систем, возникают дополнительные сложности из-за его специфики. Кроме того, объем исходного кода обычно составляет более 100 000 строк кода, что приводит к дополнительным требованиям по эффективности применяемых методов верификации.

В данной работе предложен метод верификации многопоточных программ, который масштабируется на большие объемы исходного кода. Он основан на подходе с раздельным рассмотрением потоков (thread-modular approach). В рамках данного подхода строится некоторая абстракция над взаимодействием потоков, которая называется окружением потоков. И дальнейший анализ потоков происходит независимо друг от друга, но

совместно с построенным окружением, которое моделирует это взаимодействие. Это позволяет повысить эффективность анализа.

Диссертационная работа состоит из введения, четырех глав, заключения, списка литературы и двух приложений.

Введение содержит описание актуальности работы, определение целей и задач исследования, излагается научная новизна и практическая значимость.

В **первой** главе представлен обзор существующих работ по верификации многопоточных программ. После краткого описания основных терминов приводятся описание существующих методов динамического анализа, статического анализа и статической верификации. Среди методов динамического анализа были рассмотрены подходы на основе векторных часов, статико-динамические методы, специализированные методы, нацеленные на ядра операционных систем. В методах быстрого статического анализа также рассмотрены варианты общецелевых подходов для поиска ошибок в многопоточных программах и специализированных методов, нацеленных на системное программное обеспечение. Среди методов статической верификации были выделены подходы на основе чередований потоков, подходы на основе трансляции, подходы на основе раздельного рассмотрения потоков. Основные итоги обзора содержат выводы в том числе о том, что в данный момент отсутствуют методы статической верификации, способные эффективно применяться к большим объемам системного программного кода. И именно они позволяют обеспечить наибольшее качество анализа программы.

Вторая глава посвящена теоретическому описанию разработанного метода. В ней даны формальные определения программы: конкретных состояний и отношения переходов над ними, а также задана семантика языка. После этого определяется адаптивный статический анализ (англ. Configurable Program Analysis, CPA) с абстрактными переходами, который является расширением классической теории адаптивного статического анализа над состояниями. Расширенная теория CPA позволяет определить основной алгоритм построения достижимых переходов и сформулировать основную теорему корректности (англ. soundness). Эта теорема позволяет гарантировать, что алгоритм строит множество достижимых абстрактных переходов таким образом, что не пропускает ошибочный переход.

Расширенная версия теории позволяет описать подход с раздельным рассмотрением потоков. Сначала приводится его основная идея на примере, а затем вводится формальный объект CPA, который реализует всю функциональность, в том числе приводится описание алгоритма построения

окружения потока. Отдельно доказывается, что заданный таким образом CPA удовлетворяет условию теоремы о корректности, то есть, позволяет гарантировать отсутствие пропуска ошибок в заданных предположениях. Далее описываются различные краевые случаи: анализ без абстракции, анализ, инвариантный к эффектам окружения. После этого приводятся примеры различных вариантов анализа, то есть, CPA: анализ композиции, анализ предикатов, анализ явных значений и др. Для каждого из них показывается, что он удовлетворяет условиям теоремы.

Третья глава описывает реализацию метода поиска состояния гонки в инструменте CPALockator. Сначала описывается устройство инфраструктуры CPAchecker, затем приводится одна из возможных конфигураций инструмента для поиска состояний гонки. Далее подробно описываются различные технические аспекты реализации используемых CPA-анализаторов, в том числе оптимизации для повышения эффективности. Кроме того, приводятся описания различных CPA, которые не были приведены в теоретической части работы, что, вероятно, сказывается на формальной корректности инструмента, тем не менее, значительно повышают скорость и точность работы. Кроме самих CPA, в разделе описаны технические детали алгоритма уточнения абстракции, алгоритма поиска состояний гонки по завершению построения абстракции, а также дополнительный преданализ для поиска разделяемой памяти. Кратко описан процесс визуализации результатов, то есть, найденных состояний гонки, а также используемые оптимизации.

В **четвертой** главе приведены результаты экспериментальной оценки инструмента. Сравнение различных конфигураций и режимов работы проводилось на нескольких множествах разнородных задач: множестве задач SV-COMP, множестве драйверов и двух ядрах ОС. Первый набор задач содержит небольшие рукописные тесты, второй набор содержит задачи из нескольких потоков и около 10 000 строк кода, а ядра операционных систем содержат десятки и даже сотни потоков и более 100 000 строк исходного кода. Инструмент исследовался в двух режимах работы: решение задачи достижимости в многопоточной программе и поиск состояний гонки. Проведенные эксперименты позволяют не только оценить эффективность той или иной оптимизации, но и составить определенный набор рекомендаций, для какого целевого кода некоторая оптимизация будет иметь смысл. Кроме оценки эффективности инструмента в данной главе приведены результаты анализа причин ложных срабатываний. Лишь в четверти случаев проблема заключается в неточности предложенного метода. Проведенный далее анализ

причин возможного пропуска ошибок показал, что инструмент позволяет не пропустить ошибку при определенных предположениях.

В **заключении** перечислены основные результаты.

В **приложении А** приведены доказательства теоретических утверждений. В **приложении Б** приведено описание изменений, содержащих исправления ошибок.

Диссертационная работа выполнена на достаточно высоком уровне. Научную новизну содержат следующие выносимые на защиту положения диссертации:

1. Метод поиска состояний гонки на основе раздельного анализа потоков, использующий средства абстракции состояний и переходов для управления точностью и ресурсоемкостью верификации.

2. Алгоритм построения окружения потока, который позволяет гибко настраивать уровень абстракции над взаимодействием потоков и доказана его корректность.

3. Новый алгоритм, который является обобщением существующего алгоритма статической верификации программ при помощи метода CPA, расширяющий типовой набор CPA-анализаторов средствами верификации многопоточных программ с раздельным анализом потоков, и доказана его корректность.

Результаты диссертации в полной мере отражены в публикациях автора и апробированы на международных и российских конференциях и научных семинарах.

Автореферат содержит в краткой форме все основные результаты, полученные в диссертации, и соответствует содержанию диссертации.

Результаты работы могут быть использованы в организациях и институтах, занимающихся исследованиями в области разработки операционных систем и драйверов для них.

По тексту представленной работы имеются следующие **замечания**.

1. При анализе возможности пропуска ошибок в модулях ядра ОС Linux остается непонятным, в чем причина трех таймаутов. Поможет ли увеличение лимитов или проблема является принципиальной?

2. Представлено описание реализации оптимизаций для алгоритма поиска гонки (раздел 3.13.3), можно было бы привести результаты экспериментального подтверждения ее эффективности.

3. При описании основного алгоритма вычисления множества достижимых состояний (раздел 2.3) не сказано о его завершаемости.

4. В описании теоретической модели программы (раздел 2.1) отсутствует оператор вызова функции, тем не менее, далее упоминается CallstackCPA, нужный исключительно для обработки таких операторов.

Указанные недостатки не являются существенными и не снижают ценности диссертационной работы.

Диссертация является законченной научно-квалификационной работой, решающей актуальную задачу и удовлетворяющей критериям «Положения о порядке присуждения ученых степеней» и всем требованиям, предъявляемым ВАК к кандидатским диссертациям по специальности 05.13.11 - «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей».

Автор диссертации Андрианов Павел Сергеевич заслуживает присуждения ему ученой степени кандидата физико-математических наук.

Отзыв на диссертацию обсужден и одобрен на заседании семинара отделения №6 ФИЦ ИУ РАН «Стохастические и интеллектуальные методы и средства моделирования и построения систем с интенсивным использованием данных» 27 апреля 2021 года (Протокол №4).

Руководитель Отделения ФИЦ ИУ РАН, д.ф.-м.н.

Синицин В.И.

«27» 04 2021 г.