РАЗРАБОТКА МАСШТАБИРУЕМЫХ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ОБРАБОТКИ БИОПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Н.Н. Попова, С.А. Комаров, Д. Воробьев, Д. Лопухина

Биоинформатика и моделирование лекарственных препаратов являются одними из самых быстрорастущих областей, где существует острая необходимость использования высокопроизводительных вычислений на суперкомпьютерах. Актуальность исследований в области биоинформатики состоит в том, что прорывы в сфере новых методов медицинской диагностики, лечения, создания новых лекарств и т.д. возможны только с развитием новых вычислительных моделей и платформ, учитывающих специфику био-медико-фармацевтических задач.

Существует огромное количество программных пакетов, решающих те или иные задачи в области биоинформатики. Приведем в качестве примера наиболее известные из них:

ClustalX — множественное выравнивание нуклеотидных и аминокислотных последовательностей

BLAST — поиск родственных последовательностей в базе данных нуклеотидных и ам инокислотных последовательностей

UGENE — свободный русскоязычный инструмент, множественное выравнивание нуклеотидных и аминокислотных последовательностей, филогенетический анализ, аннотирование, работа с базами данных.

И др.

Многие специалисты в области биоинформатики для решения своих задач пытаются использовать уже существующие пакеты, но данный подход не всегда приводит к необходимым результатам, т.к. существующие пакеты, как правило не учитывают особенности того или иного суперкомпьютера и могут решать поставленные задачи намного дольше, чем того требует ситуация. Поэтому нужны программные продукты, в которые заложены алгоритмы учитывающие структуру используемых массивно-параллельных вычислительных систем. Кроме того нужны системы, которые будут позволять биологам, не имеющих специальных навыков работы с суперкомпьютером, производить на них расчеты без особых сложностей.

Параллельный алгоритм спектрального метода поиска повторов в биопоследовательностях

Алгоритм, описаный далее, является параллельным алгоритом решения задачи нахождения повторов в биопоследовательностях спектральным методом.

Рассмотрим отдельно этапы получения профилей последовательностей, спектрального сравнения и спектрального индексирования профилей.

Получение профилей последовательностей

Распараллеливание на этапе построения профилей последовательностей производится по окнам. Рассмотрим параллельное построение профиля для одной последовательности, для второй действия абсолютно аналогичны. Главный процесс считывает последовательность себе в память, подсчитывает общее количество окон профиля в последовательности и распределяет их между всеми процессами приложения наиболее равномерным способом. При этом возникает необходимость рассылать исходную хромосому «с перекрытиями», что позволяет не терять окна профиля. При рассылке последовательности используется коллективная операция MIP_Scatterv. Далее каждый процесс строит профиль по полученной части последовательности.

Спектральное индексирование профилей последовательностей

Перед тем как выполнить спектральное индексирование профилей, процессы обмениваются недостающими кусками профилей с соседними процессами. При этом используется операция MPI_Sendrcv, в которой каждый из процессов отправляет пердыдущему процессу часть своего профиля и принимает часть профиля от следующего процесса. Далее выполняется спектральное индексирование имеющегося у каждого процесса профиля.

Спектральное сравнение профилей последовательностей

В начале этого этапа матрицы индексов собираются на главном процессе при помощи MPI_Gatherv. Далее все процессы представляются в виде прямоугольной сетки. не ограничивая общности, рассмотрим работу алгоритма на примере 16 процессов. Общая схема распараллеливания спектрального сравнения профилей для 16 процессов представлена на рис.2-4.

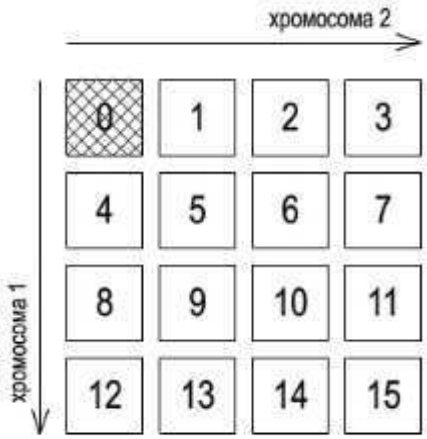

Рис. 2

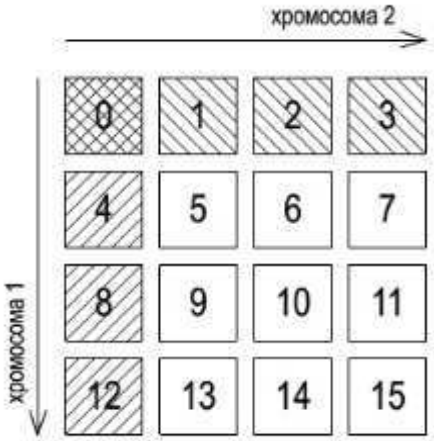

Рис. 3

Главный процесс распределяет полученные матрицы индексов для первой и второй последовательностей наиоблее равномерным способом между процессами, находящимися в первой стобце и первой строке сетки, соответственно. После описанных рассылок получаем схему, изображенную на рис.3

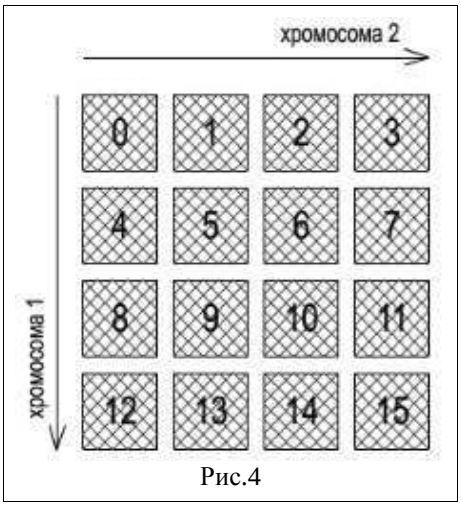

Рис.4

Далее, каждый процесс первой строки, рассылает полученную часть матрицы индексов всем процессам своего столбца. Затем, каждый процесс, находящийся в первом столбце рассылает имеющуюся часть матрицы каждому процессу своей строки. После проделанных операций у каждого процесса появляются матрицы индексов, для его частей двух последовательностей (рис. 4).

Далее каждый процесс производит сравнение коэффициентов его частей матриц индексов, и полученная информация по полученным повторам записыввется в массивы определенной структуры. Учитывая тот факт, что каждый процесс хранит только часть матрицы коэффициентов, получается так что, что могут быть повторы, которые разнесены по разным процессорам, в следствии чего возникает новая задача – задача склейки частей повтора воедино.

Для дальнейших операций (операций склейки) потребовалось все повторы разделить на несколько типов:

- Тип 0: внутренний повтор - повтор у которого ни начало ни конец не лежат на границе имеющейся матрицы
- Тип 1: повтор без конца – повтор, у которого конец лежит на границе матрицы
- Тип 2: повтор без начала – повтор, у которого начало лежит на границе матрицы
- Тип 3: повтор без начала и без конца – повтор, у которого и начало и конец лежат на границе матрицы

После того как процессоры обменяются друг с другом повторами, произойдет склейка повторов, которые являются частями одного и того же большого повтора. Повторы типа 1 могут быть склеены с повторами типа 2 (итоговый повтор будет иметь тип 0) и типа 3 (итоговый повтор будет иметь тип 1) . После того как все возможные склейки произойдут все повторы станут повторами типа 0 и будут готовы к выводу.

Очевидно, что для любого количества процессоров алгоритм описывается аналогично. Изменение числа процессов вляет лишь на количество строк и стобцов в сетке процессов на этапе спектрального сравнения и на размеры профилей и последовательностей, обрабатываемых каждым процессом.