

ПРЕДСТАВЛЕНИЕ ДАННЫХ ТРАССЫ ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОВ В ВИДЕ ТАБЛИЦ ДЛЯ РАБОТЫ С БАЗОЙ ДАННЫХ SQLite

Логинова И. П.

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: irilog@mail.ru

При проведении параллельных вычислений в системе логической оптимизации функционально-структурных описаний цифровых устройств FLC2 формируются файлы трасс, содержащие набор определенных событий. Посредством средств, предоставляемых библиотеками платформы Qt, осуществляется доступ к данным трассы параллельных процессов, из которых выбирают наборы событий для аппроксимации, филь- трования и визуализации. Для хранения информации о событиях, необходимой для анализа и последующей оценки эффективности параллельных вычислений, используется сериализация трасс в базе данных SQLite.

ВВЕДЕНИЕ

Анализ трассы параллельного приложения проводится: с учетом количества событий в трассе; временного интервала выполнения приложения; числа процессов параллельного приложения; текущих настроек прикладных программ оптимизации, используемых в различных проектных процедурах. После чтения данных о событиях из файла трассы, их сортировки и фильтрации, осуществляется визуализация трассы параллельных процессов путем формирования графического представления. В основу реализации параллельных вычислений в системе FLC2 положена организация параллельных процессов. Каждый процесс выполняется на ядре многопроцессорной вычислительной системы (ВС) [3]. В процессах производится запуск прикладных программ с помощью команды, аргументы которой определяют имя программы и исходный объект. Исходные входные объекты в параллельных процессах являются независимыми. Организовать совокупности независимых объектов для проведения параллельных вычислений с использованием программ оптимизации позволяет среда системы FLC2, в которой возможно проведение проектных операций для преобразования исходного объекта в одноуровневую сеть независимых элементов, представляющих входные объекты для операций, выполняемых в отдельных процессах. Таким образом, параллельные вычисления в системе FLC2 реализуются посредством запуска параллельных процессов, в каждом из которых выполняется программа оптимизации для одного элемента сети [4].

1. КОНЦЕПЦИЯ «МОДЕЛЬ/ПРЕДСТАВЛЕНИЕ» ДЛЯ ОБРАБОТКИ ДАННЫХ ТРАССЫ

Для организации сложных комплексных приложений платформа Qt применяет архитектуру View/Model (Представление/Модель), которая является модификацией паттерна MVC. Qt содержит набор классов, которые для управ-

ления связями между данными и способами их представления используют архитектуру «Модель/Представление» (M/V). Такое разделение позволяет отображать одни и те же данные в нескольких разных представлениях и реализовывать новые типы представлений без изменения базовых структур данных. На рис. 1 представлена архитектура, который использует одну Модель (источник данных) для одновременного представления данных из нее двумя способами: в виде таблицы и в виде горизонтальных диаграмм с накоплением. Согласно данной схеме всю работу по обновлению данных между источником данных и Моделью берет на себя Представление.

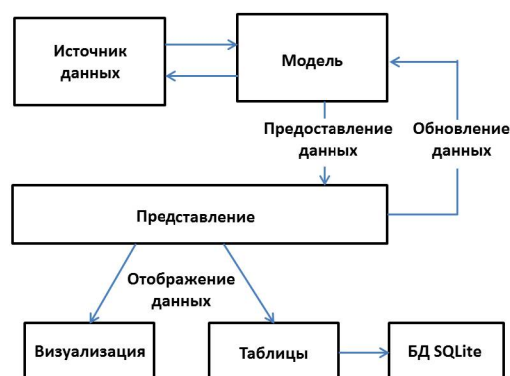


Рис. 1 – Архитектура M/V при работе с трассой параллельных процессов

Процесс работы приложения в этой архитектуре выглядит следующим образом. Модель взаимодействует с источником данных (трасса параллельных процессов), обеспечивая интерфейс для других компонентов архитектуры. Представление через модель может взаимодействовать с данными, в частности, может получать данные и отображать их. Для отображения в представлении применяется программа визуализации [3]. Когда данные изменяются, программа визуализации напрямую взаимодействует с моделью, уведомляя ее об изменении данных.

II. ПРЕДСТАВЛЕНИЕ ДАННЫХ ТРАССЫ ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОВ В ВИДЕ ТАБЛИЦЫ

Платформа Qt предоставляет набор классов, которые реализуют архитектуру M/V для управления взаимосвязями между данными и различными представлениями. Архитектура обеспечивает гибкость настройки представления и позволяет комбинировать источники данных с широким спектром представлений. Базовым классом всех моделей является класс `QAbstractItemModel`. Этот класс определяет интерфейс, который используется для доступа к данным. При необходимости представления данных в виде таблицы при реализации модели, как правило, используется класс `QAbstractTableModel`. Представления (Views) реализуются объектами классов, унаследованных от абстрактного класса `QAbstractItemView`. Например, в программе визуализации [3] объект класса `QTableView` отображает данные из модели в виде таблицы.

На рисунках 2–4 приведены представления трассы параллельных процессов, полученной в результате проведения логической оптимизации в системе FLC2 для 26-ти объектов одноуровневой сети, выполненных на 8-ми ядрах процессора.

```
Core#8, 1 Projects and 1 Programs : ForEsp
thread#1, numTask=13, project: TimeBeginTask=0.001000, TimeTask=0.664000, TimeAfterTask=0.665000
thread#1, numTask=25, project: TimeBeginTask=0.666000, TimeTask=0.379000, TimeAfterTask=1.045000
thread#2, numTask=12, project: TimeBeginTask=0.004000, TimeTask=0.230000, TimeAfterTask=0.234000
thread#2, numTask=7, project: TimeBeginTask=0.236000, TimeTask=0.204000, TimeAfterTask=0.440000
thread#2, numTask=3, project: TimeBeginTask=0.442000, TimeTask=0.182000, TimeAfterTask=0.624000
thread#2, numTask=23, project: TimeBeginTask=0.625000, TimeTask=0.208000, TimeAfterTask=0.833000
thread#2, numTask=9, project: TimeBeginTask=0.833000, TimeTask=0.215000, TimeAfterTask=1.048000
thread#2, numTask=5, project: TimeBeginTask=1.048000, TimeTask=0.219000, TimeAfterTask=1.267000
thread#2, numTask=19, project: TimeBeginTask=1.267000, TimeTask=0.245000, TimeAfterTask=1.512000
thread#2, numTask=26, project: TimeBeginTask=1.512000, TimeTask=0.325000, TimeAfterTask=1.837000
thread#2, numTask=10, project: TimeBeginTask=1.837000, TimeTask=0.324000, TimeAfterTask=2.161000
thread#2, numTask=8, project: TimeBeginTask=2.161000, TimeTask=0.170000, TimeAfterTask=2.331000
thread#2, numTask=21, project: TimeBeginTask=2.331000, TimeTask=0.501000, TimeAfterTask=2.832000
thread#2, numTask=11, project: TimeBeginTask=2.832000, TimeTask=0.002000, TimeAfterTask=2.834000
thread#2, numTask=5, project: TimeBeginTask=2.834000, TimeTask=0.351000, TimeAfterTask=3.185000
thread#2, numTask=15, project: TimeBeginTask=3.185000, TimeTask=0.004000, TimeAfterTask=3.189000
thread#2, numTask=17, project: TimeBeginTask=3.189000, TimeTask=0.222000, TimeAfterTask=3.411000
thread#2, numTask=4, project: TimeBeginTask=3.411000, TimeTask=0.171000, TimeAfterTask=3.582000
thread#2, numTask=22, project: TimeBeginTask=3.582000, TimeTask=0.615000, TimeAfterTask=4.197000
thread#2, numTask=14, project: TimeBeginTask=4.197000, TimeTask=0.005000, TimeAfterTask=4.202000
thread#2, numTask=1, project: TimeBeginTask=4.202000, TimeTask=0.346000, TimeAfterTask=4.548000
thread#2, numTask=20, project: TimeBeginTask=4.548000, TimeTask=0.526000, TimeAfterTask=5.074000
thread#2, numTask=16, project: TimeBeginTask=5.074000, TimeTask=0.006000, TimeAfterTask=5.080000
thread#2, numTask=23, project: TimeBeginTask=5.080000, TimeTask=0.226000, TimeAfterTask=5.306000
thread#2, numTask=6, project: TimeBeginTask=5.306000, TimeTask=0.238000, TimeAfterTask=5.544000
thread#2, numTask=18, project: TimeBeginTask=5.544000, TimeTask=0.455000, TimeAfterTask=6.000000
thread#2, numTask=24, project: TimeBeginTask=6.000000, TimeTask=0.643000, TimeAfterTask=6.643000
```

Рис. 2 – Фрагмент трассы параллельных процессов

На рис. 3 представление в виде горизонтальной диаграммы с накоплением демонстрирует распределение вычислительной нагрузки по ядрам процессора. На рис. 4 представление в виде таблицы содержит всю информацию трассы с указанием всех событий, начиная от идентификатора процесса, номера задачи, номера ядра и всех временных меток начала, длительности и завершения каждой задачи.

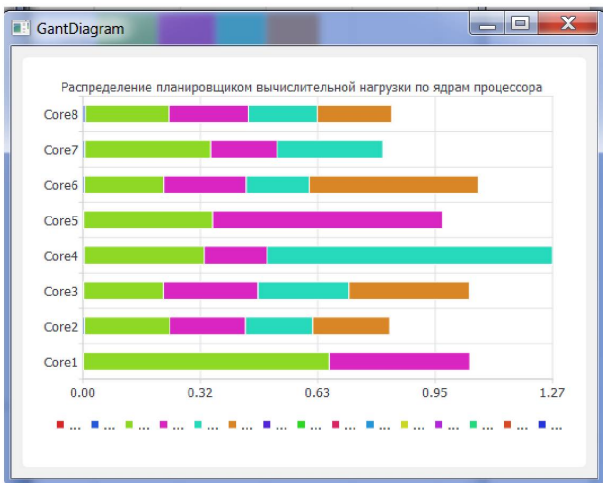


Рис. 3 – Представление временных меток трассы в виде диаграммы Ганта

Trace	Task	Cores	BegTask	TimeTask	AfterTask	Progr
1	13	1	0.001000	0.664000	0.665000	ForEsp
2	25	1	0.666000	0.379000	1.045000	ForEsp
3	12	2	0.004000	0.230000	0.234000	ForEsp
4	7	2	0.236000	0.204000	0.440000	ForEsp
5	3	2	0.442000	0.182000	0.624000	ForEsp
6	23	2	0.625000	0.208000	0.833000	ForEsp
7	9	3	0.833000	0.215000	1.048000	ForEsp
8	5	3	1.048000	0.219000	1.267000	ForEsp
9	19	3	1.267000	0.245000	1.512000	ForEsp
10	26	3	1.512000	0.325000	1.837000	ForEsp
11	10	4	1.837000	0.324000	2.161000	ForEsp
12	8	4	2.161000	0.170000	2.331000	ForEsp
13	21	4	2.331000	0.501000	2.832000	ForEsp
14	11	5	2.832000	0.002000	2.834000	ForEsp
15	2	5	2.834000	0.351000	3.185000	ForEsp
16	15	6	3.185000	0.004000	3.189000	ForEsp
17	17	6	3.189000	0.222000	3.411000	ForEsp
18	4	6	3.411000	0.171000	3.582000	ForEsp
19	22	6	3.582000	0.615000	4.197000	ForEsp
20	14	7	4.197000	0.005000	4.202000	ForEsp
21	1	7	4.202000	0.346000	4.548000	ForEsp
22	20	7	4.548000	0.526000	5.074000	ForEsp
23	16	8	5.074000	0.006000	5.080000	ForEsp
24	6	8	5.080000	0.226000	5.306000	ForEsp
25	18	8	5.306000	0.238000	5.544000	ForEsp
26	24	8	5.544000	0.455000	6.000000	ForEsp
27	24	8	6.000000	0.643000	6.643000	ForEsp

Рис. 4 – Представление трассы в виде таблицы

Данные можно хранить как в модели, так и в базе данных (БД). Qt предлагает несколько моделей, которые используются для доступа к базам данных: `QSqlQueryModel`, `SqlTableModel` и `QSqlRelationalTableModel`. Использование моделей БД и таблиц позволяют организовать сериализацию данных трассы в БД SQLite.

III. СПИСОК ЛИТЕРАТУРЫ

1. Бибило, П. Н. Система логической оптимизации функционально-структурных описаний цифровых устройств на основе продукционно-фреймовой модели представления знаний / П. Н. Бибило, В. И. Романов // Проблемы разработки перспективных микро- и наноэлектронных систем – (МЭС-2020) : IX Юбилейная Всероссийская научно-техническая конференция : Сборник трудов ; под общей ред. А. Л. Стемковского. – Москва : ИПИМ РАН, 2021 Москва : ИПИМ РАН, 2020. – С. 9–16.
2. Бибило, П. Н. Экспериментальное сравнение эффективности программ минимизации систем булевых функций в классе дизъюнктивных нормальных форм / П. Н. Бибило, И. П. Логинова // Информатика. – 2022. – Т. 19, № 2. – С. 26–55.
3. Логинова, И. П. Построение и визуальный анализ параллельных решений в системе логической оптимизации функционально-структурных описаний дискретных схем // Компьютерное проектирование в электронике (EDA Conference 2024) : сб. трудов Междунар. науч.-практ. конф. (Республика Беларусь, г. Минск, 28 ноября 2024 г.) Минск : БГУИР, 2024. – С. 108–111.