Министерство образования и науки Российской Федерации
Московский физико-технический институт
(национальный исследовательский университет)

Физтех-школа радиотехники и компьютерных технологий
Кафедра микропроцессорных технологий в телекоммуникационных сетях и вычислительных системах

Выпускная квалификационная работа бакалавра по направлению 03.03.01
«Прикладные математика и физика»

Исследование и разработка системы анализа кода для повышения производительности программ на языке программирования высокого уровня

Студентка Б01-008 группы
Лирисман К. С.

Научный руководитель
Гаврин Е. А.

Долгопрудный
2024

Содержание

Аннотация

В настоящий момент активно развивается статически типизированный управляемый язык программирования, являющийся расширенной и более быстрой версией языка TypeScript (далее TS). Основная идея разработки спецификации и компилятора этого языка программирования — сделать его максимально похожим на TS для упрощения перехода будущих разработчиков между TS и выбранным для исследования языком, а также для ускорения переписывания существующих на TS приложений. Таким образом, между целевым языком и TS формируется общая часть, - корректная с точки зрения TS. Оставшуюся часть называют не поддерживаемой TS.

В данной работе предлагается выделить из целевого ЯП подмножество, наиболее выгодное с точки зрения производительности, в том числе за счёт отсечения удобного для разработчиков, но медленного функционала. Таким образом, пока сам язык неминуемо разивается в сторону общности с TS‘ом ради легкости перехода, предложенная система помогает держать фокус на производительности.

Анализ происходит в момент компиляции исходного кода и предлагает включение желаемых проверок группами или по отдельности путем добавления флагов компиляции. Реализовано 8 пунктов проверки, опирающихся на спецификацию выбранного языка программирования высокого уровня и реализацию его компилятора. К общим с TS проверкам относятся: неявная упаковка и распаковка, ускорение проверок равенства, запрет инструкций верхнего уровня, исключая классы и функции, предложение установки модификатора для класса или метода как финального и другие. Предлагается использование двух режимов работы проверяющей системы — в состоянии предупреждений, а именно, предложений, не обязывающих разработчика к исправлением замечаний и не влияющих на результат работы программы, и в состоянии, приводящему к ошибке при ненулевом количестве предложений, ожидающих от пользователя последующих исправлений, и считающегося за ошибку компиляции. Предложенная система имеет некоторые варианты для повышения производительности. Однако существуют сценарии использования, когда разработчикам необходимо отключение этих проверок. Разработчики могут активировать или деактивировать любую опцию по одной или включать группами. Также можно снять с проверки определенные строки или целые частей кода с помощью предложенного аналога системы точечного отключения проверок Clang Tidy непосредственно в исходном коде программы в виде многострочных или однострочных комментариев. Таким образом, данная система значительно повышает

скорость работы и время запуска приложения на выбранном статически типизированном языке высокого уровня, позволяет разработчикам простыми методами улучшить их код и потенциально избежать некоторых ошибок.

Глава 1

Введение

Здесь идет текст. Вот так выглядит ссылка на библиографиюю. Аналогично добавляются еще главы, внутри них можно объявлять секции с помощью `\section`.

Глава 2

Постановка задачи

Цели работы:

1. Разработка системы анализа и предупреждений исходного кода выбранного языка программирования высокого уровня для повышения его производительности и ускорения запуска написанных на нем приложений
2. Реализация системы-аналога Clang Tidy для выборочного отключения выбранных проверок точечно, непосредственно в исходном коде.

Задачи работы:

1. Изучение существующих решений
2. Разработка системы анализа кода на этапе компиляции
3. Анализ текущей спецификация целевого языка программирования на предмет потенциально медленных языковых конструкций и функционала, сбор данных для дальнейшего тестирования.
4. Предоставить вариант исправления, ускорящий работу приложения, корректный с точки зрения выбранного языка программироания, для каждой языковой единицы среди предложенных
5. Протестировать каждое предложение: замерить скорость работы приложения до и после предложенных исправлений
6. Реализовать соответствующую поверку в системе анализа после подтверждения положительных результатов тестирования
7. Разработать систему точечного и группового отключения выбранных разработчиком проверок, поддержать наиболее популярные сценарии использования, опираясь на данные, полученные при изучении существующих решений
8. Поддержать возможность анализа в системе многофайловой сборки

Цели работы разумно считать достигнутыми при выявлении и реализации не менее пяти предложений, ускоряющих работу приложения в среднем не менее, чем на 5 %, а также разработке системы отключения проверок, успешной поддержке проектной сборки и прохождении тестирования, составленного из некоторых потенциальных сценариев применения предложенных решений.

Глава 3

Обзор существующих решений

и снова текст.

Глава 4

Теоретическая часть

так тоже неплохо.

Глава 5

Практическая часть

практикуем.

Глава 6

Заключение

заключаем

Литература

Будет добавлена.