ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ВЫЧИСЛИТЕЛЬНОЙ ГИДРОДИНАМИКИ SIMPLE

Н.Н.Богословский, А.О.Есаулов, А.В.Старченко

Томский государственный университет

Решение многих задач, в основу которых положена система уравнений Навье - Стокса, всегда являлось непростой вычислительной проблемой. Решение этой системы без существенных упрощений возможно при условии взаимодополнения эффективных численных методов применением высокопроизводительных компьютеров. Суперкомпьютерная техника с самого своего появления находила широкое применение в этой области математического моделирования. В последние годы в нашей стране резко повысилось число относительно недорогих многопроцессорных вычислительных систем, что, разумеется, сказалось и на общем уровне их доступности для проведения трудоемких расчетов.

Использование разработанных ранее эффективных последовательных методов существенно ограничивается тем, что параллельные реализации алгоритмов, как правило, существенно отличаются от аналогичных последовательных. По сути дела, зачастую проще разработать новую параллельную программу, чем распараллелить старую последовательную.

Самой распространенной задачей, на которой принято тестировать численные схемы решения системы уравнений Навье - Стокса, является задача о вихревом движении в прямоугольной полости (каверне), целиком заполненной вязкой несжимаемой жидкостью [1, 2].

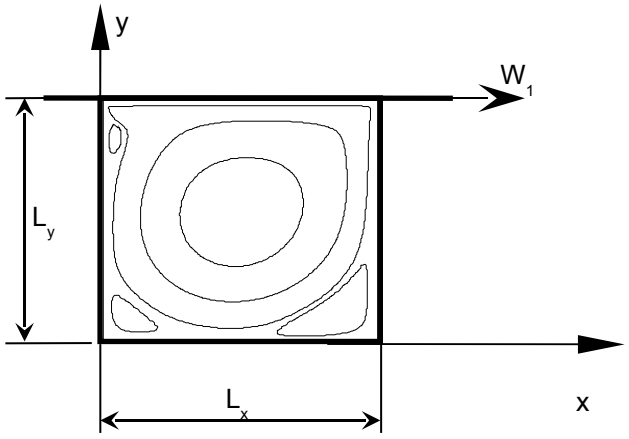

Рис. 1. Каверна

Авторами рассматривалась двумерная задача о стационарном течении несжимаемой вязкой жидкости с постоянными свойствами в прямоугольной полости (ка-

верне) с верхней стенкой, движущейся с постоянной скоростью (рис. 1). Жидкость, целиком заполняющая каверну, вовлекается в движение силами вязкости (действие массовых сил незначительное). Кроме того, требуется выполнение условия прилипания частиц жидкости к твердой стенке.

Рассматриваемое течение (в так называемых примитивных переменных) определяется уравнениями Навье - Стокса:

уравнение неразрывности

$$\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}=0;$$

уравнения движения

$$\frac{\partial(\rho u^2)}{\partial x}+\frac{\partial(\rho uv)}{\partial y}=\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right)-\frac{\partial p}{\partial x};$$

$$\frac{\partial(\rho uv)}{\partial x}+\frac{\partial(\rho v^2)}{\partial y}=\frac{\partial}{\partial x}\left(\mu\frac{\partial v}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\frac{\partial v}{\partial y}\right)-\frac{\partial p}{\partial y};$$

а также *уравнениями состояния*:

$$\rho = \text{const}; \quad \mu = \text{const}$$

и граничными условиями (условиями прилипания):

$$x=0:\begin{cases}u=0,\\ v=0,\end{cases} \quad x=L_x:\begin{cases}u=0,\\ v=0,\end{cases}$$

$$y=0:\begin{cases}u=0,\\ v=0,\end{cases} \quad y=L_y:\begin{cases}u=W_1,\\ v=0,\end{cases}$$

где u,v – горизонтальная и вертикальная компоненты скорости; μ – коэффициент динамической вязкости; ρ – плотность; p – давление.

Для дискретизации исходной системы дифференциальных уравнений применялся метод контрольного объема. Конечно-разностный аналог получался на разнесенных сетках (в центре контрольного объема вычисляется давление, а на гранях – компоненты скорости). Сетка строилась с узлами, сгущающимися к границам расчетной области. Для аппроксимации

конвективных потоков применялись обычные противопотоковые разности (выбор схемы первого порядка обусловлен тем, что авторы в первую очередь стремились получить простую параллельную реализацию классического алгоритма, тогда как в будущем предполагается использование схем более высокого порядка [2]).

Решение систем алгебраических уравнений, получающихся в результате дискретизации, производилось методом Гаусса - Зейделя с нижней релаксацией и явным методом Булеева.

Для получения поля давления, согласованного с векторным полем скорости, использовался алгоритм SIMPLE Патанкара [3], основные этапы которого можно представить следующей последовательностью шагов:

1) задаются начальные значения полей скорости и давления;
2) решаются уравнения движения и рассчитываются компоненты вектора скорости;
3) решается уравнение для поправки давления;
4) корректируются компоненты вектора скорости так, чтобы они удовлетворяли уравнению неразрывности;
5) в случае, если максимальное абсолютное значение поправки давления велико, производится возврат ко второму шагу.

Распараллеливание производилось по данным, то есть расчетная область разделялась на подобласти и в каждой подобласти вычисления производились одновременно. Из практики параллельных вычислений хорошо известно, что в целом более эффективная двумерная декомпозиция расчетной области теряет свои преимущества перед одномерной в том случае, если число процессоров параллельного компьютера невелико [4]. А так как изначально ставилась задача проведения расчетов на имеющихся в Томске кластерных системах, то использовалось одномерное разрезание вдоль горизонтальной координаты (рис. 2), так как такому разрезанию свойственна простота программной реализации.

На рис. 2 в правой части схематично изображено распределение данных между процессами. Левая половинка выделенного круга представляет область, рассчитываемую 0-процессом, а правая половинка – рассчитываемую 1- процессом. Кружками обозначены точки, в которых

вычисляется давление, а горизонтальными и вертикальными штрихами – точки, соответствующие горизонтальной и вертикальной компонентам скорости.

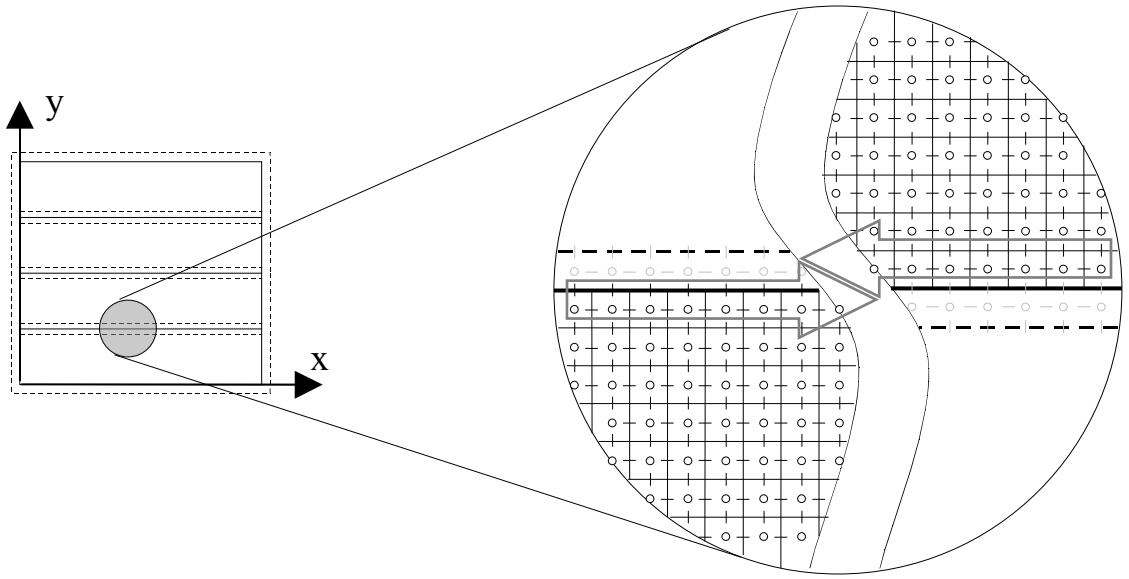

Рис. 2. Одномерная декомпозиция расчетной области

На разных этапах расчетов требовалась синхронизация, то есть обмен данными, расположенными в прилежащих к границам разрезов областях, между соседними процессами. Например, для расчета давления в граничном сеточном слое требуются четыре смежных значения (пятиточечный шаблон), три из которых рассчитываются и расположены в данном процессе, а четвертое надлежит получить от соседнего процесса. По этой причине необходима пересылка последней строки 0-процесса 1-процессу, равно как и получение от него его первой строки. Для этой цели использовались операции блокирующей парной пересылки данных MPI (функция *SendRecv*). Аналогичным образом выглядит схема расчета и обмена значениями для горизонтальной компоненты скорости. Для вертикальной компоненты скорости в принципе схема остается той же, с той лишь разницей, что вычисление скорости на границе возлагается на предыдущий процесс (в данном случае на 0-процесс), тогда как последующий ведет расчеты, начиная с первой сеточной линии, отличной от «линии разреза».

Обобщая, можно сказать, что для каждой характеристики (компонента скорости, давление) каждый процесс должен осуществлять операции посылки

первой и последней строк массива данных, так же как и принимать граничные условия, представляющие собой данные, присланные соседними процессами.

Последовательность шагов алгоритма SIMPLE при его параллельной реализации претерпит определенные изменения.

Во-первых, расчеты компонент скорости и поправки давления производятся каждым процессом в своей подобласти.

Во-вторых, после расчета и коррекции скорости требуется обмен приграничными значениями (после пп. 2 и 4).

В-третьих, после расчета поля давления (п. 3) необходима пересылка граничных значений давления; кроме того, в ходе решения уравнения для поправки давления нужно производить синхронизацию операции поправки давления.

Так как компоненты скорости не обязательно вычислять точно на п.2, было решено применить простой метод Гаусса - Зейделя для вычисления компонент скорости. Точность численного решения уравнения Пуассона для поправки давления определяет степень удовлетворения уравнения неразрывности, поэтому его итерационное решение нужно проводить достаточно точно. По этой причине использовался явный метод Н.И.Булеева. На «разрезах» ставились условия первого рода для поправки давления, то есть в качестве граничных условий использовались результаты расчетов соседних процессов. Так как в ходе решения уравнения Пуассона эти «граничные» условия менялись, потребовалась синхронизация поправки давления на каждой итерации метода Булеева. Как было практически установлено для вычислительных систем, на которых производились расчеты, время выполнения парной операции пересылки одного элемента массива примерно в 10-15 раз больше времени выполнения одной арифметической операции над этим элементом. Была произведена попытка уменьшить число пересылок путем проведения синхронизации не на каждой итерации, а скажем, через 5. Но, к сожалению, она не привела к уменьшению времени расчетов, так как ухудшились свойства сходимости алгоритма и в целом выигрыш оказался сомнительным.

Предложенная методика, без сомнения, может быть применена не только к изучению течения жидкости в каверне, но и к более сложным задачам.

На исходной задаче (течение вязкой жидкости в каверне) исследовалось ускорение параллельного алгоритма по сравнению с последовательным алгоритмом.

Расчеты велись на нерегулярных сетках (от 40x40 до 200x200 узлов) при Re=1000 с использованием удвоенной точности. Итерационный процесс продолжался до тех пор, пока максимальное значение поправки давления не становилось меньше 10^{-4}. Релаксационные коэффициенты в методах Гаусса - Зейделя и Булеева составляли соответственно 0,5 и 0,8.

Как известно, анализ параллельного алгоритма позволяет получить аналитическую формулу ускорения в виде

$$S_p = \frac{p}{1 + \gamma \frac{p(p-1)}{N_x} \cdot \frac{t_{\text{п}}}{t_{\text{а}}}},$$

где N_x – размерность расчетной сетки; p – число процессоров; $t_{\text{п}}$ – время выполнения операции пересылки; $t_{\text{а}}$ – время выполнения арифметической операции, а полуэмпирический параметр γ характеризует соотношение объема пересылаемых к общему объему данных, то есть свойства самого алгоритма. Данному алгоритму соответствует $\gamma \approx 0{,}21$.

Увеличение числа процессоров уменьшает отношение количества внутренних узлов к количеству граничных, откуда нетрудно сделать вывод, что эффективность параллельной реализации данного алгоритма будет снижаться с увеличением числа процессоров при фиксированном N_x. Наоборот, увеличение размерности расчетной сетки при фиксированном p влечет повышение эффективности.

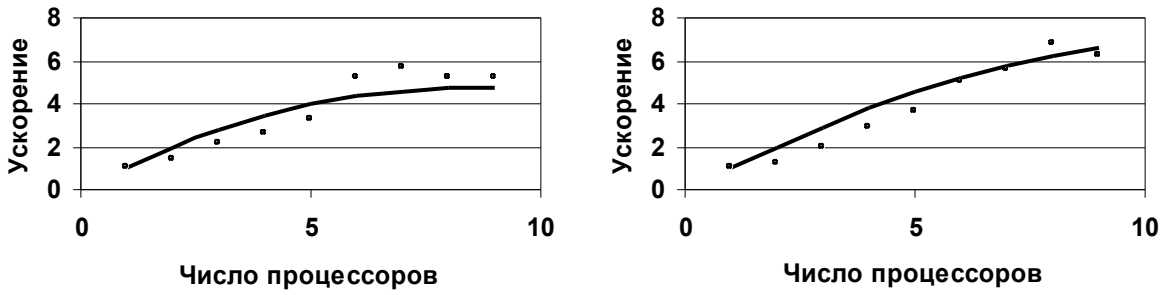

Рис. 3. Ускорение параллельного алгоритма, слева - N_x=40, справа - N_x=100

На рис. 3 изображены зависимости ускорения, показываемого параллельным алгоритмом для расчетных сеток размерностей 40x40 и 100x100 узлов. Сплошная линия соответствует аналитической зависимости, а кружки – экспериментальным данным. Рис. 4 представляет зависимость времени расчетов от размерности сетки.

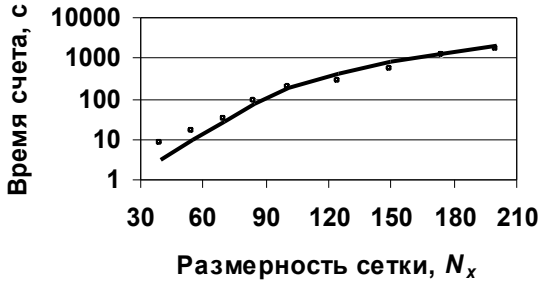

Рис. 4. Ускорение параллельного алгоритма на 9 процессорах при различных размерностях сетки

В итоге можно сделать вывод, что, несмотря на сложность рассматриваемого алгоритма, его параллельная реализация достаточно эффективна: на 9 процессорах ускорение может происходить в 6-7 раз по сравнению с последовательным алгоритмом.

Расчеты производились на кластере Томского государственного университета (9 процессорных пар Pentium-III-650).

Список литературы

1. *Лойцянский Л.Г.* Механика жидкости и газа. – М.: Наука, 1987. – 840 с.

2. *Есаулов А.О., Старченко А.В.* К выбору схемы для численного решения уравнений переноса //Вычислительная гидродинамика. – Томск: Изд-во Том. ун-та, 1999. – С.27-32.
3. *Патанкар С.* Численные методы решения задач теплообмена и динамики жидкости /Пер. с англ. – М.: Энергоатомиздат, 1984. – 152 с.
4. *Корнеев В.Д.* Параллельное программирование в MPI. – Новосибирск: Изд-во СО РАН, 2000. – 213 с.