

ПРИМЕНЕНИЕ МНОГОСЕТОЧНЫХ МЕТОДОВ ДЛЯ РЕШЕНИЯ ЭЛЛИПТИЧЕСКИХ ЗАДАЧ С ПОМОЩЬЮ ПАКЕТА OPENFOAM

Д.В. Быстриков

Кемеровский госуниверситет

Представлены результаты экспериментального исследования применения многосеточных методов для решения эллиптических задач с помощью пакета OpenFOAM. В качестве задачи, при решении которой исследуется применимость многосеточных методов, выбрана задача обрушения столба жидкости в прямоугольном резервуаре без перегородки. При решении подобных задач используется метод объема жидкости, реализованный в решателе interFoam пакета OpenFOAM.

Введение

По мере того как суперкомпьютеры становятся все более популярным инструментом для решения задач в различных областях науки, у исследователей появляется возможность решать ресурсоемкие задачи с большей точностью. При решении задач гидродинамики очень важным является выбор метода решения систем линейных алгебраических уравнений (СЛАУ). Для больших задач время решения этих систем может покрывать существенную долю общего времени расчёта. Правильный выбор метода может значительно сократить время расчета на суперкомпьютерах.

Методы оценивались по двум критериям:

- времени счета;
- эффективности распараллеливания в системах с общей и распределенной памятью.

OpenFOAM (англ. Open Source Field Operation And Manipulation CFD ToolBox) – открытая интегрируемая платформа для численного моделирования задач механики сплошных сред. Большим преимуществом пакета OpenFOAM является то, что использование данного пакета на суперкомпьютерах не требует от пользователя больших усилий. Это позволяет пользователям решать все более сложные и ресурсоемкие задачи. Запуск OpenFOAM на нескольких процессорах заключается в разбиении расчетной области на части, каждая из которой решается отдельным процессором. Разбиение производится четырьмя различными способами:

- метод *simple*. Область разбивается на части в направлениях x , y и z . Количество областей в каждом направлении задается пользователем;
- метод *hierarchical*. Отличается от метода *simple* тем, что пользователь задает порядок разбиения, например сначала направление y , потом x , потом z ;
- метод *scotch*. Пользователь указывает правило, по которому разбивается сетка, и указывает вес каждого процессора. Очень полезный метод, если задача решается на процессорах с разной производительностью;
- метод *manual*. Ручное разбиение, в котором пользователь явно задает расположение каждой подобласти для определенного процессора.

Во многих задачах для решения системы уравнений Навье-Стокса используются методы, в которых для нахождения давления необходимо решать уравнение Пуассона. Как правило, соответствующая матрица линейных алгебраических уравнений является симметричной и положительно определенной. Весьма эффективными для решения СЛАУ являются многосеточные методы.

Многосеточный метод – метод решения системы линейных алгебраических уравнений, основанный на использовании последовательности уменьшающихся сеток и операторов перехода от одной сетки к другой. Многосеточные методы эффективны для задач эллиптического и параболического типов. В работах [1,2] теоретически обосновывается применимость многосеточных методов для симметричных и положительно определенных матриц. Многосеточные методы являются оптимальными относительно числа операций, затрачиваемых на получение решения с заданной точностью. Важной особенностью является то, что, обладая высокой эффективностью, многосеточные методы допускают наиболее естественное распараллеливание, что позволяет отнести данные методы к перспективному разделу современных высокопроизводительных алгоритмов.

В качестве методов решения СЛАУ использовались следующие:

- метод сопряженных градиентов с предобуславливанием типа неполной факторизации Холецкого (DICPCG);
- геометро-алгебраический многосеточный метод (GAMG) с предобуславливанием типа неполной факторизации Холецкого. В качестве сглаживателя выбран метод Гаусса-Зейделя;
- метод сопряженных градиентов с геометро-алгебраическим многосеточным предобуславливателем (GAMGPCG).

В качестве многосеточного цикла использовался V-цикл. Идея данного подхода заключается в том, что сначала ищется решение на подробной сетке, затем проводится последовательно вычисление невязок на все более грубых сетках, затем – несколько коррекций с грубой сетки и сглаживающие итерации на все более и более подробных сетках.

Результаты

Первый этап экспериментального исследования заключался в проведении расчетов на одном узле вычислительного кластера (в системе с общей памятью). Исследование проводилось на различном числе процессоров (1, 2, 4 и 8) и сетке, состоящей из различного числа ячеек (от 10 тысяч ячеек до 100 тысяч ячеек с шагом 10 тысяч). Каждый расчет был произведен несколько раз, представленные результаты являются средним арифметическим проведенных экспериментов.

Использование различных методов разбиения области показало, что методы *simple* и *hierarchical* показывают лучшие результаты по времени, чем метод *scotch*. Это связано с тем, что, во-первых, геометрия области простая, а во-вторых, производительность каждого процессора одинаковая. На рис. 1 представлены результаты расчетов на 1 процессоре с использованием различных методов решения СЛАУ. На всех представленных рисунках по оси абсцисс – количество ячеек сетки, по оси ординат – время расчета.

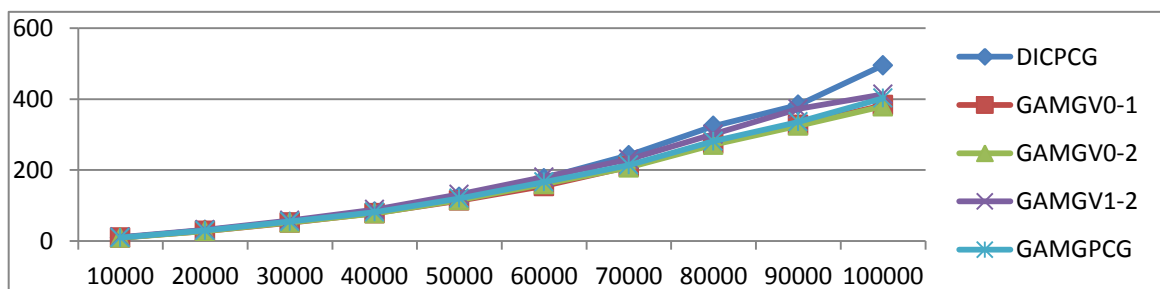


Рис. 1. Результаты расчета на 1 процессоре

Из данных, представленных на рис. 1, видно, что методы GAMG с предобуславливанием типа неполной факторизации Холецкого с V-циклом 0-1 и V-циклом 0-2 по-

казывают лучшие результаты по скорости и позволяют сократить время расчета на 20% при достаточно подробной сетке по сравнению с методом DICPCG.

На рис. 2 представлены результаты расчетов на 2 процессорах с использованием различных методов решения СЛАУ.

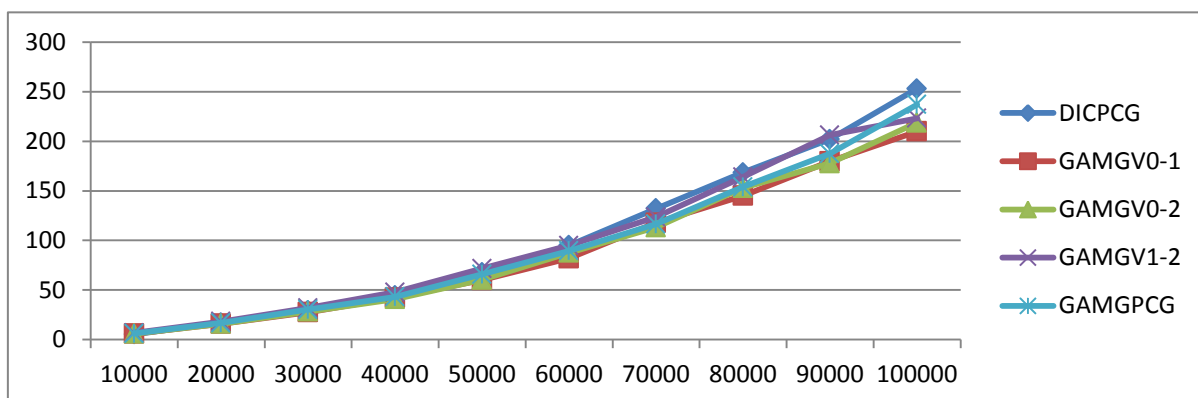


Рис. 2. Результаты расчета на 2 процессорах

Из результатов, представленных на рис. 2, видно, что те же методы показывают наилучшие результаты по времени расчета. Стоит отметить, что все алгоритмы распараллеливания, используемые в пакете OpenFOAM, показали высокую эффективность.

Результаты расчета ускорения и эффективности алгоритмов распараллеливания при расчете на 2 процессорах представлены в таблице 1. В таблицах 1-3 по оси ординат отложено количество ячеек сетки, S – ускорение, E – эффективность.

Таблица 1. Ускорение и эффективность расчетов на 2 процессорах

	DICPCG		GAMGV0-1		GAMGV0-2		GAMGV1-2		GAMGPCG	
	S	E(%)	S	E(%)	S	E(%)	S	E(%)	S	E(%)
10000	1,704	85,2	1,628	81,4	1,58	79,4	1,624	81,2	1,604	80,2
20000	1,809	90,5	1,789	89,5	1,779	88,9	1,753	87,7	1,75	87,5
30000	1,824	91,2	1,873	93,6	1,849	92,5	1,815	90,7	1,815	90,7
40000	1,85	92,5	1,853	92,6	1,903	95,2	1,861	93,1	1,887	94,3
50000	1,832	91,6	1,898	94,9	1,913	95,6	1,832	91,6	1,827	91,4
60000	1,859	92,7	1,889	94,5	1,831	91,6	1,899	95	1,858	93
70000	1,829	91,5	1,779	88,9	1,828	91,4	1,863	93,2	1,839	92
80000	1,922	96,1	1,891	94,6	1,773	88,7	1,837	92	1,828	91,4
90000	1,9	95	1,821	91	1,829	91,4	1,807	90,4	1,792	89,6
100000	1,957	97,86	1,824	91,2	1,736	86,8	1,852	92,6	1,697	84,9

Анализ табл. 1 показывает, что параллельные версии многосеточных методов дают меньшее ускорение, чем метод DICPCG, по сравнению со своими последовательными аналогами, однако время расчета по-прежнему на 20% меньше, чем при использовании метода DICPCG.

На рис. 3 представлены результаты расчета на 4 процессорах с использованием различных методов решения СЛАУ.

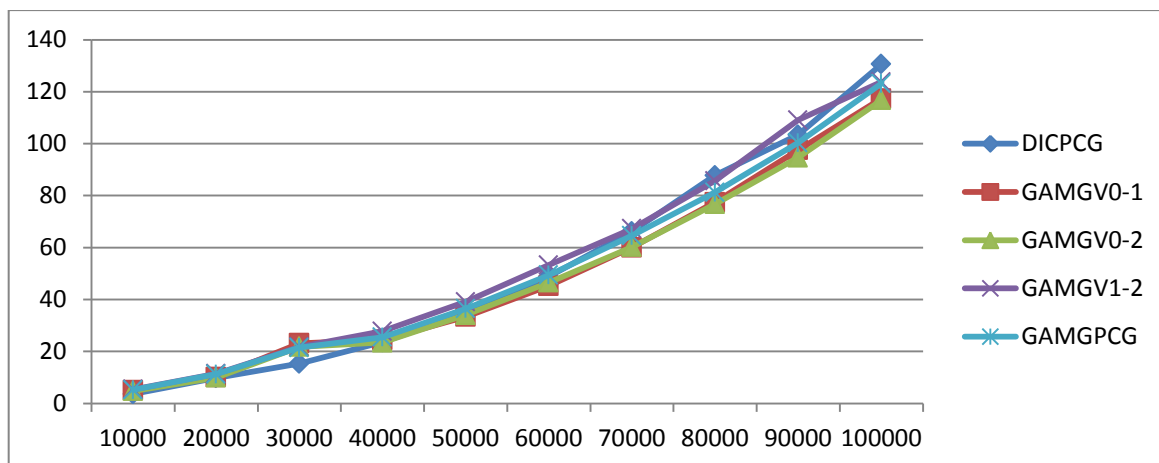


Рис. 3. Результаты расчета на 4 процессорах

Тенденция временных затрат используемых методов аналогична случаю использования 1, 2 процессоров. Кроме того, можно заметить, что при размерности сетки 30000 ячеек наблюдается скачок у всех многосеточных методов. Данное явление объяснить затруднительно. Стоит отметить, что все алгоритмы распараллеливания, используемые в пакете OpenFOAM, показали эффективность ниже, чем при расчете на 2 процессорах. Это объясняется уменьшением счета на каждом процессоре и увеличением коммуникаций в параллельной программе. Результаты расчета ускорения и эффективности алгоритмов распараллеливания при расчете на 4 процессорах представлены в табл. 2.

Таблица 2. Ускорение и эффективность расчетов на 4 процессорах

	DICPCG		GAMGV0-1		GAMGV0-2		GAMGV1-2		GAMGPCG	
	S	E(%)	S	E(%)	S	E(%)	S	E(%)	S	E(%)
10000	2,581	64,5	1,934	48,4	2	50	2,051	51,3	1,883	47,1
20000	3,163	79,1	2,805	70,1	2,871	71,8	2,798	69,9	2,626	65,7
30000	3,507	87,7	2,232	55,8	2,405	60,1	2,672	66,8	2,566	64,2
40000	3,491	87,3	3,207	80,2	3,317	82,9	3,204	80,1	3,189	79,7
50000	3,573	89,3	3,415	85,4	3,389	84,7	3,360	84	3,302	82,5
60000	3,62	90,5	3,423	85,6	3,443	86,1	3,382	84,5	3,368	84,2
70000	3,649	91,2	3,488	87,2	3,44	86	3,424	85,6	3,315	82,9
80000	3,69	92,2	3,544	88,6	3,535	88,4	3,515	87,9	3,463	86,6
90000	3,71	92,7	3,353	83,8	3,433	85,8	3,414	85,4	3,347	83,7
100000	3,79	94,8	3,262	81,5	3,251	81,3	3,339	83,5	3,268	81,7

На рис. 4 представлены результаты расчета на 8 процессорах с использованием различных методов решения СЛАУ.

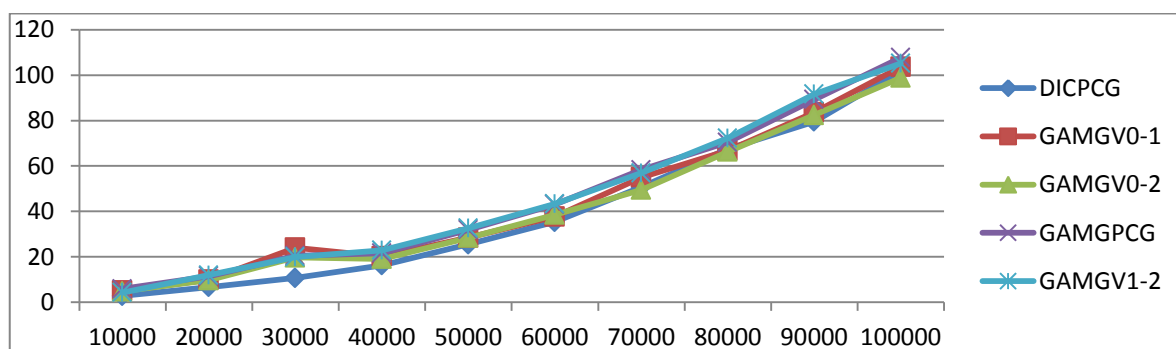


Рис. 4. Результаты расчета на 8 процессорах

Результаты, представленные на рис. 4, показывают, что время расчета на 8 процессорах примерно равно времени расчета на 4 процессорах, что говорит о плохой масштабируемости параллельных алгоритмов. Следовательно, увеличивать число процессоров не имеет смысла, так как это может привести даже к увеличению времени расчета. Кроме того, у всех многосеточных методов снова наблюдается скачок времени расчета при размерности 30000 ячеек.

Результаты расчета ускорения и эффективности алгоритмов распараллеливания при расчете на 8 процессорах представлены в таблице 3.

Таблица 3. Ускорение и эффективность расчетов на 8 процессорах

	DICPCG		GAMGV0-1		GAMGV0-2		GAMGV1-2		GAMGPCG	
	S	E(%)	S	E(%)	S	E(%)	S	E(%)	S	E(%)
10000	3,34	41,7	1,915	23,9	2	25	2,583	32,3	1,663	20,8
20000	4,632	57,9	2,89	36,1	2,939	36,7	2,674	33,4	2,578	32,2
30000	5,048	63,1	2,149	26,9	2,658	33,2	2,922	36,5	2,695	33,7
40000	5,062	63,3	3,877	48,5	4,077	50,9	3,891	48,6	3,766	47,1
50000	4,875	60,9	4,008	50,1	4,065	50,8	4,028	50,3	3,771	47,1
60000	4,997	62,5	4,111	51,4	4,163	52,1	4,163	52,1	3,87	48,4
70000	4,764	59,5	3,809	47,6	4,187	52,3	4,047	50,1	3,672	45,9
80000	4,796	59,9	4,103	51,3	4,093	51,1	4,167	52,1	3,993	49,9
90000	4,825	60,3	3,912	48,9	3,944	49,3	4,064	50,8	3,763	47
100000	4,858	60,7	3,693	46,2	3,84	48	3,929	49,1	3,735	46,7

Расчеты проводились на кластере МСЦ РАН.

Результаты показали, что использование многосеточных методов может существенно сократить время расчета в системах с общей памятью.

В настоящее время проводятся эксперименты для вычисления эффективности распараллеливания методов решения СЛАУ при более подробном разбиении расчетной области (и соответственно большем числе ячеек) в системах с общей и распределенной памятью.

Литература

1. Федоренко Р.П. Введение в вычислительную физику: учеб. пособие. М.: Издательство Московского физико-технического института, 1984. – 526 с.
2. Федоренко Р.П. Релаксационный метод решения разностных эллиптических уравнений // Журнал вычислительной математики и математической физики. – 1961. – 1(5). С. 922-927.