

Высокоэффективная масштабируемая реализация явного конечно-разностного алгоритма для решения волновых уравнений.

И.А. Валуев^{1,2}, А.В. Закиров²

¹ОИВТ РАН, ²ООО «Кинтех Лаб»

Традиционные реализации явных конечно-разностных схем для расчета распространения волн в среде (например, решения уравнений Максвелла или акустики) используют пошаговую синхронизацию всей расчетной сетки по времени. Такая реализация неэффективна с точки зрения иерархии памяти современного компьютера и приводит к тому, что скорость всего расчета лимитируется наиболее медленным из применяемых каналов загрузки данных в процессор(ы), осуществляющие операции с плавающей точкой. В зависимости от размера задачи таким лимитирующим фактором может быть скорость загрузки данных в регистры, в кэши разного уровня, в оперативную память. В результате зависимость используемой производительности вычислительной машины от размера решаемой задачи резко падает при увеличении объема обрабатываемых данных (размера сетки).

Поясним это на примере сеточного решения уравнений Максвелла с помощью традиционного послойного алгоритма Йи (применяется в большинстве коммерческих программ FDTD). На Рис.1 показана зависимость производительности послойного расчета от размера задачи при использовании разного уровня параллелизма на 4-ядерном процессоре Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz. Обновление (переход к следующему временному слою) одной ячейки сетки, хранящей 6 чисел с плавающей точкой, предполагает 24 операции multiply-add. Это дает возможность оценить пиковую производительность вычислительной машины на данном алгоритме, предполагая, что операции multiply-add производятся за 1 такт с максимально допустимым уровнем параллелизма, т.е. одновременно число ядер процессора умножить на размер векторного регистра (напр. SSE, AVX). Пиковые производительности показаны на Рис. 1 пунктирными линиями для разных уровней задействованного параллелизма. Из рисунка видно, что при малых размерах задачи производительность расчета приближается к пиковой, но на размерах 1-100 МБ происходит резкое падение скорости, которая в дальнейшем лимитируется пропускной способностью шины данных.

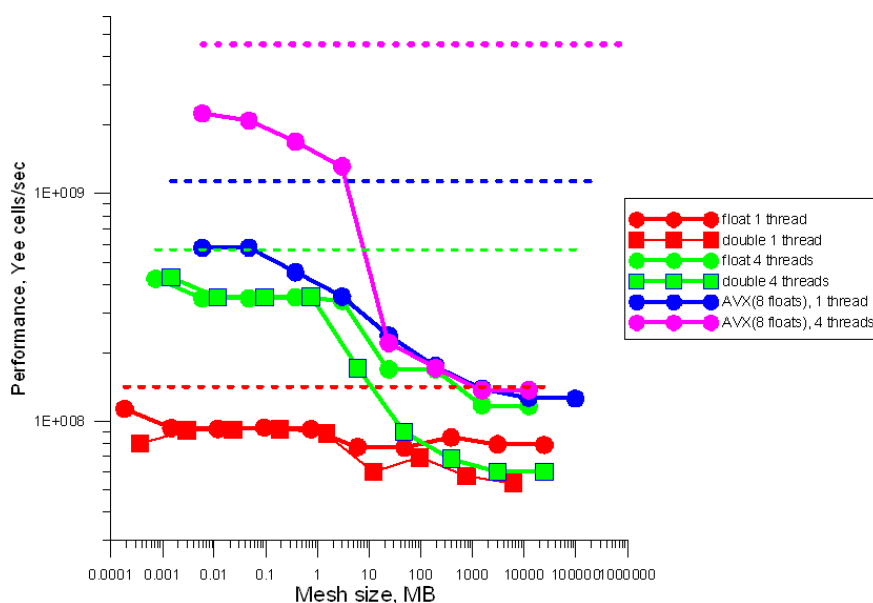


Рис. 1 Производительность расчета алгоритма Йи в зависимости от размера сетки и используемого уровня параллелизма на процессоре Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz

Несмотря на то, что ситуация показанная на Рис.1, является типичной для большинства реализаций сеточных алгоритмов, задача явного пространственно-временного обновления сетки имеет значительный резерв повторного использования данных, загружаемых в память на разных уровнях иерархии. Для использования этого резерва требуется учет зависимостей по данным между ячейками различных временных слоев, которые являются локальными (затрагивающими небольшое количество соседних ячеек) и определяются сеточным шаблоном.

В данной работе будет продемонстрирована пространственно-временная реализация явного сеточного решения уравнений Максвелла на основе локально-рекурсивного нелокально-асинхронного алгоритма, ЛРНА. Этот тип алгоритмов разработан в ИПМ им. Келдыша РАН[1] и основан на рекурсивном анализе многомерного графа зависимостей задачи по данным и его разбиении на подграфы. Учет временной составляющей конечно-разностного алгоритма и отказ от пошаговой синхронизации позволяет повторно использовать данные, загруженные в память каждого из уровней иерархии. При этом оптимальная эффективность алгоритма достигается без явного учета параметров фактической иерархии памяти, т.е.(cache oblivious) [2].

Оптимальная реализация алгоритмов ЛРНЛА является алгоритмически более сложной по сравнению с традиционными схемами. Эта сложность связана с требованием максимального учета структуры графа зависимостей задачи на этапе компиляции программного кода, поскольку перенесение анализа графа на этап исполнения ведет к большим накладным расходам по вычислениям. В данной работе обсуждается переносимая реализация ЛРНЛА на языке C++ с использованием стандартного компилятора. Приводятся результаты тестов масштабируемости для различных многоядерных архитектур, включая Intel Xeon E5-2690, Intel Xeon Phi (если успеет возможно GPU Nvidia Tesla). Обсуждается использование параллелизма сетевого уровня (стратегия реализации на MPI).

В результате применения ЛРНА скорость обновления сетки может быть доведена до 75-90% расчетной пиковой производительности процессора при сохранении полной масштабируемости по размеру данных (Рис. 2). Так, для реальных физических задач, применение ЛРНА позволяет уменьшить время расчета для полноволновой модели светодиода (объем сетки порядка 10^9 ячеек) в 10-30 раз по сравнению с традиционной реализацией. Обсуждаемая реализация алгоритма обновления сетки станет основой нового отечественного программного продукта для моделирования распространения электромагнитных волн в среде, создаваемого ООО «Кинтех Лаб» на базе библиотеки EMTL[3].

[1] А. Ю. Перепёлкина, В. Д. Левченко, И. А. Горячев. Трехмерный кинетический код CFHall для моделирования замагниченной плазмы. Математическое моделирование. 2013 г., Т. 25, 11.

[2] H. Prokop. Cache-Oblivious Algorithms. Master's thesis, MIT. 1999.

[3] <http://fdtd.kintechlab.com>