УДК 004.414.23

МЕТОДЫ ИНТЕГРАЦИИ СИМУЛЯТОРОВ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Прошкин Н.А. (Университет ИТМО)
Научный руководитель - к.т.н. Кустарев П.В. (Университет ИТМО)

Введение. Симуляция вычислительных систем (ВС) на функциональном, архитектурном или микроархитектурном уровне – это распространенный метод изучения и анализа алгоритмов, архитектур и реализаций различных ВС путем программной (обычно) имитации работы реальных процессоров и систем во времени.
Достаточно широко представлены и доступны симуляторы “стандартных” платформ, например, PC-совместимых компьютеров. Однако современные программно-аппаратные комплексы могут иметь произвольную организацию, включать в себя несколько разнотипных процессоров, в том числе специализированных, а также различные интерфейсы и устройства ввода-вывода. То есть архитектура такой ВС может быть совершенно уникальной, и при разработке ПО будет необходимо учитывать и использовать ее (архитектуры) особенности, а не абстрагироваться от деталей за счет использования ОС и высокоуровневых языков.
И, наконец, в ситуации, когда установленные сроки разработки не позволяют дождаться изготовления аппаратного прототипа ВС для разработки ПО, использование симуляторов, предоставляющих возможность исполнения и отладки без аппаратуры, становится обязательным.

Основная часть. При компьютерной симуляции специализированных ВС не редки ситуации, когда использования только одного инструмента моделирования недостаточно:

- Симулятор может не поддерживать особенности архитектуры разрабатываемой системы, такие как процессорная гетерогенность. Например, симулятор QEMU обладает очень высокой производительностью. Но он предназначен для моделирования ограниченного перечня классических архитектур, поэтому существуют проекты интеграции QEMU с другими симуляторами, такими как SystemC [1].
- Как правило, в одном симуляторе используются модели одного типа (или функциональные, или микроархитектурные), чего может быть недостаточно. Например, при разработке ВС микроархитектурная модель процессора в большинстве случаев недоступна, но при этом проект может включать в себя специально разработанный аппаратный блок, для которого уже есть описание на уровне регистровых передач (RTL) и внутреннее поведение которого необходимо детально анализировать. В таком случае можно использовать подход, называемый косимуляцией, когда совместно используются модели различного уровня абстракции.

Возможным решением обеих проблем является совместное использование нескольких симуляторов, поэтому в данной работе предлагается разработать средства интеграции уже существующих симуляторов вычислительных систем.

В работе решаются следующие задачи:

- Выбор симуляторов. Необходимо выбрать несколько симуляторов, которые обеспечивают различный уровень точности моделирования. Инструменты должны быть расширяемыми, поддерживать различные процессорные архитектуры и периферийные устройства, а также симуляцию нетиповых архитектур ВС.
- Анализ используемых систем отслеживания времени в выбранных симуляторах и подходов к синхронизации времени между ними. Например, функциональные симуляторы могут отслеживать время по принципу “одна выполненная команда - одна единица времени (например, наносекунда)”. Потактовые симуляторы же используют гораздо более точные модели, учитывающие внутренние детали системы, моделируя

недетерминированное время выполнения разных операций в разных состояниях системы. Нужно понять, как можно синхронизировать настолько разные подходы.

- Анализ подходов к обмену данными между симуляторами. Так как симуляторы представляют собой набор процессов, то ясно, что для организации обмена данными между ними необходимо будет задействовать средства межпроцессного взаимодействия, предоставляемые ОС. Основная сложность здесь заключается в том, что необходимо разработать достаточно гибкий механизм, который позволит передавать данные между разными симуляторами и сможет учитывать при этом различные режимы и интерфейсы обмена между частями моделируемой ВС.
- Анализ существующих библиотек и инструментов для интеграции симуляторов. Необходимо оценить возможности, недостатки и применимость уже существующих инструментов для косимуляции. Например, существует библиотека компонентов от компании Xilinx [2], позволяющая связать между собой симуляторы QEMU, SystemC и потактовые симуляторы, имеющие поддержку SystemC, например Verilator или gem5.

Выводы. В работе решается задача по поиску, исследованию и доработке подходов и методов интеграции существующих симуляторов вычислительных систем. Подготовленные инструменты главным образом будут использоваться при разработке системного ПО для ВС нетиповой архитектуры. В дальнейшем данные методы и средства интеграции могут быть использованы в качестве связующей основы для универсальных инструментов моделирования ВС: они позволят объединять несколько симуляторов для поддержки большего числа процессорных архитектур в одной ВС, а также позволят подключать интерфейсы аппаратных моделей на ПЛИС в качестве составной части косимуляции.

Список использованных источников.

1. Guillaume D., Mark B., Frederic K., Bertrand G., Christophe J. QBox: an industrial solution for virtual platform simulation using QEMU and SystemC TLM-2.0 // 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016). – Toulouse, France, 2016. - 11 с.
2. QEMU Co-simulation – Xilinx Wiki [Электронный ресурс]. – URL: https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/862421112/Co-simulation (дата обращения: 08.01.2024).

Прошкин Н.А. (автор)

Кустарев П.В. (научный руководитель)