

УДК 004.4'22

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ГРАФОВЫХ СТРУКТУР

Ключев В.А. (ИТМО)

Научный руководитель – доктор технических наук, профессор Платунов А.Е.
(ИТМО)

Введение. Современные гетерогенные вычислительные системы [1] могут состоять из множества параллельных процессов, работающих на основе различных моделей вычислений, что усложняет их проектирование. Для их описания требуются удобные формальные методы, позволяющие учитывать различные уровни параллелизма, ограничения реального времени [2] и зависимости между процессами.

Графовые структуры являются эффективным инструментом моделирования, так как они позволяют наглядно представлять вычислительные процессы и их взаимодействие. Автоматизированная трансляция графового описания в программный код помогает снизить трудозатраты и повысить корректность и надежность разработки.

Работа посвящена развитию методов представления параллельных вычислительных процессов с использованием графов, а также алгоритмам синтеза программного кода для автоматизации проектирования гетерогенных вычислительных систем.

Основная часть. Параллельные вычислительные системы предлагается описывать с помощью взаимосвязанных графов, которые могут быть представлены на разных уровнях детализации. Такой подход позволяет моделировать сложные структуры, объединяя несколько параллельных процессов в единый граф или, наоборот, детализируя отдельные компоненты системы независимо от дальнейшей аппаратной или программной их реализации. Гибкость этого представления упрощает адаптацию модели под различные сценарии проектирования и анализ вычислительных процессов.

В текущей реализации инструмента поддерживаются две основные модели вычислений: сеть процессов Кана [3] и акторная модель [4]. Сеть процессов Кана организует передачу данных между вычислительными узлами через очереди сообщений, что делает ее удобной для потоковых вычислений и строгого управления зависимостями. В отличие от этого, акторная модель ориентирована на асинхронное взаимодействие между независимыми вычислительными процессами, что делает ее более подходящей для распределенных и масштабируемых систем.

Описание системы в виде графа формируется с использованием специализированного языка, представленного в формате JSON. В этом описании фиксируются вычислительные узлы, связи между ними, параметры выполнения и выбранная модель вычислений. Такое описание затем используется для автоматизированного синтеза программного кода, что значительно упрощает процесс разработки.

На данный момент код генерируется для языка Kotlin, так как он предоставляет удобные инструменты для работы с параллельными процессами и асинхронными операциями, такие как корутины и асинхронные механизмы. В будущем планируется расширить поддержку, добавив новые модели вычислений и целевые языки программирования, что повысит гибкость и расширит область применения разработанного инструмента. Одним из ключевых преимуществ предлагаемого подхода является возможность детализации отдельных компонентов системы, что позволяет более точно анализировать их работу. Например, можно выделить отдельные

вычислительные узлы и исследовать их взаимодействие с другими элементами системы. Это может быть полезно при проектировании сложных распределенных систем, где важно учитывать множество факторов, таких как задержки передачи данных, синхронизация процессов и управление ресурсами. Кроме того, использование графового представления позволяет визуализировать структуру системы, что упрощает понимание ее работы и выявление потенциальных проблем.

Выводы. Предлагаемый подход к описанию параллельных вычислительных систем с помощью графов может являться полезным инструментом для проектирования, анализа и обучения. Он позволяет гибко адаптировать модель под различные задачи, упрощает процесс разработки, избавляя разработчика от необходимости ручной настройки взаимодействия между компонентами системы.

Список использованных источников:

1. Edward A. Lee, Marten Lohstroh, Christian Menard, Soroosh Bateni. Toward a Lingua Franca for Deterministic Concurrent Systems // ACM Transactions on Embedded Computing Systems. - 2021
2. Hermann Kopetz. Real-Time Systems // Design Principles for Distributed Embedded Applications. Springer US. - 2011
3. Edwards, S.A. Kahn Process Networks. In: Languages for Digital Embedded Systems // The Springer International Series in Engineering and Computer Science, vol 572. Springer, Boston, MA. - 2000
4. C. Hewitt, "Actor model of computation: Scalable robust information systems". - 2015