

Потенциальные возможности использования распределенных вычислительных систем при решении концептуальных проблем построения информационных комплексов обработки данных высокопроизводительного геномного секвенирования и глубокого протеомного профилирования

УДК 004.9 ББК Ч23

Potential capabilities of using distributed computing systems in solution of conceptual problems for building processing complexes for high throughput computing for processing genomic sequencing and proteomic profiling data.

Авторы

ФИО (полностью), место работы, должность, ученую степень, ученое звание, контактные данные (e-mail).

Тягт Александр Викторович, ФГБУН НИИ ФХМ ФМБА России, младший научный сотрудник, at@niifhm.ru

Русаловский Илья МСЦ РАН, стажер исследователь, vikirus@gmail.com

Попенко Анна Сергеевна, ФГБУН НИИ ФХМ ФМБА России, лаборант-исследователь, a.s.popenko@niifhm.ru

Кострюкова Елена Сергеевна, ФГБУН НИИ ФХМ ФМБА ,зав. лаб., к.б.н.Б el-es@yandex.ru,

Алексеев Дмитрий Глебович, ФГБУН НИИ ФХМ ФМБА России, научный сотрудник, к.б.н., alexeev@niifhm.ru

Аннотация (рус, eng)

Современные исследования в области молекулярной биологии требуют значительных вычислительных ресурсов. В особенности это касается высокопроизводительного геномного секвенирования и протеомного профилирования, т.к. в процессе эксперимента генерируются терабайты данных, подлежащих специфической обработке. В рамках данной работы был создан алгоритм обработки результатов секвенирования с приборов высокопроизводительного секвенирования SOLiD и Ion Torrent, включающий в себя фильтрацию прочтений секвенатора по качеству, картирование на каталог шаблонных (референсных) последовательностей и количественное профилирование бактериального состава метагенома. Модуль картирования был оптимизирован для распределенных вычислений. Также был реализован алгоритм автоматизированной обработки масс-спектров, полученных в протеомных экспериментах. Демонстрируется хорошая масштабируемость предложенных решений и применимость подхода распределенных вычислений на удаленных кластерах для растущих потребностей биоинформатики.

Contemporary research molecular biology requires significant computational resources. Especially it concerns high-throughput genomic sequencing and proteome profiling. The project resulted in implementation of algorithm that proceeds sequencing results from HT

sequencing machines SOLiD and Ion Torrent, which includes quality read filtering, mapping them to reference sequences and quantitative profiling of bacterial composition of metagenome. Mapping module was optimized for distributed computing. Moreover, an algorithm was implemented, that automatically processes mass-spectra yielded from proteomic experiments. A good scalability of the suggested solutions and applicability of distributed computations on remote clusters for growing bioinformatic demands are shown

Ключевые слова (рус, eng)

распределенные вычислительные системы, высокопроизводительное секвенирование, метагеномный анализ, биоинформатика
distributed computational systems, high-throughput sequencing, metagenomic analysis, bioinformatics

Введение

Развитие новых технологий в биологии и переход на компьютеризованный анализ экспериментальных данных привел к развитию биоинформатики как важной составляющей в процессе научного познания мира живой материи [Attwood, 2011]. В современной молекулярной и вычислительной биологии одним из основных источников потока данных является геномика, изучающая ДНК-тексты - последовательности геномов живых организмов - на основе их частичных прочтений (ридов, англ. reads). Прочтение осуществляется приборами секвенирования. На сегодняшний день рынок оборудования высокопроизводительного секвенирования составляет более 800 млн.долларов в год, по оценкам JP Morgan [JP Morgan Next-Gen Sequencing report, 2011], при этом прирост годовых показателей составляет от 18 до 20%. Отмечаются также следующие тенденции в структуре затрат на высокопроизводительные анализы: во-первых, при сохранении бюджетов мировых потребителей оборудования, стоимость реагентов снижается до 50% за год (что вызвано отчасти появлением новых технологий). Во-вторых, утилизация оборудования также возрастает на 15-20% в год, что приводит в итоге к росту объема получаемой информации по результатам анализов секвенирования.

Одной из особенностей текущего положения в области геномики является стихийное накопление геномных данных. В версии GenBank [<http://www.ncbi.nlm.nih.gov/genbank/>] (мирового репозитория геномных данных) от августа 2012 содержится более 156 миллионов геномных последовательностей общим объемом более 143 миллиардов пар нуклеотидов. Однако рост объема получаемой информации не эквивалентен росту знаний, которые можно извлечь из этой информации. Реконструкция геномного текста подразумевает использование таких алгоритмов, как сборка *de novo* (с нуля) - объединения ридов, полученные в ходе эксперимента, в протяженные контиги (большие неразрывные участки), являющиеся частями исходного генома, а также выравнивание (картирование) ридов на шаблонную последовательность с целью выявления сходств и

различий на уровне нуклеотидного состава. Наглядно тренды приведены на рис.1. по данным исследования метагеномных проектов [Temperton, Giovannoni, 2012].

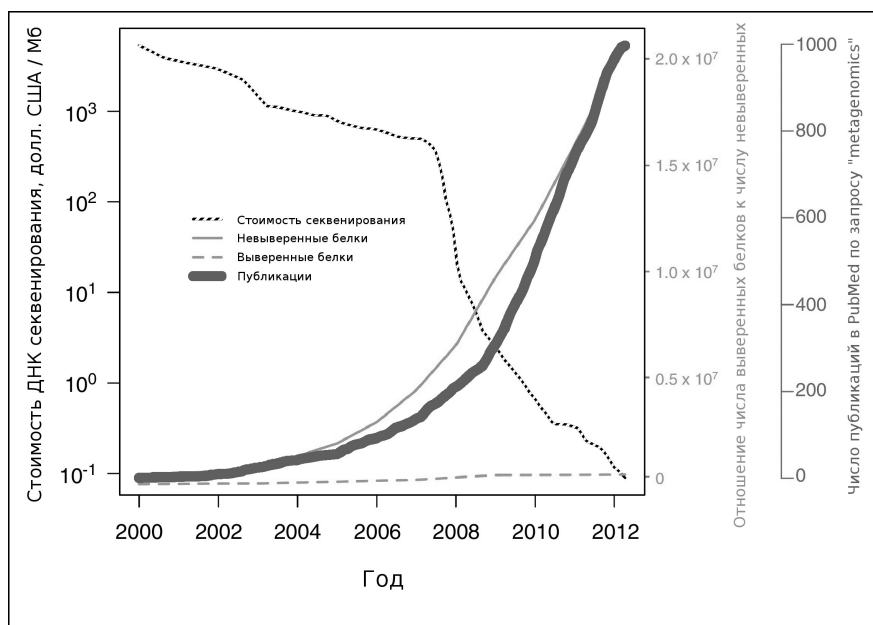


Рис.1 . Рост геномного секвенирования, падение стоимости секвенирования и прирост необработанной информации .

Основной проблемой обработки данных на сегодняшний день становится обеспечение квалификации персонала, выполняющего расчеты, и вычислительные ресурсы, необходимые для выполнения расчетов. Мировая тенденция в расчетах такого рода в последнее время заключается в создании распределенных вычислительных сервисов для обработки данных. На конец 2012 года существует более десяти ресурсов, представляющих разного рода сервисы для полной или частичной информационной обработки данных секвенирования – как платных, так и бесплатных. Одним из наиболее производительных является RAST - ресурс Американского Института Биотехнологической Информации (NCBI) – представляющий удаленный доступ как к вычислительным, так и алгоритмическим ресурсам [Aziz и др., 2008].

В настоящее время выделяют несколько специфичных вычислительных архитектур, каждая из которых имеет свои преимущества перед другими при решении определенных задач [Sadashiv, Kumar, 2011]. Если основной целью является получение результата работы конкретного алгоритма за минимальное время, то очевидное решение - выделить независимые части алгоритма и выполнять их параллельно на различных вычислителях. Наиболее часто рассматривается параллелизм по данным. В этом случае можно выделить два типа задач - с сильной зависимостью по данным и слабой зависимостью. Задачи со слабой зависимостью легко разделяются (без существенных изменений последовательного алгоритма) на множество независимых подзадач, выполняющихся над различными

наборами входных данных. В случае же сильной зависимости, требуется применение специальных приемов программирования. Кроме того, крайне желательно использование определенных вычислительных архитектур.

Существующие технические решения для высокопроизводительных вычислений включают в себя суперкомпьютеры, облачные и GRID технологии. Наиболее классический вариант - суперкомпьютеры, объединяющие множество сходных (или полностью идентичных) вычислительных узлов быстрой коммуникационной сетью. Такая схема позволяет решать задачи, требующие постоянного обмена промежуточными данными между процессами в ходе вычисления гораздо быстрее, нежели остальные, так как предоставляет возможность прямой пересылки информации между узлами. Внешнее хранилище обычно используется только для первоначального ввода данных и последующего вывода результатов. Применение такой схемы для расчета задач с малой зависимостью по данным не оправдано, так как отсутствует необходимость в дорогом и важном компоненте суперкомпьютера - низколатентной сети.

Другое крайне популярное в последнее время решение - облачные вычисления. С технической точки зрения они представляют собой множество выделяемых по требованию виртуальных вычислительных узлов с заданными характеристиками. Возможность прямой передачи данных между узлами отсутствует, ибо основная задача - обеспечить поддержку большого потока данных между облаком и клиентом. Облака обычно включают распределенную систему хранения данных, доступ к которой с выделенных виртуальных узлов так же происходит на высокой скорости. Облачные технологии идеально подходят для задач с хорошо параллелизуемыми алгоритмами. Современные решения предлагают высокий уровень сервиса, в частности, ускорить вычисления с использованием технологии map-reduce не составляет никакого труда, кроме того существуют стандартные API для автоматизированного использования предоставляемых возможностей. Однако, неизбежная загрузка и выгрузка данных по каналам общего пользования может занимать значительное время, что является неприемлемым в случае большого потока данных, способного постоянно загружать вычислитель. Более того, активное использование виртуализации может сводить на “нет” применение оптимизирующих компиляторов.

Под GRID - технологиями обычно понимают объединение множества децентрализованных вычислительных систем, управляемых независимо друг от друга [Foster и др., 2008]. Таким образом, организации могут получать значительные вычислительные мощности, не затрачивая средств на оборудование, путем использования компьютеров волонтеров. Наиболее известна серия проектов @Home (SETI@Home, Folding@Home, и т.д.). GRID'ы, так же как и “облака”, хорошо подходят для задач с идеальным параллелизмом по данным. Но отсутствие контроля над вычислительными узлами, в том числе, над их доступностью может доставлять значительные неудобства. Кроме обычных пользовательских компьютеров, GRID'ом может являться объединение крупных суперкомпьютерных центров, например, LHC Computing Grid. Такие

специализированные решения можно приспособить для классических HPC задач; проблемы контроля и доступности практически отсутствуют, географически удаленные суперкомпьютеры объединяются выделенными высокоскоростными каналами связи, что упрощает передачу больших объемов данных.

В составе всех перечисленных схем присутствует множество вычислительных узлов. В некоторых узлах, кроме процессора общего назначения, могут находиться дополнительные вычислительные модули (графические ускорители, программируемые вентильные матрицы - FPGA), позволяющие значительно повысить производительность. Системы, содержащие такие компоненты, называются гибридными. Плюсы подобных решений очевидны, графические ускорители, при своей небольшой стоимости, дают прирост, сравнимый с использованием небольшого кластера. Однако, программирование таких гибридных систем сопровождается большими трудностями, связанными с необходимостью учитывать особенности реализации конкретной архитектуры. Перед началом расчетов необходимо копирование данных в память видеоадаптера, объем которой может быть значительно меньше объема памяти узла, что может сделать применение ускорителей нецелесообразным для некоторых задач, либо, требовать существенной переработки алгоритма.

Оптимальным решением для вычислительных конвейеров обрабатывающих геномные и протеомные данные становится множество вычислительных узлов объединено в вычислительное «облако» под управлением общего планировщика расчетных задач. Планировщик является интеллектуальной компонентой системы, осуществляющей оптимальное распределение различных типов задач и данных между узлами, в соответствии с техническими характеристиками узла. После обчета, результаты от узлов компонуются планировщиком и передаются следующим этапам конвейера. Учитываемые при распределении характеристики включают в себя:

- тип процессорной архитектуры,
- объем оперативной памяти и дискового пространства,
- пропускную способность канала,
- квоту использования (в случае коллективного пользования узлом).

Адаптивность распределения задач между узлами существенна, т.к. задачи, порождаемые геномным конвейером, значительно различаются по требованиям к компьютеру, на котором будут обчитываться.

Проблематика

Одним из важнейших вопросов, встающих при анализе существующего в России высокопроизводительного оборудования является занятость ресурсов. По оценкам загруженность составляет не более 60%, и это считается хорошим показателем. Вопрос дозагрузки вычислительного ресурса до полной мощности может быть решен созданием

центральной системой планирования при использовании однотипных задач – и именно современная биоинформатика представляет такого рода данные – потоки информации от сотен лабораторий должны быть обработаны стандартными алгоритмами, и именно скорость обработки (отсутствие очереди) является критичным фактором. Мы проанализировали задачи возникающие в стандартной геномно-протеомной лаборатории и показали возможности масштабирования и параллелизации расчетов без потери скорости.

Для наглядности мы выбрали потоки данных в крупнейшем на сегодняшний день Российском геномном проекте «Метагеном.ру» [Метагеном.ру, 2012]. В ходе реализации проекта от 200 пациентов был собран генетический и белковый материал – соответствующий бактериальному составу кишечника, общее количество полученных цифровых данных на сегодняшний день превышает 20 Тб.

Решения

Нами был разработан конвейер позволяющий в непрерывном режиме обрабатывать потоки экспериментальной информации. В процессе разработки мы выделили обработку геномных данных как наиболее ресурсоемкую часть. Объем поступающих данных на два порядка превосходит данные протеомного анализа, среднее время обработки геномных данных для одного образца в 10-20 раз превосходит время на обработку одного образца протеомных данных, поисковая специфика проводимых исследований приводит к необходимости повторять анализ до 10 раз (из-за использования разных параметров).



Рис. 2. Схема высокопроизводительного конвейера

Основные этапы обработки приведены на рис.2, наиболее ресурсоемкие их них:

1) предобработка ридов, фильтрация и коррекция ошибок (ресурсоемкий процесс, хорошо параллелизуется, однократный)

-

2) выравнивание – ресурсоемкий процесс, происходит большое количество раз, хорошо параллелизуется. Так как стадия отображения на референсный набор - хорошо параллелизуемая задача, которая не требует больших затрат оперативной памяти. планировщик направит задачи отображения на референс, по возможности, серверу с большим числом процессорных ядер, у каждого свое пространство оперативной памяти. 3) Сборка *de novo*. Для стадии сборки *de novo* технические требования отличаются: при использовании известных эффективных алгоритмов, задача плохо поддается параллелизации; сборка *de novo* будет выполняться на узле SMP (симметричный многопроцессорный узел) с большим объемом памяти, доступной каждому из процессоров.

Для исследования возможностей параллелизации и масштабирования, исходя из того что, 80% времени созданный пайплайн тратил именно на эту задачу, в качестве пробной было выбрано картирование. Кроме того, именно эта задача будет занимать все больше времени пропорционально ожидаемому росту данных.

Для демонстрации требований к вычислительным ресурсам было проведено исследование масштабируемости одной из типовых задач — картирование на референсную последовательность. Такой класс задач должен давать практически идеальное ускорение, так как имеется возможность выполнять выравнивание отдельных наборов ридов, с последующим слиянием результатов. Был произведен набор тестовых запусков программы Bowtie [Langmead и др., 2009], с различными входными данными и различным набором используемых в процессе вычисления ядер, измерялось время выполнения программы. Bowtie предоставляет возможность ускорения работы в пределах одного вычислительного узла, выполняя расчет в несколько параллельных потоков. Для расчета на нескольких узлах одновременно, входной набор данных разделялся на равные части, результирующие файлы объединялись. Время выполнения данных операций не измерялось, так как оно определяется производительностью системы хранения, и обычно мало (порядка минут) по сравнению с временем выравнивания. Результаты работы отражены на рис. 2 и в таблице 1. При измерениях в пределах одного узла, с помощью встроенных возможностей bowtie задействовалось 3, 6, 12 и 24 ядра. Далее производились запуски на 2, 4 и 8 узлах, с использованием всех доступных 24 ядер на каждом. Тесты проводились на входных файлах объемом 1, 1.5, 2.2 и 3.8 гигабайт. Время файловых операций не измерялось, что, скорее всего и вызывает нелинейность графиков при расчете на двух и более узлах. При увеличении количества узлов в два раза, объем необходимых для передачи на узел данных уменьшается не ровно вдвое, а чуть меньше, так как во всех тестах происходит чтение индексных файлов неизменного размера. Сеть и хранилище обеспечивают доступ к входным данным на максимальной скорости со всех 8 узлов одновременно (1Гбит на узел), но загрузка в этом случае близка к теоретическому максимуму (10Гбит на головном узле), следовательно, возможно высокое влияние задач других пользователей, выполняющихся в процессе тестирования. Файловая система также может давать снижение производительности на параллельных операциях. Таким образом, предел масштабируемости в результате тестирования не выявлен. Однако, большие объемы данных предъявляют высокие требования к производительности хранилища и сети. Использование сети Infiniband для файловых операций должно решить проблему пропускной способности Ethernet сети, более того, используемая архитектура системы хранения не позволит достичь максимальной скорости передачи данных.

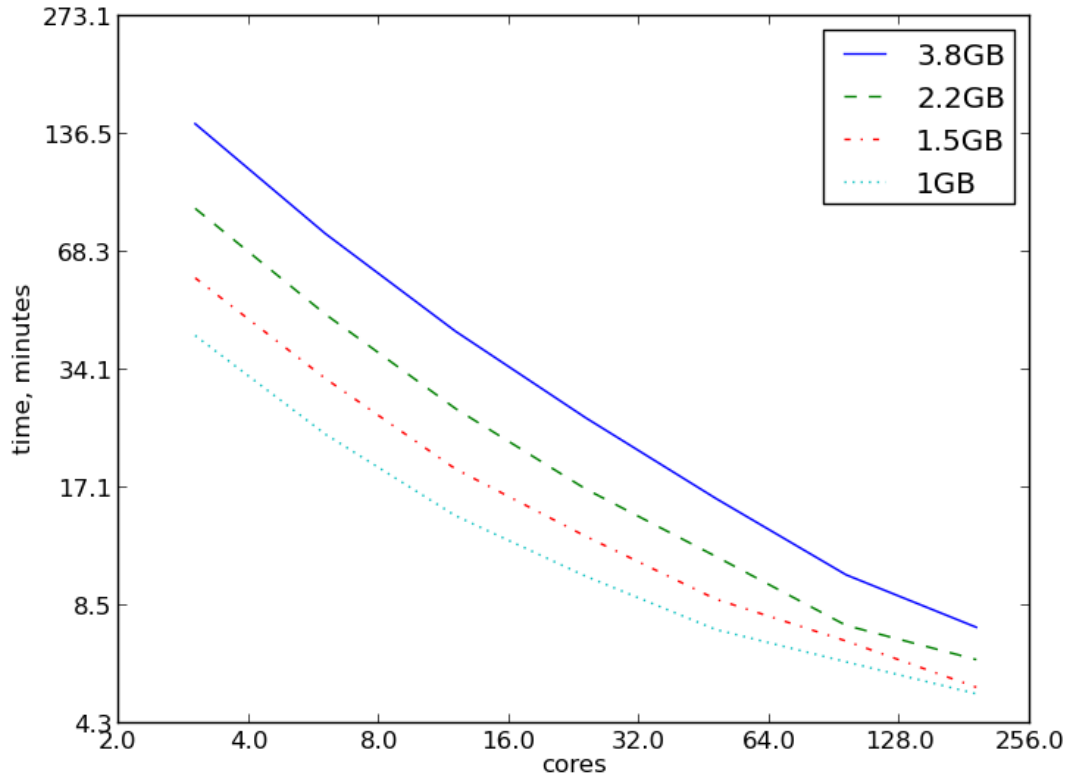


Рис. 3. График зависимости времени выполнения от количества потоков и объема входных данных

Таблица 1. (Цифры по рис. 3)

Количество ядер	1GB входных данных. Время, мин.	1.5GB. Время, мин.	2.2GB. Время, мин.	3.8GB. Время, мин.
3	42	59	88	145
6	23	32	47	76
12	14	19	27	43
24	10	13	17	26
48	7	9	11	16
96	6	7	8	10
192	5	5	6	8

Выводы

Ускорение расчетов алгоритмов метагеномного анализа должно учитывать различие их требований к аппаратной части. Таким образом, применение смешанной архитектуры имеет наибольший потенциал. Созданный для биоинформатического анализа данных программный конвейер был оптимизирован для вычисления на суперкомпьютере в части, наиболее легко поддающейся параллелизации на распределенных вычислителях. Он может быть легко перенесен на другие архитектуры, следует лишь учитывать проблему передачи больших объемов данных. С другой стороны, специфика таких плохо параллелизуемых модулей, как сборка *de novo*, требует узлов с большим объемом ОЗУ. Ресурсы такого рода практически невозможно получить в рамках GRID и "облачных" вычислений. Необходимо разработка новых классов алгоритмов сборки, которые имеют более гибкие требования к архитектуры распределенной системы.

На примере типовых рабочих объемов и форматов данных метагеномных данных показан близкий к линейному рост производительности с увеличением числа используемых потоков. Применимость оптимизированного программного комплекса был продемонстрирована в крупном научно-исследовательском проекте с участием нескольких исследовательских лабораторий. Предположительно, в силу специфики алгоритма, картирование может быть значительно оптимизировано с помощью средств массового параллелизма GPGPU. В этом может заключаться одно из направлений дальнейшего усовершенствования конвейера.

Выполнение данной работы была профинансировано Государственными контрактами 16.740.11.0371, 16.512.11.2111, 12-07-90008-Бел_а.

Литература

1. Attwood T.K. Concepts, historical milestones and the central place of bioinformatics in modern biology: a European perspective. : Intech Online Publishers, 2011.
2. Aziz R.K. и др. The RAST Server: rapid annotations using subsystems technology. // BMC genomics. 2008. Т. 9. С. 75.
3. Foster I. и др. Cloud Computing and Grid Computing 360-Degree Compared // 2008 Grid Computing Environments Workshop. : IEEE, 2008. С. 1–10.
4. Langmead B. и др. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. // Genome biology. 2009. Т. 10. № 3. С. R25.
5. Sadashiv N., Kumar S.M.D. Cluster, grid and cloud computing: A detailed comparison // 2011 6th International Conference on Computer Science & Education (ICCSE). : IEEE, 2011. С. 477–482.
6. Temperton B., Giovannoni S.J. Metagenomics: microbial diversity through a scratched lens // Current Opinion in Microbiology. 2012. Т. null. № null.
7. JP Morgan Next-Gen Sequencing report.

8. Метагеном.ру [Электронный ресурс]. URL: www.metagenome.ru.