Экспериментальное исследование масштабируемости программного комплекса FlowVision HPC 3.07 на кластере СКИФ ВлГУ «Мономах»

М.С. Драгомиров[1)], М.К. Михайлова[2)], А.А. Павлов[3)], М.А. Цыганов[1)], С.А. Харченко[2)], А.Е. Щеляев[2)]

[1)] ОАО «НИПТИЭМ», г Владимир
[2)] ООО «ТЕСИС», г. Москва
[3)] Владимирский государственный университет

Введение

Моделирование задач вычислительной аэро- и гидродинамики является актуальной проблемой многих отраслей промышленности. Существует несколько программных продуктов, позволяющих решать подобные задачи, среди которых можно выделить российскую разработку FlowVision HPC компании «ТЕСИС». Новая версия указанной программы позволяет проводить параллельные вычисления при разных вариантах построения вычислительных систем – многопроцессорный персональный компьютер, сеть персональных компьютеров, вычислительный кластер.

Одной из основных проблем при практическом применении численного моделирования движения жидкостей и газов является невысокая скорость проведения расчетов, которая обусловлена двумя обстоятельствами. Во-первых, для адекватного воспроизведения тонких физических эффектов в геометрически сложных трехмерных областях требуются подробные расчетные сетки, содержащие от сотен тысяч до десятков миллионов расчетных ячеек. Во-вторых, моделирование подобных задач требует огромных вычислительных ресурсов и применение персональных компьютеров либо крайне неэффективно, либо вообще невозможно. Решение практических задач с разумной производительностью проведения расчетов может быть выполнено только на самой современной высокопроизводительной параллельной вычислительной технике.

Современные суперкомпьютерные вычислительные системы, как правило, имеют неоднородную архитектуру. С одной стороны, имеется набор вычислительных узлов с распределенной памятью, обмен данными между которыми может быть осуществлен по быстрой обменной сети. С другой стороны, каждый узел представляет собой многопроцессорный/многоядерный компьютер с общим доступом к оперативной памяти. Известно, что эффективная хорошо масштабируемая организация параллельных вычислений при большом общем числе доступных процессоров/ядер предполагает хорошую балансировку вычислений на различных процессорах между собой. Чем больше число задействованных процессоров/ядер, тем более точная балансировка вычислений требуется для поддержания заданного уровня масштабируемости.

В новой версии программы FlowVision HPC применен оригинальный подход - алгоритм решения распараллеливается по распределенной памяти (по узлам вычислений) с использованием стандарта Message Passing Interface (MPI), а по общей памяти каждого узла распараллеливание по процессорам/ядрам осуществляется с динамическим распределением нагрузки на основе технологии Intel® Threading Building Blocks (TBB) [1].

В работе приведены результаты экспериментального исследования масштабируемости вычислений программы FlowVision для двух задач – внешнее обтекание тел сложной формы и задача с подвижным телом (вентилятор).

Основные особенности реализации параллельных вычислений во FlowVision HPC

Ключевым моментом в реализации параллельных вычислений во FlowVision HPC является использование комбинации двух технологий – стандарта MPI для распараллеливания по узлам кластера и библиотеки TBB для создания потоков вычислений на

всех ядрах узла. При этом предполагается, что на каждом узле имеется только один MPI процесс, который затем порождает на этом узле нужное количество одновременно работающих потоков вычислений. Подробно данный подход изложен в работе [2].

Основная идея внутриузловой динамической балансировки состоит в том, чтобы на каждом этапе вычислений по мере выполнения предыдущих вычислений каждое ядро внутри текущего узла динамически осуществляло выбор следующего возможного вычисления и производило его. Однако для достижения хорошей масштабируемости параллельных вычислений на кластерах с большим числом вычислительных узлов использование только внутриузловой динамической балансировки вычислений недостаточно. Необходимо каким-то образом балансировать между собой также и вычисления на большом числе узлов. Эффективная балансировка параллельных вычислений предполагает перенос хотя бы части вычислений с одних вычислительных устройств на другие. При внутриузловой динамической балансировке это перенос вычислений с одних ядер на другие, при этом за счет использования общей памяти в узле собственно явное перебрасывание данных не требуется, необходимо лишь правильно синхронизовать данные в общей памяти узла. При межузловой балансировке необходимо перебросить также и данные с одного вычислительного узла на другой, поскольку оперативная память в этом случае – распределенная. С этой точки зрения естественным образом возникает вопрос о минимизации объемов перебрасываемых данных при осуществлении балансировки. Также необходимо найти некоторую базовую информацию, на основе которой будет осуществляться балансировка вычислений. В случае общей памяти все просто – как только ядро освободилось, его можно использовать для следующего вычисления по принципу клиент-сервер, при этом клиентами выступают все ядра узла, а сервером – некоторый объект в общей памяти, изменяемый всеми ядрами через семафоры доступа. В случае распределенных вычислений подобная организация вычислений также возможна, но она будет иметь недопустимые накладные расходы.

В случае межузловой балансировки по распределенной памяти используется предыстория параллельных вычислений для оптимизации последующих. На основе информации о геометрических связях доменов между собой составляется матрица, описывающая для каждого домена все его соседние домены. На основе анализа CPU времен каждого домена и CPU времен для связей доменов на текущем шаге по физическому времени домены адаптивно перераспределяются по узлам суперкомпьютера, начиная с некоторого исходного распределения. Начальная декомпозиция расчетной сетки производится из условия близости числа расчетных ячеек во всех доменах при минимальной междоменной границе.

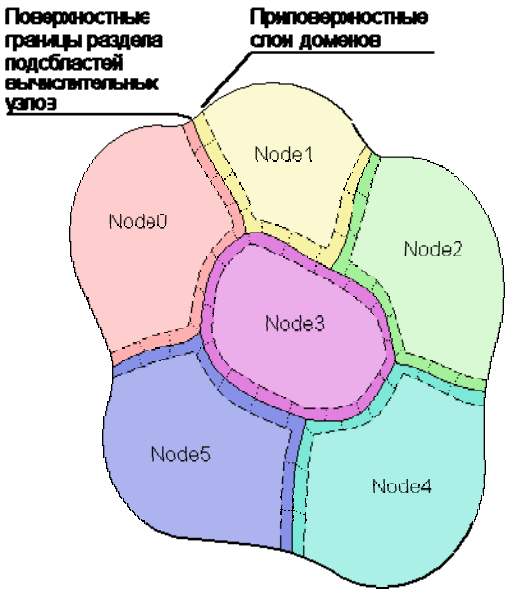


Рисунок 1. Распределение доменов по узлам кластера.

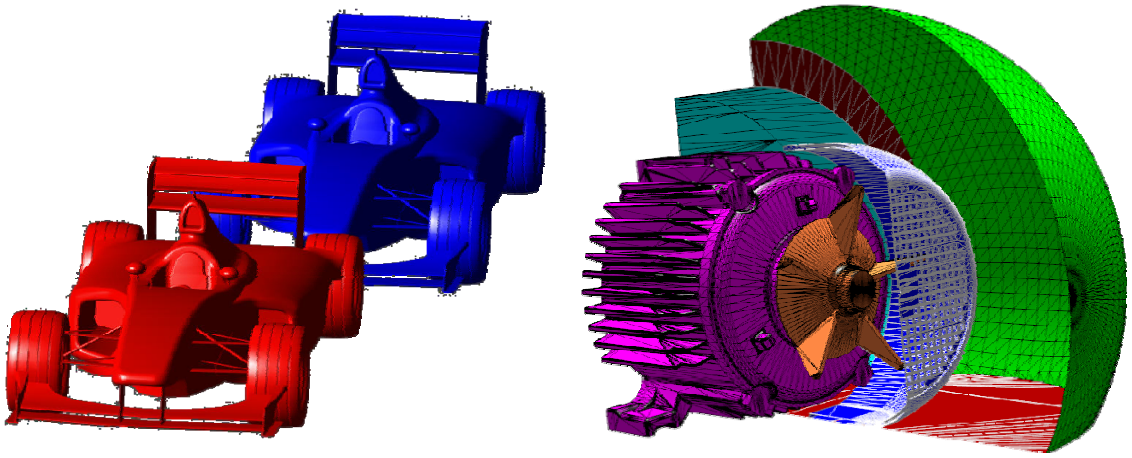
Для минимизации затрат на перенос данных для большинства доменов фиксируется предыдущее распределение доменов по узлам суперкомпьютера, а перенести расчеты некоторых доменов на другие узлы разрешается только для доменов из областей, близких к доменным поверхностным границам узлов (рис. 1) [3]. Таким образом, существенно снижаются объемы данных, которые возможно придется перебрасывать с одних узлов на другие по межузловой обменной сетке. В результате такого подхода адап-

тивно улучшается межузловая балансировка параллельных вычислений на текущем шаге по физическому времени.

Результаты экспериментальных исследований производительности параллельных вычислений

Численные эксперименты проводились на кластере СКИФ ВлГУ «Мономах» на 4 узлах под управлением Windows HPC 2008 Server. Каждый узел содержал по 2 четырехядерных процессора Intel(R) Xeon(R) CPU E5345, 2.33 GHz, каждый узел содержал 8 Gb DDR2 памяти. Узлы связаны между собой обменной сетью Mellanox Technologies InfiniBand.

Рассматривались две тестовые задачи – обтекание стационарным потоком тел сложной формы (рис. 2а) и вынужденное движение воздушного потока при работе вентилятора (рис. 2б). Во втором случае модель вентилятора являлась подвижным телом с заданной угловой скоростью.



а) *б)*

Рисунок 2. Геометрия тестовых задач для исследования масштабируемости программы FlowVision HPC.

Расчетная сетка в обеих задачах была локально адаптирована вблизи тел сложной формы и в целом содержала 1,4 млн. ячеек.

Результаты исследования масштабируемости приведены на рис. 3 (абсолютное время выполнения одной итерации) и на рис. 4 (ускорения расчета относительно использования одного ядра).

На рис. 5 и 6 приведены результаты замеров текущего времени итерации при различных вариантах запуска (первая цифра обозначает количество MPI процессов, вторая – количество нитей расчета по технологии TBB). Видно, что внутри одного узла (т.е. с использованием только технологии TBB) задача с вентилятором ведет себя нестабильно. Это вызвано тем, что в текущей версии FlowVision HPC не реализована полная поддержка подвижных тел с внутриузловой балансировкой, и реализация этого разработчиками планируется в ближайшее время.

В целом, полученные результаты позволяют говорить о том, что использование нового подхода в реализации параллельных вычислений в программе FlowVision на вычислительных кластерах может обеспечить проведение серии высокопроизводительных поисковых или оптимизационных расчетов.

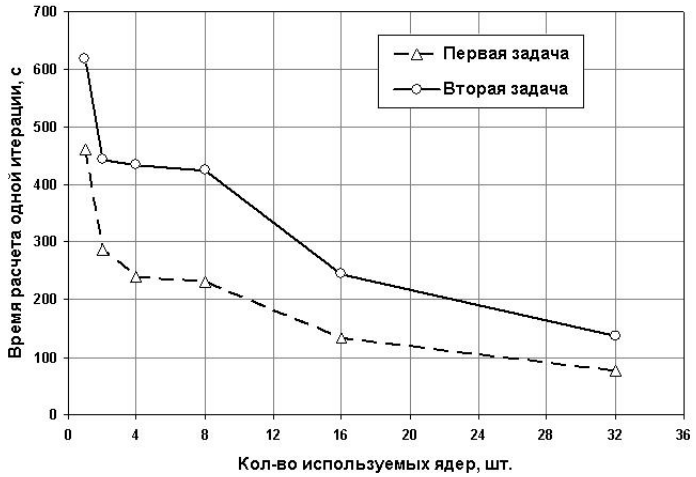


Рисунок 3. Абсолютное время выполнения одной итерации для двух тестовых задач при использовании разного числа вычислительных ядер кластера «СКИФ-Мономах».

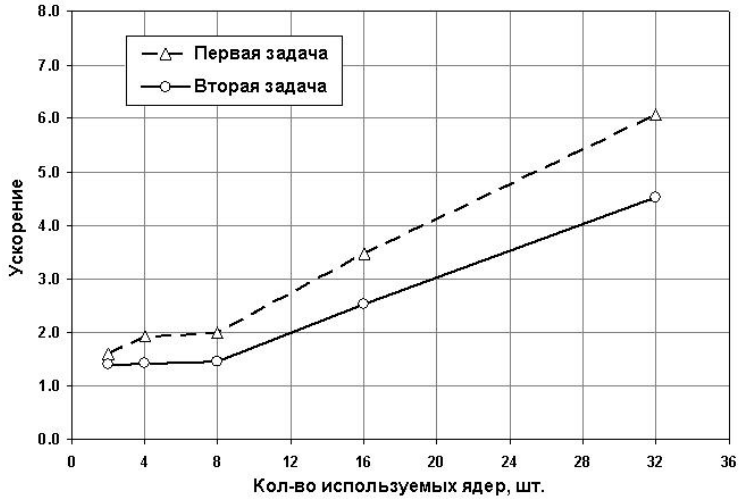


Рисунок 4. Получаемые ускорения выполнения итераций для двух тестовых задач при использовании разного числа вычислительных ядер кластера «СКИФ-Мономах».

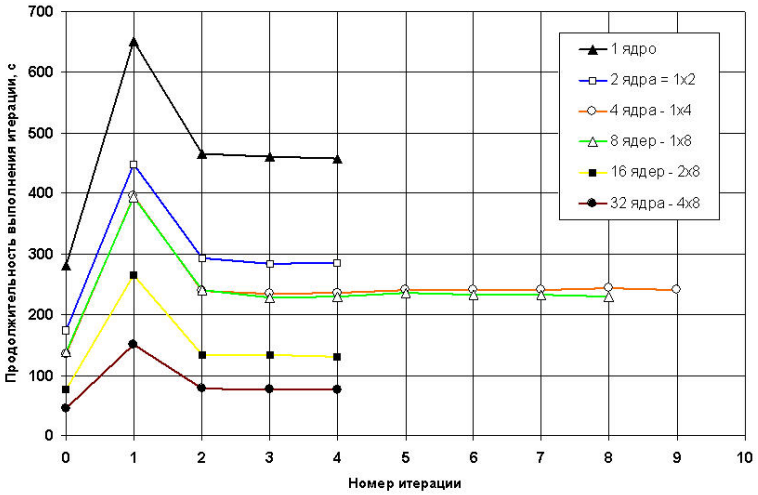


Рисунок 5. Текущее время выполнения итераций для первой тестовой задачи при использовании разного числа вычислительных ядер кластера «СКИФ-Мономах».

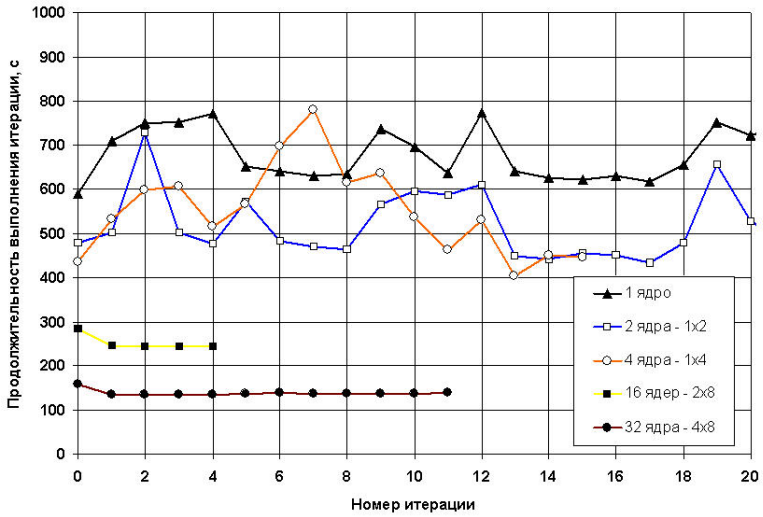


Рисунок 6. Текущее время выполнения итераций для второй тестовой задачи при использовании разного числа вычислительных ядер кластера «СКИФ-Мономах».

Литература

1. Intel Threading Building Blocks Tutorial – 1.6 / Intel Corp. 2007
2. Сушко Г.Б., Харченко С.А. “ Экспериментальное исследование на СКИФ МГУ "Чебышев" комбинированной MPI+threads реализации алгоритма решения систем линейных уравнений, возникающих во FlowVision при моделировании задач вычислительной гидродинамики” // Труды международной научной конференции Параллельные вычислительные технологии (ПаВТ'2009), Нижний Новгород, 30 марта – 3 апреля 2009 г. Челябинск, Изд. ЮУрГУ, 2009, с.316-324.
3. Аксенов А.А., Дядькин А.А., Харченко С.А. **“**Исследование эффективности распараллеливания расчета движения подвижных тел и свободных поверхностей во FlowVision на компьютерах с распределенной памятью”. Вычислительные методы и программирование, т. 10, 2009 г., с. 132-140.