УДК 004.4

Увеличение производительности режима прямого исполнения в программном симуляторе

А.А. Кравцов[1], Г.С. Речистов[1], Е.А.Юлюгин[1]

[1]АО «Интел А/О»

Одной из задач программного моделирования является разработка программного обеспечения для еще не вышедшей в производство аппаратуры. В этом случае к модели предъявляются высокие требования к производительности.

В данной работе исследуется полноплатформенный симулятор Wind River® Simics [1] и программная модель вычислительной системы архитектуры Intel® IA-32 [2] на предмет повышения скорости симуляции.

Симулятор Simics имеет три режима работы: интерпретация (самый медленный и простой в реализации), двоичная трансляция и прямое исполнение (самый быстрый) [3]. Для максимально эффективной работы симулятор должен основное время проводить в режиме прямого исполнения (РПИ) и только изредка переключаться на интерпретацию для исполнения неподдерживаемых в аппаратуре и привелигированных инструкций.

Однако есть набор сценариев, на которых наблюдается следующее поведение: рядом расположены несколько инструкций, которые не могут быть исполнены напрямую. В этом случае симулятор на каждой такой инструкции будет выходить из РПИ, интерпретировать эту инструкцию и заходить обратно. Входы в режим прямого исполнения и выходы из него являются медленными, и, когда они встречаются часто, накладные расходы превышают полезную работу. Получается, что режим прямого исполнения в таком случае работает медленнее интерпретации.

Такие участки с высокой частотой входов и выходов следует полностью промоделировать программно и только после этого заходить в режим прямого исполнения. Поэтому в основу реализации инструмента положена следующая эвристика. Для того, чтобы симулятор находился в режиме прямого исполнения, между двумя вызывающими выход инструкциями должно быть такое расстояние, т.е. количество инструкций, что время симуляции будет больше, чем время прямого исполнения этого участка плюс накладные расходы на вход в РПИ после первой инструкции и выход из него после последней.

Другие исследователи занимались подобной проблемой. Например, в [4] описаны некоторые подходы к ее решению. В нашем случае работа механизма основана на

запоминании истории выходов и дальнейшем ее использовании при повторном исполнении участка кода.

На рис. 1 приведены результаты ускорения моделирования всех сценариев из набора регулярного тестирования на микросерверной платформе после внедрения технологии. Из графика видно, что разработанная методика значительно увеличила производительность симулятора на многих сценариях.

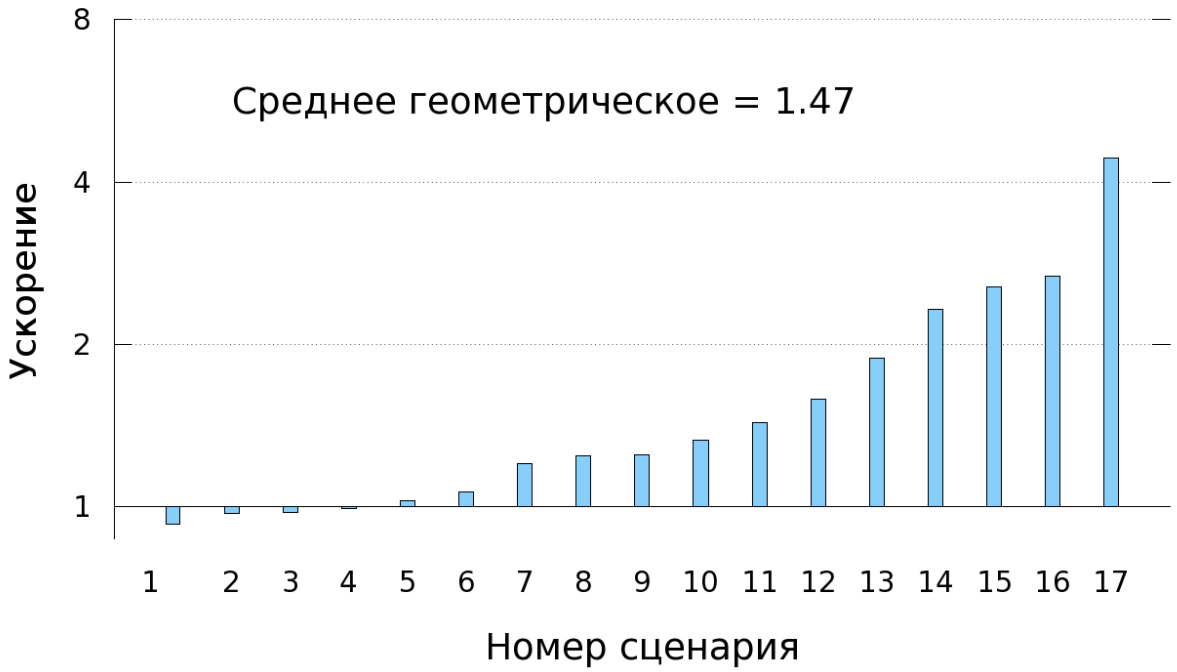

Рис. 1. Прирост скорости моделирования

Литература

1. *Magnusson P.S. [et al.]*. Simics: A Full System Simulation Platform. // Computer Vol. 35, IEEE Computer Society Press, February 2002, P. 50–58.
2. Intel Corporation. Intel® 64 and IA-32 Architectures Software Developer’s Manual. 2015.
3. *Речистов Г.С., Юлюгин Е.А., Иванов А.А. [и др.]*. Основы программного моделирования ЭВМ: Учебное пособие. 2-е изд., М.: Издательство МФТИ, 2013. – 222 с.
4. *Agesen O., Mattson J., Rugina R., Sheldon J.* Software Techniques for Avoiding Hardware Virtualization Exits. // USENIX, 2012. P. 373–385.