

Тестирование конкурентных структур
данных на соответствие моделям
согласованности

Морковкин Василий

2020

Мотивация

Чтение данных с записывающего устройства, проведение сетевых запросов и одновременная обработка поступающих команд - одни из задач, стоящих перед программным обеспечением. Результатом оптимизации затрат процессорного времени на решение этих задач стало появление ряда техник *конкурентного программирования*. Однако необходимость использования ресурсов, не поддерживающих одновременный доступ, зависимость от системного планировщика задач, а также особенности устройства процессорных кешей могут приводить к неопределенным или нежелательным исполнениям компьютерных программ. Целью конкурентного программирования является предоставление механизмов и методик, которые бы позволили писать корректный, обладающий определенным поведением код. Одним из важнейших механизмов являются *структуры данных*, пригодные для использования в конкурентной среде. Для описания их свойств и предоставляемых гарантий был разработан ряд *моделей согласованности*.

Проверка алгоритмов на соответствие этим моделям может быть осуществлена с помощью методов формальной верификации. Однако для проверки конкретных их реализаций на языках программирования доступно лишь тестирование с перебором некоторого подмножества всех сценариев исполнения. В этой работе будут рассмотрены существующие модели согласованности конкурентных структур данных, разработаны варианты их тестирования, и, наконец, реализованы в виде библиотеки на языке программирования Scala.

Оглавление

1	Введение	4
1.1	Основные понятия	4
1.2	Основные понятия	5
1.3	Цели работы	5
1.4	План работы	5
2	Конкурентное программирование	6
2.1	Проблемы	6
2.1.1	Состояние гонки	6
2.1.2	Гонка на данных	6
2.1.3	Взаимная блокировка	6
2.2	Модели согласованности	6
2.2.1	Последовательная согласованность	6
2.2.2	Линеаризуемость	6
2.2.3	Ослабленная линеаризуемость	6
3	Тестирование моделей согласованности	7
3.1	Алгоритмическая сложность тестирования	7
3.2	Алгоритмы тестирования	7
3.2.1	Последовательная согласованность	7
3.2.2	Линеаризуемость	7
3.2.3	Ослабленная линеаризуемость	7

3.3	Разработка требований к библиотеке	7
3.3.1	Интерфейс	7
3.3.2	Конфигурация	7
4	Реализация библиотеки	8
4.1	Интерфейс	8
4.2	Генерация тестовых сценариев	8
4.3	Запуск тестовых сценариев	8
4.4	Валидация результатов исполнения	8
4.5	Генерация отчета тестирования	8
5	Выводы	9
6	Литература	10

Введение

1.1 Основные понятия

Введем основные понятия, необходимые в дальнейшем.

Операция — действие, состоящее из некоторого числа программных инструкций. Под этим словом будут подразумеваться отдельные взаимодействия со структурой данных (вставка, удаление, поиск и т.д.).

Процесс — последовательность операций, исполняемая любой единицей исполнения (системным процессом, системной нитью, нитью виртуальной машины, легковесной нитью и т.д.)

Структура данных — формат хранения и управления данным, предоставляющий интерфейс для доступа к ним и их изменения.

Конкурентные операции — две или более операций, временные интервалы исполнения которых пересекаются.

Конкурентные процессы — процессы, исполняющие конкурентные операции.

Конкурентная структура данных — структура данных, подчиняющаяся законам некоторой модели согласованности. Иными словами, структура данных, пригодная для использования конкурентными процессами.

Модель согласованности — набор гарантий о предсказуемости результатов чтения, записи и изменения данных, позволяющий рассуждать о поведении компьютерной программы.

1.2 Основные понятия

1.3 Цели работы

1.4 План работы

Конкурентное программирование

2.1 Проблемы

2.1.1 Состояние гонки

2.1.2 Гонка на данных

2.1.3 Взаимная блокировка

2.2 Модели согласованности

2.2.1 Последовательная согласованность

2.2.2 Линеаризуемость

2.2.3 Ослабленная линеаризуемость

Тестирование моделей согласованности

3.1 Алгоритмическая сложность тестирования

3.2 Алгоритмы тестирования

3.2.1 Последовательная согласованность

3.2.2 Линеаризуемость

3.2.3 Ослабленная линеаризуемость

3.3 Разработка требований к библиотеке

3.3.1 Интерфейс

3.3.2 Конфигурация

Реализация библиотеки

4.1 Интерфейс

4.2 Генерация тестовых сценариев

4.3 Запуск тестовых сценариев

4.4 Валидация результатов исполнения

4.5 Генерация отчета тестирования

Выводы

Литература