

МЕТОДИКА ПРОЕКТИРОВАНИЯ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ

М. К. Буза, О. М. Кондратьева

Кафедра многопроцессорных систем и сетей, Белорусский государственный университет

Минск, Республика Беларусь

E-mail: {bouza, kondratjeva}@bsu.by

В работе рассматривается подход к проектированию параллельных программ для систем с общей памятью (многопоточных программ). Предлагается программный инструмент, который упрощает и ускоряет процесс создания многопоточных приложений. Основу инструмента составляют объектно-ориентированная многопоточная библиотека и инструментальная оболочка проектирования. Приводится методика проектирования параллельных программ с использованием разработанного инструмента.

ВВЕДЕНИЕ

Сложность разработки программного обеспечения возросла с появлением многопроцессорной вычислительной техники. Повсеместное распространение многоядерных процессоров усилило интерес к проектированию многопоточных программ, в частности, к созданию инструментальных систем, облегчающих разработку таких программ.

При разработке параллельной программы важным является вопрос о выборе технологии параллельного программирования. При выборе технологии рекомендуется оценить следующие ее характеристики: возможность создания эффективных программ, возможность быстрого создания параллельных программ и гарантии сохранения эффективности параллельной программы при ее переносе с одного компьютера на другой. В некоторых случаях выбор определяется просто. Например, вполне реальной является ситуация, когда воспользоваться можно только тем, что знает программист или установлено на компьютере.

I. ТЕХНОЛОГИИ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ

На практике существует ряд технологических подходов к программированию многопоточных приложений. Среди них – использование в качестве конструктивных элементов заранее распараллеленных процедур из специализированных библиотек и использование инструментальных систем, облегчающих создание и проектирование параллельных программ. В данной работе предлагается методика проектирования параллельных программ, которая объединяет оба названных технологических подхода. В основе методики – объектно-ориентированная многопоточная библиотека и инструментальная оболочка проектирования.

В настоящее время существует довольно много кроссплатформенных многопоточных библиотек для языка программирования C++. Например, в [1] представлен список из 20 библио-

тек. Относительно новыми в этом списке являются:

- Стандартная библиотека STL (C++11) – новый стандарт языка C++, принятый в 2011 году;
- Parallel Patterns Library (библиотека параллельных шаблонов) – продукт Microsoft, который впервые появился в комплекте с Visual Studio 2010, разработан в стиле стандартной библиотеки C++ и хорошо работает с новым стандартом C++11.

Названные библиотеки являются объектно-ориентированными и позволяют использовать преимущества объектно-ориентированного и обобщенного программирования языка C++ при разработке параллельных приложений. Если в стандартную библиотеку C++11 включены классы для управления потоками, защиты разделяемых данных, синхронизации операций между потоками, низкоуровневые атомарные операции, то мы определяем классы более высокого уровня, например, «Модель делегирования с общей очередью заданий».

II. ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА

Существуют универсальные и полезные элементы, появляющиеся во многих проектах – так называемые образцы (шаблоны, паттерны). Шаблон проектирования в разработке программного обеспечения представляет собой повторяемую архитектурную конструкцию, которая является решением проблемы проектирования в рамках некоторого часто возникающего контекста. Основная польза от использования шаблонов состоит в снижении сложности разработки за счёт готовых абстракций для решения целого класса задач. Концептуально применение шаблонов сродни использованию готовых библиотек кода.

В настоящее время используются различные параллельные паттерны. В [2] приводятся модели создания и функционирования потоков:

- Модель делегирования;
- Модель с равноправными узлами;
- Конвейер;

– Модель «изготовитель-потребитель».

Эти модели представляют собой общие решения по организации многопоточной программы, могут использоваться в качестве образцов и позволяют автоматизировать процесс проектирования. Разработанная объектно-ориентированная библиотека включает классы, соответствующие перечисленным моделям.

В 1995 году Ian Foster впервые предложил методику проектирования параллельных алгоритмов РСАМ (Partitioning, Communication, Agglomeration, Mapping) [3]. Сейчас эта методика вошла в учебники по проектированию параллельных программ и включает этапы [4]:

- Разделение вычислений на независимые части;
- Выделение информационных зависимостей;
- Масштабирование набора подзадач;
- Распределение подзадач между вычислительными элементами.

Инструментальная система поддерживает все этапы разработки параллельной программы.

Инструментальная система «выросла» из экспериментальной среды проектирования параллельных приложений [5], которая использовалась для отладки методики проектирования программ.

III. МЕТОДИКА ПРОЕКТИРОВАНИЯ

Методика разработки параллельной программы с использованием предлагаемого инструмента включает следующие этапы:

1. Определение модели программы;
2. Настройка модели с помощью Мастера;
3. Ручная конкретизация по типу данных;
4. Получение экспериментальных характеристик;
5. Анализ результатов.

На этапе определения модели программы пользователь выбирает модель создания и функционирования потоков в зависимости от решаемой задачи и цели проектирования. В настоящее время реализована поддержка всех моделей, рассмотренных в [2].

При настройке модели с помощью Мастера пользователь уточняет параметры выбранной модели. Например, можно в генерируемый код включить замеры времени. В результате выполнения этого шага строится каркас многопоточно-го приложения.

На этапе ручной конкретизации полученный на предыдущем шаге каркас приложения вручную дорабатывается пользователем. Для более простых задач этот этап может не понадобиться.

На этапе получения экспериментальных характеристик проводятся эксперименты с полученной параллельной программой. Например, с целью определения ее эффективности.

Процесс проектирования имеет итеративную природу. Если в результате анализа эффективности полученной параллельной программы разработчик окажется неудовлетворенным результатом, то он может перепроектировать свою программу, вернувшись на предыдущие этапы. Большинство проблем программирования имеют несколько параллельных решений. Поиск лучшего решения может иметь принципиальное значение для разработчика.

Апробация методики проводится посредством использования предлагаемого инструмента для проектирования наиболее распространенных задач параллельного программирования. Полученные результаты иллюстрируют снижение трудоемкости разработки, а также подтверждают, что абстрагирование позволяет ускорить и улучшить процесс проектирования параллельного кода, при этом несущественно влияет на эффективность исполнения созданных программ. Пока практическая реализация процесса проектирования ограничена достаточно простыми приложениями.

ЗАКЛЮЧЕНИЕ

Проектирование параллельных программ является трудоемким процессом и требует высокой квалификации разработчиков. Это ограничение может быть частично устранено за счет развития специализированного программного инструментария.

Предлагаемая инструментальная система берет на себя рутинную работу, упрощает процесс создания параллельных приложений, ускоряет его и уменьшает количество ошибок.

Разработанная методика проектирования параллельных приложений позволяет эффективно проектировать многопоточные программы для многоядерных процессоров, получать оценки применения различных моделей при решении конкретной задачи. Ее можно также использовать при преподавании соответствующих учебных дисциплин.

1. List of C++ multi-threading libraries [Electronic resource] – Mode of access: https://en.wikipedia.org/wiki/List_of_C%2B%2B_multi-threading_libraries – Date of access: 8.09.2015.
2. Хьюз, К. Параллельное и распределенное программирование с использованием C++ / К. Хьюз, Т. Хьюз // М.: Вильямс, 2004. – 672 с.
3. Foster, I. Designing and Building Parallel Programs: Concepts and Tools for Parallel / I. Foster // Addison Wesley, 1995. – 430 p.
4. Гергель, В. П. Высокопроизводительные вычисления для многопроцессорных многоядерных систем / В. П. Гергель // Издательство МГУ, 2010. – 544 с.
5. Буза, М. К. Экспериментальная среда проектирования параллельных приложений / М. К. Буза, О. М. Кондратьева // Проблемы и перспективы инновационного развития университетского образования и науки: материалы Междунар. науч. конф. – Гродно: ГрГУ, 2015. – С. 90–92.