АННОТАЦИЯ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Наименование темы: Оптимизация ПО с открытым исходным кодом “gVisor” при помощи использования разделяемой памяти для перехвата системных вызовов

Выполнена студенткой Кокуниной Наталией Викторовной

Факультет информационных технологий, Новосибирский государственный университет

Кафедра систем информатики Группа 19205

Образовательная программа 09.04.01 Информатика и вычислительная техника.

09.04.01 Информатика и вычислительная техника. Кафедра систем информатики

Объем работы: \_\_\_\_\_\_\_ страниц

Количество иллюстраций: \_\_\_\_\_\_

Количество таблиц: \_\_\_\_\_\_

Количество литературных источников: \_\_\_\_\_

Количество приложений: \_\_\_\_

Ключевые слова: исполняемая среда, контейнеризация, безопасность, оптимизация, разделяемая память

**Объект исследования:**

Механизм перехвата и исполнения системных вызовов в исполняемой среде для контейнеров gVisor

**Задача:**

Разработать и реализовать новый модуль с механизмом перехвата системных вызовов с использованием разделяемой памяти в gVisor

**Цель:**Повышение производительности выполнения системных вызовов в gVisor

**Актуальность:**gVisor – ПО с открытым исходным кодом для увеличения безопасности контейнера. Проблема безопасности контейнеров стала актуальна с развитием и ростом популярности систем контейнеризации. Основное предназначение gVisor – создание из контейнера “песочницы”, в которой можно было бы безопасно выполнять приложения. Это достигается с помощью перехватов системных вызовов и исполнения их в пользовательском пространстве. Поскольку перехват реализован с помощью ptrace, возникают огромные накладные расходы на переключение контекста процессов, что приводит к снижению производительности приложений в контейнере.

**Методы исследования:**Проведен анализ существующего механизма перехвата системных вызовов в gVisor на основе ptrace. Для оптимизации было решено написать механизм, использующий разделяемую память. Модуль написан на Golang с использованием API системных вызовов Linux и механизма разделяемой памяти; для эмуляции системных вызовов будет использоваться возможность ядра Linux “Syscall User Dispatch”

**Результаты:**

Был спроектирован и внедрен новый модуль в gVisor. Были проведены тесты, которые показывают, что цель достигнута – улучшение производительности выполнения системных вызовов в 1.5 раза.

**Область применения:**Системы контейнеризации для развертывания приложений, для уменьшения вероятности взлома контейнеризированного приложения без больших потерь в производительности.

\_\_\_\_\_\_\_\_\_\_\_\_\_ФИО\_\_\_\_\_\_\_\_\_ / ………………. \_\_.\_\_.20\_\_ г.