ДВУХУРОВНЕВОЕ MPI+OPENMP РАСПАРАЛЛЕЛИВАНИЕ АЛГОРИТМА ДЛЯ РАСЧЕТОВ ЗАДАЧ ГАЗОВОЙ ДИНАМИКИ И АЭРОАКУСТИКИ НА ТЫСЯЧАХ ПРОЦЕССОРОВ СУПЕРКОМПЬЮТЕРА

А.В. Горобец, С.А. Суков

1. Введение

В результате активного развития вычислительной техники, производительность существующих на данный момент многопроцессорных систем измеряется уже сотнями TFLOPS. Число процессорных ядер суперкомпьютеров исчисляется десятками тысяч. Однако новая архитектура предъявляет особенные требования к алгоритмам и их реализациям. Большое количество процессоров требует высокой степени масштабируемости алгоритма, многоядерные вычислительные узлы требуют особенного внимания к оптимизации доступа к памяти, возрастающий объем расчетных данных требует новых средств параллельной обработки.

Для расчетов становятся доступны всё большие вычислительные мощности. Например, суперкомпьютеры Ломоносов, МВС100К и СКИФ МГУ, располагают существенными вычислительными ресурсами. Для эффективных расчетов на таких системах требуется расширенная параллельная реализация, соответствующая кластерной структуре вычислительной системы с многопроцессорными узлами. Также требуются средства обработки столь больших объемов расчетных данных, которые возникают при расчетах на подробных пространственных сетках с сотнями миллионов и миллиардами узлов. Необходимы новые алгоритмы и технологии выполнения расчетов, ведь речь идет не о десятках или сотнях, а о десятках тысяч процессоров, использующихся для выполнения одного расчета.

Данная работа посвящена адаптации параллельного алгоритма комплекса программ NOISETTE к современной архитектуре суперкомпьютеров для того, чтобы иметь возможность использовать до нескольких десятков тысяч процессорных ядер и подробные сетки размером до нескольких сотен миллионов узлов. Для достижения этой цели был решен целый ряд задач. Это в частности разработка и реализация двухуровневого MPI+OpenMP распараллеливания вычислительного ядра, а также расширение функциональности средств параллельной обработки сверх больших объемов расчетных и сеточных данных для неструктурированных сеток.

2. Базовые численные алгоритмы исследовательского комплекса программ NOISETTE

Комплекс программ NOISETTE предназначен для численного моделирования задач газовой динамики и аэроакустики. В нем реализован класс моделей аэроакустики [1], построенных на основе уравнений Эйлера, в который входят три основные модели: линейная (линеаризованные уравнения Эйлера), нелинейная для пульсационных компонент течения, описываемая различными формами NLDE (NonLinear Disturbance Equations) уравнений, нелинейная для всего течения, описываемая полными уравнениями Эйлера. Также реализовано действие молекулярной вязкости и теплопроводности, в рамках моделей на основе уравнений Навье–Стокса и их линейных аналогов.

Для пространственной дискретизации используются неструктурированные тетраэдральные сетки. Аппроксимация потоков через грань контрольного объема имеет повышенный порядок точности (до шестого включительно) и реализуется на расширенном шаблоне, включающем два противопотоковых тетраэдра и все соседние узлы их вершин. Более подробно со схемой высокого порядка точности, реализованной в комплексе NOISETTE, можно ознакомиться в [2-4]. Для интегрирования по времени используются явные схемы Рунге-Кутта до 4-го порядка и неявные схемы до 2-го порядка на основе линеаризации по Ньютону. Реализована инфраструктура для крупномасштабных расчетов, включающая параллельные средства для декомпозиции и измельчения сеток.

3. Двухуровневая модель распараллеливания

Современная кластерная архитектура вычислительных систем представляет собой набор вычислительных узлов, соединенных коммуникационной средой. Каждый узел имеет свое адресное пространство оперативной памяти, поэтому набор узлов представляет собой систему с распределенной памятью. При этом каждый узел имеет несколько процессорных ядер и таким образом представляет собой параллельную систему с общей памятью. Так называемое гибридное распараллеливание представляет собой двухуровневый подход, при котором на первом уровне находится описанное выше MPI распараллеливание для модели с распределенной памятью. На втором уровне дополнительно применяется OpenMP для распараллеливания внутри многопроцессорного узла.

Эффективное использование многопроцессорных узлов суперкомпьютера является достаточно сложной задачей из-за ряда проблем. Это в частности проблема эффективного использования разделяемых параллельными процессами ресурсов узла – оперативной памяти и коммуникационной инфраструктуры. В случае NUMA архитектуры добавляется также проблема неоднородного доступа к памяти и выбора привязки процессов и нитей к процессорным ядрам. При этом тенденции таковы, что рост производительности

обеспечивается в основном за счет увеличения числа ядер, но объем оперативной памяти и ее пропускная способность растут заметно медленнее.

Использование двухуровневого распараллеливания позволяет повысить эффективность параллельных вычислений на суперкомпьютерах с многоядерными узлами, поскольку в T раз сокращается число MPI процессов, где T – число OpenMP нитей. Применение OpenMP дает следующие преимущества:

1. MPI-процессу доступно примерно в T раз больше памяти.
2. Сокращается количество обменов данными, поскольку в T раз уменьшается количество участвующих в обмене данными процессов, а также уменьшается объем пересылки.
3. Не возникает простаивание процессов в ожидании своей очереди на доступ к разделяемым коммуникационным ресурсам узла.
4. Снижается нагрузка на файловую систему, поскольку количество процессов, параллельно выполняющих операции с файлами, также уменьшается в T раз.

Однако OpenMP также имеет некоторые особенности. Одна из основных проблем, это пересечение по данным между параллельными нитями. Присутствие критических секций и атомарных операций существенно снижает производительность. Избежать пересечения иногда можно простым способом, реплицируя данные для каждой нити, однако это ведет к росту потребления оперативной памяти. Более универсальный и эффективный способ, который был реализован, это двухуровневое разбиение расчетной области, когда подобласть MPI-процесса разбивается далее на подобласти OpenMP нитей. Элементы сетки переупорядочиваются таким образом, чтобы внутренние элементы подобластей были сгруппированы в памяти и отделены от интерфейсных элементов (т. е. элементов сетки, принадлежащих более чем одной OpenMP подобласти). Это позволяет локализовать пересечение по данным. Нити OpenMP параллельно обрабатывают данные, соответствующие внутренним элементам, а затем последовательно (или параллельно, но с наложением вычислений) обрабатываются интерфейсные элементы, по которым могут быть пересечения. При этом, поскольку количество интерфейсных элементов, как правило, намного меньше, чем внутренних, достигается высокая параллельная эффективность.

4. Производительность двухуровневого распараллеливания на суперкомпьютере Ломоносов

В качестве тестов были взяты расчеты реальных фундаментальных задач 1) "струя, набегающая на цилиндр" и 2) "течение вокруг конечного цилиндра", предназначенных для исследования акустических источников в турбулентном следе. Данное исследование направлено на разработку методов снижения аэродинамического шума самолетов. В расчетах используется явная схема повышенного порядка точности. Моментальные картины течения для тестовых задач представлены для наглядности на рис. 1.

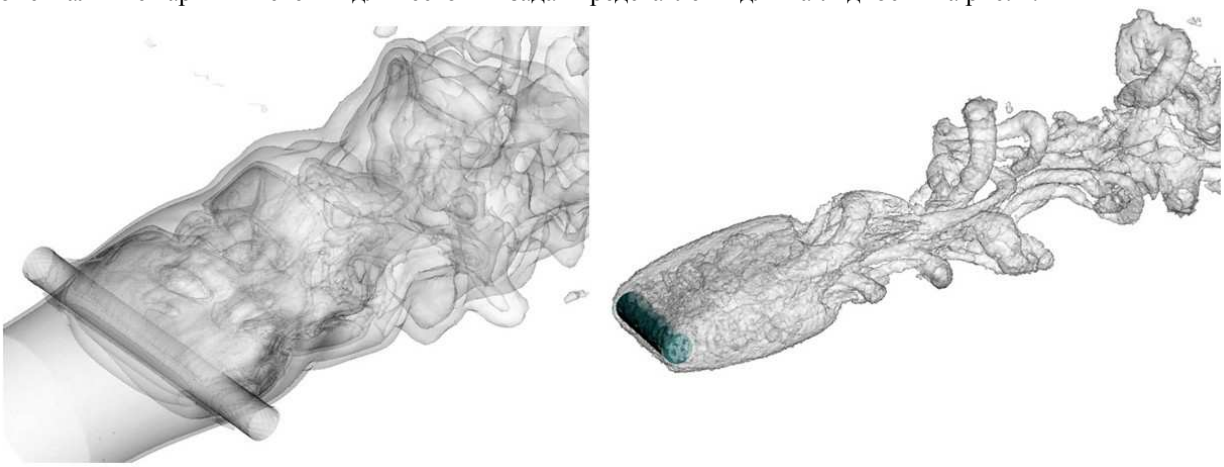

Рис. 1. Моментальные картины течения для задачи "струя, набегающая на цилиндр" (сверху) и "течение вокруг конечного цилиндра" (снизу).

Тест на ускорение с OpenMP выполнен для первой задачи на огрубленной сетке 2047992 узлов. Число MPI процессов фиксировано и равно 128, число нитей варьируется от 1 до 8. Было получено ускорение 5.32, что соответствует эффективности 66%. Результаты показаны на рис. 2.

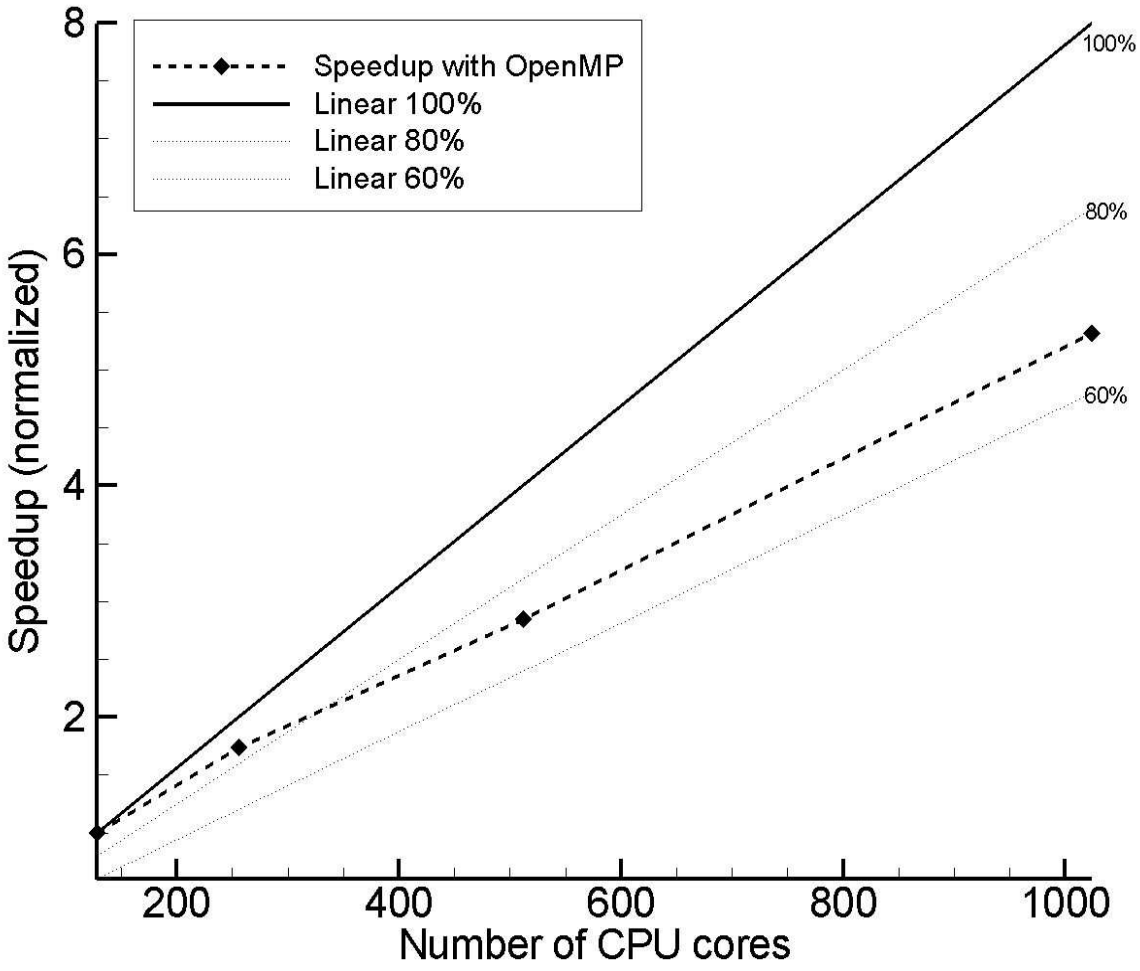

Рис. 2. Ускорение OpenMP (128 MPI процессов)

Сравнение MPI и MPI+OpenMP выполнено для первой и второй задачи на огрубленных (чтобы исчерпать параллелизм на доступном числе процессоров) сетках . Размеры сеток у тестовых задач отличаются более чем в два раза, что позволяет оценить влияние удельного объема вычислений на процессорное ядро на соотношение производительности подходов MPI и MPI+OpenMP. Результаты представлены на рис. 3 для первой (справа) второй задачи (слева). Показано сравнение MPI распараллеливания и распараллеливания MPI+OpenMP с количеством нитей на один MPI процесс равным 4 и 8.

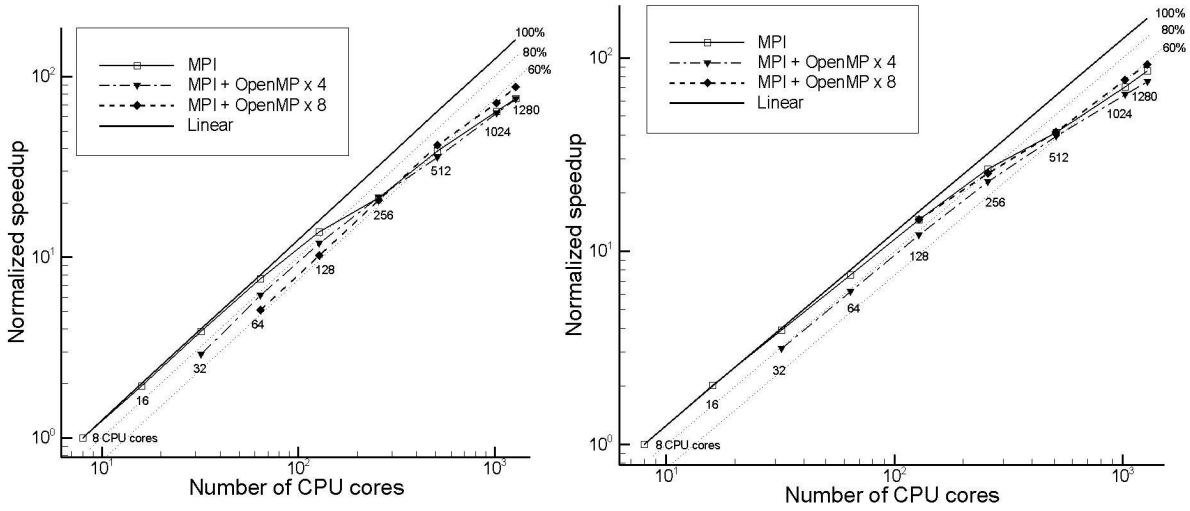

Рис. 3. Сравнение MPI и MPI+OpenMP, логарифмическая шкала. Сетка 783650 узлов (слева) и 2047992 узлов (справа)

Как видно из результатов, MPI более эффективен на небольшом числе процессоров, когда вычислительная нагрузка на ядро достаточно велика и обмены данными не оказывают существенного влияния. На большом числе процессоров гибридное распараллеливание становится более эффективным. Таким образом, гибкая конфигурация распараллеливания позволяет адаптировать комплекс программ NOISETTE к различным вычислительным системам и к различному числу процессоров посредством изменения количества нитей OpenMP на MPI процесс. Для второй задачи выигрыш в производительности составил более 10% на 1280 процессорах. Сравнение графиков, представленных на рис. 3 показывает, что при увеличении размера сетки, то есть при увеличении вычислительной нагрузки на ядро, точка пересечения графиков для MPI и MPI+OpenMP смещается в сторону увеличения числа процессоров. Таким образом, применение OpenMP в дополнение к MPI наиболее эффективно при большом числе процессоров и это число растет с увеличением размера сетки.

Тест на ускорение с двухуровневым распараллеливанием MPI+OpenMP для первой задачи был выполнен на неогрубленной сетке 16 миллионов узлов. Ускорение, нормированное по времени на 64 ядрах, составило 70.4 на 6400 ядрах. Результаты показаны на рис. 4. Один шаг по времени 4-шаговой схемы Рунге-Кутта занимал 26.8 и 0.38 секунды на 64 и 6400 ядрах соответственно.

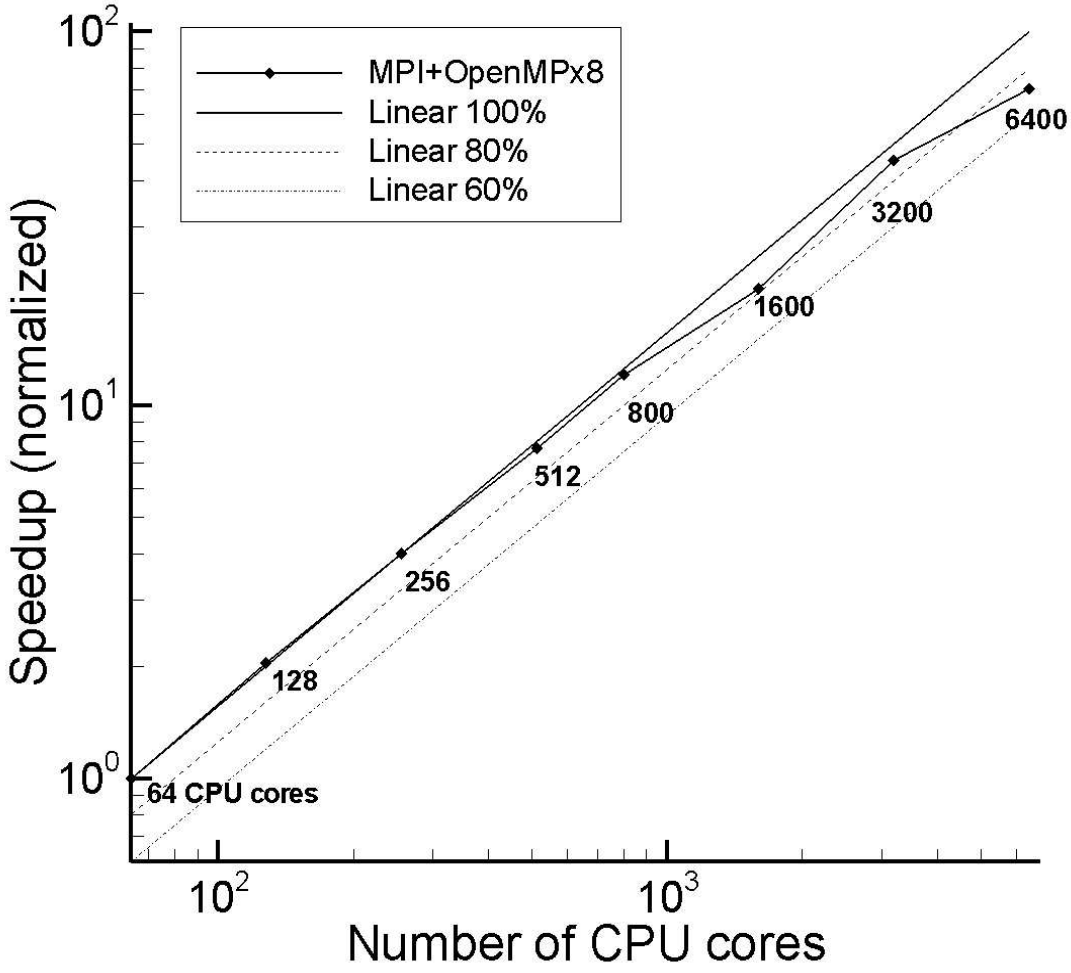

Рис. 4. Ускорение с MPI+OpenMP, сетка 16 миллионов узлов, логарифмическая шкала.

5. Оценка возможностей вычислительного ядра и инфраструктуры

На основе полученных ранее данных для MPI распараллеливания, представленных в [5], и новых данных для двухуровневого MPI+OpenMP распараллеливания, представленных в данной работе, можно сделать выводы об эффективном диапазоне чисел процессорных ядер и о максимальном размере тетраэдральной сетки. Как было показано в [5], только с использованием MPI, достаточно высокая параллельная эффективность достигается на числе ядер до нескольких тысяч даже для сравнительно небольших сеток. Применение OpenMP в дополнение к MPI позволяет расширить эффективный диапазон примерно в 8 раз на суперкомпьютере, имеющем 8-ядерные узлы, как например "Ломоносов". Таким образом, можно сделать вывод о применимости вычислительного ядра NOISETTE для расчетов на нескольких десятках тысяч процессоров.

Применение распределенного формата хранения топологии сетки и улучшенных параллельных средств обработки сеточных данных позволяет использовать в расчетах тетраэдральные сетки, содержащие до

нескольких сотен миллионов узлов (более миллиарда тетраэдров). Возможность расчета на сетке 200 миллионов узлов была также продемонстрирована в [5]. Функциональность средств обработки сеточных данных была дополнена возможностью построения двухуровневого разбиения. Последовательные средства генерации сеток (открытые или коммерческие, как например Gambit) позволяют стоить сетки до нескольких миллионов узлов. Построение сетки по блокам и последующая сшивка позволяет строить сетки порядка нескольких десятков миллионов узлов. Для получения сеток с сотнями миллионов используется параллельный алгоритм равномерного измельчения [6], каждый шаг которого увеличивает число узлов в 8 раз. Для разбиения сетки реализован параллельный алгоритм декомпозиции, использующий библиотеку ParMetis [7]. В рамках данной работы алгоритм декомпозиции был дополнен возможностью построения для каждой подобласти списков тетраэдров в локальной расширенной нумерации, которая включает все тетраэдры подобласти плюс интерфейсные тетраэдры, содержащие узлы из соседних подобластей до необходимого уровня. Эти списки используются для последовательной декомпозиции второго уровня для каждой подобласти. Таким образом, обеспечивается возможность построения и двухуровневой декомпозиции сеток размером порядка нескольких сотен миллионов узлов.

6. Заключение

Реализованное двухуровневое распараллеливание позволило повысить параллельную эффективность и расширить возможности комплекса программ NOISETTE, который теперь может применяться для крупномасштабных расчетов на сетках с числом узлов до нескольких сотен миллионов с использованием нескольких десятков тысяч процессорных ядер суперкомпьюетра. Следует отметить, что вместо OpenMP распараллеливание на втором уровне могло бы быть выполнено средствами POSIX threads или того же MPI. Выбор был сделан из соображений простоты реализации и требований переносимости кода. Параллельная эффективность, а также применение NOISETTE для актуальных задач газовой динамики и аэроакустики были продемонстрированы на суперкомпьютере "Ломоносов" МГУ.

ЛИТЕРАТУРА:

1. Abalakin, A. Dervieux, and T. Kozubskaya. Computational Study of Mathematical Models for Noise DNS. – AIAA-2002-2585 paper.
2. Ilya Abalakin, Alain Dervieux, and Tatiana Kozubskaya. High Accuracy Finite Volume Method for Solving Nonlinear Aeroacoustics Problems – In Proc. of East West High Speed Flow Field Conference, October 19–22, 2005, Beijing, China, pp. 314–320.
3. Abalakin, I.V., Dervieux, A., and Kozubskaya T.K. A vertex centered high order MUSCL scheme applying to linearised Euler acoustics – INRIA report RR4459, April 2002.
4. Debiez, C., and Dervieux, A. Mixed element volume MUSCL methods with weak viscosity for steady and unsteady flow calculation – Computer and Fluids, Vol. 29, 1999, pp. 89-118.
5. Г.И.Савин, Б.Н.Четверушкин, С.А.Суков, А.В.Горобец, Т.К.Козубская, О.И.Вдовикин, Б.М.Шабанов, “Моделирование задач газовой динамики и аэроакустики с использованием ресурсов суперкомпьютера МВС-100К”, – Доклады академии наук, 2008, том 423, №3, с. 312-315.
6. Sukov S.A. Iakobovski M.V., Boldyrev S.N. Big Unstructured Mesh Processing on Multiprocessor Computer Systems. Parallel Computational Fluid Dynamics: Advanced numerical methods software and applications. Proc. of the Parallel CFD 2003 Conference Moscow, Russia (May 13-15, 2003), Elsevier, Amsterdam, pages 73-79, 2004.
7. Parallel static and dynamic multi-constraint graph partitioning. Kirk Schloegel, George Karypis, and Vipin Kumar. Concurrency and Computation: Practice and Experience. Volume 14, Issue 3, pages 219 - 240, 2002.