ISSN 1814-1196
Научный вестник НГТУ
том 58, № 1, 2015, с. 148–160

http://journals.nstu.ru/vestnik
Science Bulletin of the NSTU
Vol. 58, No. 1, 2015, pp. 148–160

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

MODERN INFORMATION TECHNOLOGIES

УДК 004.75

Мультисерверный подход к высокопроизводительному вычислению молекулярных дескрипторов*

С.Р. ГАРСИА-ДЖАКАС[1], Т.В. АВДЕЕНКО[2]

[1] *Куба, Гавана, Университет информатики, магистр информатики, профессор. E-mail: crjacas@uci.cu*
[2] *630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, доктор технических наук, профессор. E-mail: tavdeenko@mail.ru*

Настоящая статья описывает новый модуль программного обеспечения QuBiLS-MIDAS для распределенного вычисления 3D мультилинейных алгебраических молекулярных дескрипторов. Программа QuBiLS-MIDAS первоначально имела следующие особенности: многоядерные вычисления, режим пакетной обработки, модуль очистки данных, возможности пользовательской конфигурации дескрипторов, полная кросс-платформность. Основной мотивацией для разработки нового модуля явилась вычислительная сложность, с которой столкнулись разработчики программы при вычислении дескрипторов на больших наборах данных. Для выполнения этой задачи была разработана мультисерверная вычислительная платформа T-arenal, которая предназначена для организаций, имеющих множество рабочих станций, связанных через локальную сеть, без выделения ресурсов, специально предназначенных для вычислительных задач. Составляющую front-end платформы T-arenal формирует графический пользовательский интерфейс. Эта новая система развернута на 337 рабочих станциях, принадлежащих вычислительным лабораториям Университета информационных наук г. Гавана, Куба, и идеально интегрируется с программным обеспечением QuBiLS-MIDAS. Система T-arenal была конфигурирована с корневым сервером и тремя серверами запроса, каждый с равным количеством клиентов (рабочих станций). Для иллюстрации применимости платформы T-arenal были проведены тесты производительности над набором данных, содержащим 15 000 соединений. При этом получено 52- и 60-кратное снижение времени обработки соответственно для 2-линейных и 3-линейных индексов. Таким образом, можно констатировать, что программное обеспечение T-arenal при использовании распределенных вычислений представляет собой эффективную стратегию для выполнения высокоскоростных расчетов 3D мультилинейных дескрипторов.

Ключевые слова: хемоинформатика, распределенная вычислительная система, мультисерверная архитектура, платформа распределенных задач, молекулярные дескрипторы, параллельные вычисления, графический пользовательский интерфейс, клиент-серверная архитектура

DOI: 10.17212/1814-1196-2015-1-148-160

---

* *Статья получена 1 декабря 2014 г.*

ВВЕДЕНИЕ

Хемоинформатика (химическая информатика) – это научная область, одной из задач которой является кодирование химической информации с использованием вычислительных методов с целью ускорения идентификации и/или оптимизации соединений с желаемыми свойствами [1]. С этой целью используется большое количество логико-математических методов, позволяющих кодировать особенности и свойства молекул числовыми величинами, называемыми молекулятными дескрипторами (MD) [2]. Известно несколько программных пакетов для вычисления MD, таких как, например, DRAGON [3], Mold2 [4], BlueDesc [5] PowerMV [6], PADEL [7] . За исключением PADEL эти программы не используют параллельные методы вычисления MD и тем самым не позволяют максимально повышать быстродействие используемых алгоритмов.

В недавней публикации [8] была разработана программа QuBiLS-MIDAS (Quadratic, Bilinear and N-Linear Maps based on N -tuple Spatial Metric Similarity Matrices and Atomic Weightings), использующая квадратичные, билинейные и N -линейные отображения на основе N -кратных пространственных метрических матриц подобия и атомных весов, которая вычисляет геометрические (3D) свободные от выравнивания дескрипторы для нековалентных отношений между N атомами молекулы, называемые *3D N-линейными MD* [9]. Эта программа имеет следующие особенности: многоядерные вычисления, режим пакетной обработки, модуль очистки данных, возможности пользовательской конфигурации дескрипторов, полная кросс-платформность и др. Данное программное обеспечение доступно по ссылке http://tomocomd.com/.

3D 2-линейные MD, как подсемейство *3D N-линейных MD,* используются в том случае, когда учитываются нековалентные отношения между двумя атомами. Эти отношения кодируются с использованием метрик схожести, а полученные значения представляются двумерными матрицами пространственной схожести. Кроме того, естественным расширением данного семейства моделей являются *3D 3-линейные* и *4-линейные MD,* позволяющие рассматривать отношения между тремя и четырьмя атомами с использованием мультиметрик (например, угол связи для описания отношений между тремя атомами или двугранный угол для описания отношений между четырьмя атомами). Эта информация концентрируется в трехмерные или четырехмерные пространственные матрицы схожести, что впервые было использовано в определении 3D молекулярных дескрипторов [9].

Как уже было отмечено, одной из основных особенностей ПО QuBiLS-MIDAS является возможность многоядерных вычислений. Однако этого недостаточно для удовлетворения потребностей высокоскоростного вычисления *3D N-линейных MD* на больших наборах данных. Например, в предыдущем исследовании [8] было замечено, что вычисление 20 280 *3D 4-линейных MD* для набора данных, включающего 1000 компонент, требует приблизительно 106 370 секунд (29 часов) с использованием 16 обрабатывающих нитей (8 родных ядер с многопоточной обработкой каждый). Несмотря на то что это время обработки значительно сокращено по сравнению

со случаем одного процессора (639 006 секунд), оно все еще достаточно высоко. В самом деле, если размер набора данных увеличивается на m молекул, то время обработки будет увеличено приблизительно в $106m$ раз. Таким образом, для таких MD необходимо рассмотреть другие вычислительные альтернативы.

В настоящее время распределенные вычисления являются распространенным методом обработки больших массивов данных. Так, системы, представляющие собой добровольные вычислительные проекты, такие как BOINC [10], OurGrid [11], Hadoop [12], Java Heterogeneous Distributed Computing (JHDC) [13] и др., базируются на принципе использования вычислительных ресурсов, доступных в сети, для обработки вычислительно сложных задач. Все эти системы поддерживают обработку Bag-of-Tasks (BoT), т. е. каждая задача разделяется на независимые подзадачи, которые обрабатываются узлами, связанными в систему.

Для реализации этой идеи была использована система JHDC по следующим причинам: 1) является полностью кроссплатформенной без перекомпиляции исходного кода; 2) использует модульный принцип, который позволяет, чтобы каждая компонента была изолирована и, таким образом, независимо разрабатывалась, модифицировалась и тестировалась; 3) одновременно поддерживает несколько вычислительных задач различной сложности; 4) разработана на языке Java и, таким образом, легко интегрируется с библиотекой для вычисления *3D N-линейных MD* [8]; 5) это система с открытым кодом; 6) имеет широкую сферу применения, в частности успешно используется в приложениях био- и хемоинформатики [14]; 7) система спроектирована для работы в среде INTRANET.

Однако, как почти все распределенные вычислительные платформы, система JHDC базируется на клиент-серверной архитектуре (*client/server architecture* – CSA) и, таким образом, не всегда пригодна для поддержки огромного числа связанных рабочих станций. В последнем случае подходящей альтернативой может быть использование множества серверов. Для этой цели предлагается многосвязная распределенная вычислительная платформа (Multi-Tiered Distributed Computing Platform). Эта платформа состоит из *n-арного дерева узлов*, в котором внутренние узлы работают как планировщики, а конечные узлы (листья) выполняют обработку. Однако, корректное развертывание и использование этой платформы может быть осложнено следующими факторами: 1) нет априорных знаний о том, каковы должны быть ширина и глубина дерева, чтобы гарантировать качественную обработку; 2) обрабатывающие узлы должны быть статистически скомпонованы к одному из многих планирующих узлов, и, таким образом, трудно выполнить начальную конфигурацию системы; 3) рекурсивное разбиение задачи на подзадачи для приложений био-хемоинформатики является платформо-зависимым. В данной работе предлагается метод решения обозначенных проблем путем создания программного обеспечения, посредством которого программа QuBiLS-MIDAS способна выполнять высокопроизводительные распределенные вычисления *3D N-линейных* молекулярных дескрипторов MD на больших наборах данных с использованием мультисерверного подхода.

1. ОПИСАНИЕ МУЛЬТИСЕРВЕРНОЙ РАСПРЕДЕЛЕННОЙ КОМПЬЮТЕРНОЙ ПРОГРАММЫ

Распределенная вычислительная система T-arenal, предлагаемая в данной статье, создана с целью управления использованием большого количества вычислительных ресурсов в локальной сети без ограничения на размер системы. Для этого предлагается мультисерверный подход, сочетающий децентрализованную (P2P) и клиент-серверную архитектуры в одной модели таким образом, что серверы могут работать вместе без потери вычислительной мощности [15]. T-arenal разделяется на два основных компонента: back-end и front-end. Посредством составляющей front-end пользователь получает доступ к функциональности системы, в то время как составляющая back-end ответственна за выполнение всех запросов, сделанных посредством составляющей front-end. Компонент front-end может быть графическим интерфейсом, обеспечиваемым программой T-arenal, либо любым другим разрабатываемым интерфейсом. Для обмена сообщениями между компонентами T-arenal была использована коммуникационная технология JAVA RMI (Remote Method Invoke) [16], а для обмена файлами, содержащими большие объемы данных, технология Java Sockets или Apache FTP.

Описание составляющей back-end. Мультисерверный подход

Составляющая back-end программного обеспечения T-arenal базируется на мультисерверной модели, организованной в виде дерева, состоящего из трех уровней (рис. 1). Она разделена на три компоненты ПО: корневой сервер (*root server)*, сервер запросов (*request server)* и клиент (*client)*. Корневой сервер создает связь между составляющими back-end и front-end и, таким образом, отвечает за управление и назначение запросов, выполняемых через front-end, на один из серверов запросов. Более того, он распределяет очередную вычислительную задачу на наиболее подходящий для ее обработки сервер запроса и определяет, когда сервер запроса должен обратиться к другому серверу для выполнения задачи. Кроме того, корневой сервер имеет механизм назначения клиентов системы на соответствующий сервер запроса в соответствии с правилами назначения, определенными пользователями. Таким образом, развертывание T-arenal осуществляется достаточно просто, так как клиенты всегда устанавливают соединение с корневым сервером для получения конфигурации (IP-адрес и порт) сервера запроса, с которым он взаимодействует.

С другой стороны, сервер запроса отвечает за разбиение назначенной ему задачи на более мелкие подзадачи (рабочие единицы) в соответствии с пользовательским алгоритмом, а также за сбор результатов, полученных от вычислений, выполненных клиентами, для построения конечного решения. Другой функцией сервера запроса является управление рабочими единицами, созданными, но еще не обработанными. Число серверов запроса, необходимое для использования в системе, определяется логической структурой, в соответствии с которой группируются рабочие станции (например, через подсети). Опять же новые серверы запроса могут быть динамически добавлены или удалены без влияния на распределенную систему.

Важно заметить, что сервер запроса может обрабатывать несколько задач с различными приоритетами в одно и то же время. Число задач может быть определено пользователем (по умолчанию равно единице), и его минимальное значение равно нулю. Эта нижняя граница показывает, что все ресурсы, принадлежащие серверу запроса, будут использованы при совместном выполнении задачи, размещенном на другом сервере запроса. Эта схема взаимодействия также используется, когда число задач меньше, чем число доступных серверов запроса. Сервер запроса, взаимодействующий в задаче, работает как посредник (прокси) между клиентами и сервером запроса, которому он помогает. Таким образом, благодаря этим многочисленным взаимодействиям в системе поддерживается высокая степень согласованности на основе CSA. В этом случае, если имеется m серверов запроса, каждый с n клиентами, и только один сервер запроса работает, он получает максимум $(m-1)+n$ запросов, что намного меньше, чем если бы все существующие клиенты $(m \cdot n)$ выполняли бы запросы одного и того же сервера.

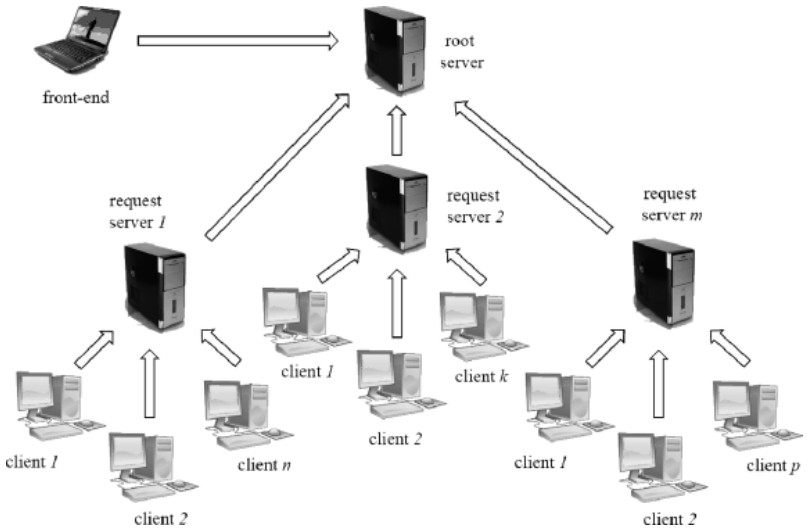

Рис. 1. Общая архитектура программной системы T-arenal

Наконец, модуль клиента отвечает за обработку рабочих единиц, назначенных соответствующему серверу запроса. Для этого клиент часто запрашивает рабочую единицу, выполняет ее обработку и возвращает результат серверу запроса, затем снова запрашивает рабочую единицу. Если в какой-то момент времени клиент не получает рабочую единицу для обработки, он входит в «спящий режим» на некоторый период перед выполнением следующей операции. Кроме того, если рабочая единица назначена, клиент не контактирует с соответствующим сервером запроса до тех пор, пока не выполнена подзадача или не возникнет исключительная ситуация. Все коммуникации между сервером запроса и его клиентами инициализируются последними. Если клиент теряет коммуникацию или оказывается неспособным работать с соответствующим сервером запроса, то он снова связывается с корневым сервером для получения конфигурации другого сервера запроса.

Библиотека T-arenal

Для программного взаимодействия с системой добавлена библиотека T-arenal. Эта библиотека полностью скрывает технические детали, топологию и протокол связи компьютерной системы от разработчиков и, таким образом, облегчает разработку приложений, использующих систему T-arenal. Имеется класс *SystemConnection*, который задает стартовую точку для установления связи с системой. Для построения экземпляра этого класса необходимо передать в качестве параметра информацию о конечном пользователе (username и password) и корневой сервер (IP-адрес и порт связи). Если объект класса *SystemConnection* успешно создан, то конечный пользователь аутентифицируется в системе и может получать доступ к соответствующему функционалу.

Составляющая front-end. Графический пользовательский интерфейс T-arenal

Графический пользовательский интерфейс T-arenal формирует составляющую front-end, обеспечиваемую платформой T-arenal. Это отдельное приложение соединяется и разъединяется с корневым сервером без влияния на работу системы. Его разработка выполнена с использованием API, а целью разработки явился доступ ко всему функционалу T-arenal. Например, владелец отдельной задачи может проводить мониторинг процесса, а также выгружать результаты вычислений в Zip архив. Более подробная информация изложена в руководстве, свободно доступном по адресу: http://tomocomd.com/ .

2. РАСПРЕДЕЛЕННАЯ ПРОГРАММНАЯ СИСТЕМА QUBILS-MIDAS

Задача распределенного вычисления *N*-линейных алгебраических молекулярных дескрипторов

Для оценивания эффективности мультисерверного подхода программа T-arenal была применена для вычисления *3D N-линейных MD* [10]. С этой целью была разработана библиотека программ. В данной библиотеке классы *QuBiLsDataManager* и *QuBiLsTask* составляют два основных класса приложений. Экземпляр первого класса получает на входе один или несколько наборов данных в форматах MDL MOL/SDF, а также один или несколько проектов, предварительно сохраненных в программном обеспечении QuBiLS-MIDAS [9]. В переопределенном методе *generateWorkUnit*, используемом для построения подзадач, для каждой рассматриваемой пары *набор_данных/проект* получается подмножество молекул и соответствующий дескриптор *3D N-линейный MD*. Это подмножество молекул выбирается в соответствии с параметром *гранулярности*, который задается пользователем в предоставляемом этим приложением конфигурационном файле (минимальное значение 1). Важно заметить, что *гранулярность* соответствует вычислительной сложности дескриптора MD, а также характеристикам рабочих станций, на которых строится T-arenal. При выборе подходящей *гранулярности* мы избегаем перегрузки в сети взаимных связей, так как

при этом достигается постоянная связь клиентов с соответствующим сервером запроса.

После того как рабочие единицы назначены, в каждом клиенте создается экземпляр класса *QuBiLsTask* для выполнения обработки. Полученные результаты посылаются обратно к экземпляру *QuBiLsDataManager* с помощью переопределенного метода *processResults*. Эти результаты сохраняются в одном CSV (Comma Separated Values) выходном файле в том же порядке, в котором создаются рабочие единицы. Однако результаты не обязательно получаются в вышеуказанном порядке, так как программа T-arenal спроектирована для работы в динамическом окружении. Поэтому, чтобы избежать этой ситуации, полученные результаты сохраняются в независимых CSV файлах. Распределенные вычисления завершаются, когда *3D N-линейные MD* вычисляются для всех компонент.

Модуль распределенного вычисления для ПО QuBiLS-MIDAS

Чтобы запустить распределенное приложение в системе T-arenal, необходимо загрузить на удаленный компьютер файлы *.class*, скомпилированные JAVA и соответствующие необходимым классам. Для этого в программном обеспечении QuBiLS-MIDAS был разработан модуль взаимодействия с распределенной системой. Таким образом, необходимо только выбрать опцию, указывающую, что *3D N-линейный MD* будет вычисляться в системе T-arenal. Далее высвечивается окно диалога, чтобы пользователь аутентифицировался в системе, и, если этот процесс успешно завершен, высвечивается другое окно для установки параметров вычисления. Как только задача сконфигурирована, автоматически создается *проблема* с именем *DistributedQuBiLS*, если таковая еще не записана в T-arenal. Далее от этой *проблемы* инициализируется конкретный распределенный вычислительный процесс с передачей набора данных и MD, выбранного пользователем. Наконец, пользователь может закрыть программу QuBiLS-MIDAS или выполнить другое вычисление локально на рабочей станции или с использованием платформы T-arenal.

Новый модуль также позволяет пользователю проводить мониторинг или останавливать выполнение определенного вычисления, а также выгружать или удалять из системы полученные решения. Кроме того, в меню T-arenal обеспечивается опция присоединения результатов, принадлежащих рабочим единицам, созданным в процессе вычисления, в единый выходной файл.

Тестирование производительности распределенных вычислений

Для выполнения распределенного вычисления *3D N-линейных* молекулярных дескрипторов система T-arenal с помощью *ghost image* (клонирование диска, используемое для развертывания программ одновременно на большом числе рабочих станций) была развернута на 337 рабочих станциях, принадлежащих вычислительным лабораториям Университета информационных наук г. Гавана, Куба. Система T-arenal была конфигурирована с корневым сервером и тремя серверами запроса, каждый с равным количеством клиентов (рабочих станций). Каждый сервер имеет 2.20 GHz Core(TM)2 Duo E4500

с 1 GB RAM и 160 GB памяти на жестком диске. Клиенты имеют разнообразное аппаратное обеспечение (см. табл. 1), операционную систему Nova GNU/Linux (https://humanos.uci.cu/nova/) и связаны в систему T-arenal посредством неспециализированной сети Ethernet LAN до 100 Mb/s.

Таблица 1

Особенности вычислительной системы T-arenal, связывающей 337 рабочих станций

Процессоры			RAM		
Тип	Количество	Процент	Объем (MB)	Количество	Процент
Pentium(R) 4	224	66.47	≤512	69	20.47
Celeron(R)	72	21.36	>512 и ≤768	37	10.97
Core(TM)2 Duo	25	7.41	>768 и ≤1024	21	64.39
Pentium(R) Dual-Core	14	3.26	>1024 и ≤1536	5	1.48
Pentium(R) D	1	0.30	>1536	9	2.67
Atom(TM)	1	0.30			

Для оценки быстродействия распределенного приложения были рассмотрены метрики *ускорение* и *эффективность*. Первая метрика вычисляется делением «*наилучшего» последовательного времени* (без распараллеливания) алгоритма на параллельное время, полученное с использованием *p* процессоров (рабочих станций). Вторая метрика вычисляется делением *ускорения* на соответствующее число используемых процессоров. Метрика *ускорение* показывает, в какой степени параллельный алгоритм улучшает свою последовательную версию с максимальным значением, соответствующим числу процессоров, используемых для исполнения алгоритма. Метрика *эффективность* представляет собой производительность (число между 0 и 1), с которой алгоритм использует *p* назначенных процессоров. Эти метрики обычно вычисляются для алгоритмов, выполняемых на специализированных системах с однородными архитектурами. Однако, из-за неоднородности вычислительных ресурсов, связанных в систему T-arenal (см. табл. 1), «*наилучшее» последовательное время* было определено на рабочей станции с наилучшими характеристиками на множестве компьютеров: Intel(R) Core(TM)2 Duo CPU E4500 2.20 GHz с 2 GB RAM.

Для оценивания быстродействия приложения было вычислено 12 480 *2-линейных* и 7488 *3-линейных* молекулярных дескрипторов на наборе данных, содержащем 15 000 структур (http://www.otavachemicals.com/-download-compound-libraries/cat_view/110-diversity-sets/133-primscreen-15). Важно подчеркнуть, что поскольку T-arenal развертывается в динамическом окружении, число рабочих станций, используемых для вычисления метрики *ускорение,* априори не может быть фиксированным. В этом смысле количество авторизованных клиентов (могут не обязательно быть включенными) для выполнения рабочих запросов постепенно увеличивается до достижения наибольшего

значения ресурсов, работающих над вычислениями. Используемое значение гранулярности было получено после нескольких внутренних тестов и равнялось шести молекулам для 2-линейных MD и двум молекулам для 3-линейных MD.

В табл. 2 показано время обработки (на молекулу и на дескриптор), ускорение и эффективность, полученные в процессе вычислений. Время обработки графически представлено на рис. 2. Как можно видеть, время обработки всегда уменьшается с увеличением числа процессоров (клиентов или рабочих станций). Можно увидеть, что последовательное время вычислений уменьшается с 49 349 секунд (13 часов) до 950 секунд (16 минут) и с 166 017 секунд (46 часов) до 2783 секунд (46 минут) при вычислении 3D 2-линейных и 3-линейных MD при использовании максимального числа клиентов соответственно 265 и 282.

Таблица 2

Результаты распределенного вычисления 2-линейных и 3-линейных QuBiLS-MIDAS MD

Число процессоров	Время обработки (с)	Ускорение	Эффективность	Время обработки на одну молекулу (с)	Время обработки на один дескриптор (с)
Показатели для 2-линейных QuBiLS-MIDAS MD					
1	49 349	1.000	1.000	3.290	3.954
27	6534	7.553	0.280	0.436	0.524
45	3697	13.348	0.297	0.246	0.296
71	2220	22.229	0.313	0.148	0.178
115	1571	31.412	0.273	0.105	0.126
157	1443	34.199	0.218	0.096	0.116
187	1238	39.862	0.213	0.083	0.099
265	950	51.946	0.196	0.063	0.0760
Показатели для 3-линейных QuBiLS-MIDAS MD					
1	166 017	1.000	1.000	11.068	22.171
66	11 773	14.102	0.214	0.785	1.572
93	8639	19.217	0.207	0.576	1.154
161	7339	22.621	0.141	0.489	0.980
179	6531	25.420	0.142	0.435	0.872
184	5346	31.054	0.169	0.356	0.714
190	4362	38.060	0.200	0.291	0.583
282	2783	59.654	0.212	0.186	0.372

С другой стороны, при анализе показателя *ускорение* обнаруживается, что этот показатель не пропорционален числу используемых процессоров. Аналогичное поведение мы наблюдаем и для показателя *эффективность*. В обоих случаях такое поведение можно объяснить тем фактом, что максимальное число используемых клиентов непостоянно в процессе вычисления, так как некоторые из них могут отказать (например, клиенты могут быть выключены). Кроме того, метрики *ускорение* и *эффективность* были вычислены с использованием «*наилучшего» последовательного времени* выполнения, определенного для архитектуры Core(TM)2 Duo, в то время как большинство рабочих станций имеют не такое хорошее аппаратное обеспечение (87.83 % Pentium(R) 4 и Celeron(R), см. табл. 1). Таким образом, эти реальные рабочие станции имеют меньшую вероятность быть доступными для выполнения вычислений и, следовательно, большее время вычислений, чем если бы мы имели все рабочие станции на основе архитектуры Core(TM)2 Duo.

Несмотря на хорошее достигнутое быстродействие, важно заметить, что распределенное вычисление *3D N-линейных MD* не всегда предпочтительно. Иногда вычисление дескрипторов быстрее выполнить на единственной рабочей станции, чем в распределенной системе, в частности, для небольшого количества изучаемых веществ и малого количества вычисляемых индексов. Поэтому разработанный модуль для ПО QuBiLS-MIDAS рекомендуется для случая, когда мы рассматриваем тысячи веществ и/или дескрипторов.

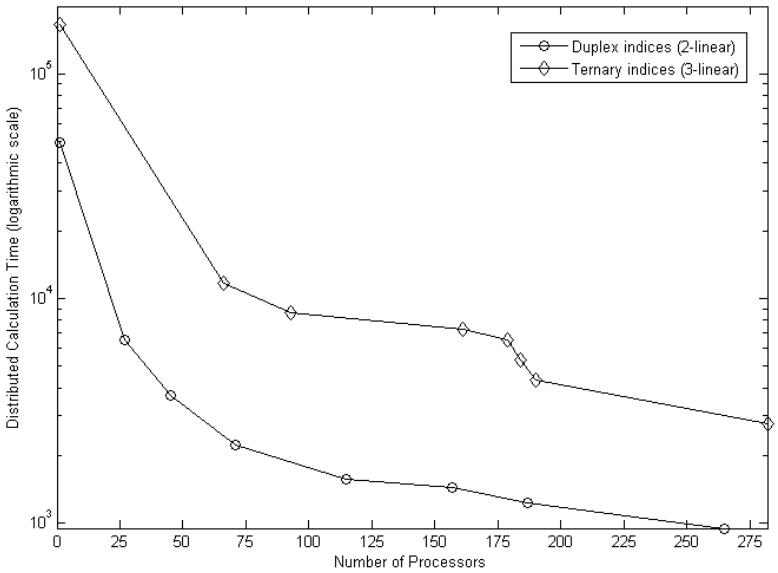

Рис. 2. Графическое представление времени обработки, достигнутое в процессе вычисления 2-линейных (Duplex) и 3-линейных (Ternary) алгебраических молекулярных дескрипторов

ЗАКЛЮЧЕНИЕ

В данной работе представлен новый модуль программного обеспечения QuBiLS-MIDAS для распределенного вычисления 3D *N*-линейных молекулярных дескрипторов. С этой целью была разработана распределенная

вычислительная платформа T-arenal с использованием мультисерверного подхода для высокопроизводительного решения вычислительных задач. Платформа T-arenal была развернута в динамической среде, состоящей из 337 неоднородных рабочих станций. Для оценки ее эффективности было выполнено вычисление 12 480 *2-линейных* и 7488 *3-линейных MD* на наборе данных, состоящем из 15 000 соединений. Полученные результаты демонстрируют, что последовательное время вычислений может быть сокращено в 52 раза в первом случае, и в 60 раз – во втором, если вычисление распределяется между различными рабочими станциями. Следовательно, можно сделать вывод, что новый модуль является полезным инструментом для высокоскоростного вычисления *3D N-линейных MD* на больших наборах данных.

СПИСОК ЛИТЕРАТУРЫ

1. *Brown F.K.* Chapter 35. Chemoinformatics: what is it and how does it impact drug discovery // Annual Reports in Medicinal Chemistry. – 1998. – Vol. 33. – P. 375–384. – doi: 10.1016/S0065-7743(08)61100-8.

2. *Todeschini R., Consonni V.* Molecular descriptors for chemoinformatics. Vol. 1: Alphabetical listing. – 1st ed. – Weinheim: Wiley-VCH, 2009. – 967 p.

3. DRAGON, v 6.0 / Milano chemometrics and QSAR research group. – Milano, Italy, 2010.

4. Mold(2), molecular descriptors from 2D structures for chemoinformatics and toxicoinformatics / H. Hong, Q. Xie, W. Ge, F. Qian, H. Fang, L. Shi, Z. Su, R. Perkins, W. Tong // Journal of Chemical Information and Modeling. – 2008. – Vol. 48, iss. 7. – P. 1337–1344. – doi: 10.1021/ci800038f.

5. Blue Desk. – Tübingen, Germany: University of Tübingen, 2008.

6. *Liu K., Feng J., Young S.S.* Power MV: A software environment for molecular viewing, descriptor generation, data analysis and hit evaluation // Journal of Chemical Information and Modeling. – 2005. – Vol. 45, iss. 2. – P. 515–522. – doi: 10.1021/ci049847v.

7. *Yap C.W.* PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints // Journal of Computational Chemistry. – 2011. – Vol. 32, iss. 7. – P. 1466–1474. – doi: 10.1002/jcc.21707.

8. QuBiLS-MIDAS: a parallel free-software for molecular descriptors computation based on multilinear algebraic maps / C.R. García-Jacas, Y. Marrero-Ponce, L. Acevedo-Martínez, S.J. Barigye, J.R. Valdés-Martiní, E. Contreras-Torres // Journal of Computational Chemistry. – 2014. – Vol. 35, iss. 18. – P. 1395–1409. – doi: 10.1002/jcc.23640.

9. N-linear algebraic maps for chemical structure codification: a suitable generalization for atom-pair approaches? / C.R. García-Jacas, Y. Marrero-Ponce, S.J. Barigye, J.R. Valdés-Martiní, O.M. Rivera-Borroto, J.O. Verbel // Current Drug Metabolism. – 2014. – Vol. 15, iss. 4. – P. 441–469. – doi: 10.2174/1389200215666140605124506.

10. *Anderson D.P.* BOINC: a system for public-resource computing and storage // Fifth IEEE/ACM International Workshop on Grid Computing, 8 November 2004: Proceedings. – Pittsburgh, USA, 2004. – P. 4–10. – doi: 10.1109/GRID.2004.14.

11. Labs of the World, unite!!! / W. Cirne, F. Brasileiro, N. Andrade, L. Costa, A. Andrade, R. Novaes, M. Mowbray / Journal of Grid Computing. – 2006. – Vol. 4, iss. 3. – P. 225–246. – doi: 10.1007/s10723-006-9040-x.

12. *Lam C.* Hadoop in action. – Greenwich, Connecticut, USA: Manning Publications, 2010. – 336 p.

13. Distributed Java platform with programmable MIMD capabilities / T. Keane, R. Allen, T.J. Naughton, J. McInerney, J. Waldron // Scientific Engineering for Distributed Java Applications: International Workshop, FIDJI 2002, Luxembourg, November 28–29, 2002. – Revised Papers. – Berlin; Heidelberg: Springer, 2003. – P. 122–131. – (Lecture Notes in Computer Science; vol. 2604). – doi: 10.1007/3-540-36520-6_11.

14. *Keane T.M., Naughton T.J.* DSEARCH: sensitive database searching using distributed computing // Bioinformatics. – 2005. – Vol. 21, iss. 8. – P. 1705–1706. – doi:10.1093/bioinformatics/bti163.

15. *Livny M., Melman M.* Load balancing in homogeneous broadcast distributed systems // ACM SIGMETRICS Performance Evaluation Review. – 1982. – Vol. 11, iss. 1. – P. 47–55. – doi: 10.1145/1010631.801689.

16. *Pitt E., McNiff K.* Java.rmi: The Remote method invocation guide. – Boston, Massachusetts, USA: Addison-Wesley Longman, 2001. – 320 p.

Гарсиа-Джакас Сезар, магистр информатики, профессор, математический вычислительный центр Университета информатики, Гавана, Куба. Основное направление научных исследований – распределенные вычисления. Имеет 15 публикаций. E-mail: crjacas@uci.cu

Авдеенко Татьяна Владимировна, доктор технических наук, профессор, заведующая кафедрой экономической информатики Новосибирского государственного технического университета. Основное направление научных исследований – математическое моделирование. Имеет 130 публикаций. E-mail: tavdeenko@mail.ru

A multi-server approach to high-throughput molecular descriptor calculation[*]

C.R. GARCÍA-JACAS[1], *T.V. AVDEENKO*[2]

[1] *Bioinformatics Research Group, Mathematical Computational Center, University of Informatics Sciences, Havana, Cuba, Master in Computer Sciences, professor. E-mail: crjacas@uci.cu*

[2] *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation D.Sc. (Eng.), professor. E-mail: tavdeenko@mail.ru*

The present paper introduces a new module of the QuBiLS-MIDAS software for the distributed computation of the 3D Multi-Linear algebraic molecular indices. The program QuBiLS-MIDAS has the following features: multi-core processing, a batch processing mode, a data cleaning module, MD custom configuration facilities, full cross-platform and others. The main reason for developing a new module was to cope with the computational complexity experienced during calculations of large dataset descriptors. To accomplish this task, a multi-server computing platform named T-arenal was developed for institutions with many workstations interconnected through a local network and without allocating resources specially designed for computational tasks. A graphical user interface forms a front-end component of the Y-aerial platform. This new system was deployed in 337 workstations belonging to the computing laboratories of the University of Informatics Sciences, Havana, Cuba, and it was perfectly integrated with the QuBiLS-MIDAS software. T-arenal was configured with a *root server* and three *request servers* with an equal number of *clients* (workstations). To illustrate the applicability of the T-arenal platform, performance tests on a dataset comprised of 15 000 compounds were carried out yielding a 52- and 60-fold reduction in the sequential processing time for the 2-Linear and 3-Linear indices, respectively. Thus, it can be stated that the T-arenal based distribution of computation tasks constitutes a suitable strategy for performing high-throughput calculations of 3D Multi-Linear descriptors.

Keywords: chemoinformatics, distributed computing system, multi-server architecture, platform of distributed tasks, molecular descriptors, parallel computing, graphical user interface, client-server architecture

DOI: 10.17212/1814-1196-2015-1-148-160

[*] *Received 1 December 2014.*

REFERENCES

1. Brown F.K. Chapter 35. Chemoinformatics: what is it and how does it impact drug discovery. *Annual Reports in Medicinal Chemistry*. 1998, vol. 33, pp. 375–384. doi: 10.1016/S0065-7743(08)61100-8

2. Todeschini R., Consonni V. *Molecular descriptors for chemoinformatics.* Vol. 1: *Alphabetical listing*. 1st ed. Weinheim, Wiley-VCH, 2009. 967 p.

3. *DRAGON, v 6.0*. Milano chemometrics and QSAR research group. Milano, Italy, 2010.

4. Hong H., Xie Q., Ge W., Qian F., Fang H., Shi L., Su Z., Perkins R., Tong W. Mold(2), molecular descriptors from 2d structures for chemoinformatics and toxicoinformatics. *Journal of Chemical Information and Modeling*, 2008, vol. 48, iss. 7, pp. 1337–1344. doi: 10.1021/ci800038f

5. *Blue Desk.* University of Tübingen, Tübingen, Germany, 2008.

6. Liu K., Feng J., Young S.S. Power MV: A software environment for molecular viewing, descriptor generation, data analysis and hit evaluation. *Journal of Chemical Information and Modeling*, 2005, vol. 45, iss. 2, pp. 515–522. doi: 10.1021/ci049847v

7. Yap C.W. PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of Computational Chemistry*, 2011, vol. 32, iss. 7, pp. 1466–1474. doi: 10.1002/jcc.21707

8. García-Jacas C.R., Marrero-Ponce Y., Acevedo-Martínez L., Barigye S.J., Valdés-Martiní J.R., Contreras-Torres E. QuBiLS-MIDAS: a parallel free-software for molecular descriptors computation based on multilinear algebraic maps. Journal of Computational Chemistry, 2014, vol. 35, iss. 18, pp. 1395–1409. doi: 10.1002/jcc.23640

9. García-Jacas C.R., Marrero-Ponce Y., Barigye S.J., Valdés-Martiní J.R., Rivera-Borroto O.M., Verbel J.O. N-linear algebraic maps for chemical structure codification: a suitable generalization for atom-pair approaches? *Current Drug Metabolism*, 2014, vol. 15, iss. 4, pp. 441–469. doi: 10.2174/1389200215666140605124506

10. Anderson D.P. BOINC: a system for public-resource computing and *Anderson D.P.* BOINC: a system for public-resource computing and storage. *Proceedings of Fifth IEEE/ACM International Workshop on Grid Computing*, Pittsburgh, USA, 8 November 2004, pp. 4–10. doi: 10.1109/GRID.2004.14

11. Cirne W., Brasileiro F., Andrade N., Costa L., Andrade A., Novaes R., Mowbray M. Labs of the World, unite!!! *Journal of Grid Computing*, 2006, vol. 4, iss. 3, pp. 225–246. doi: 10.1007/s10723-006-9040-x

12. Lam C. *Hadoop in Action.* Greenwich, Connecticut, USA, Manning Publications, 2010. 336 p.

13. Keane T., Allen R., Naughton T.J., McInerney J., Waldron J. Distributed Java platform with programmable MIMD capabilities. *Lecture Notes in Computer Science: Scientific Engineering for Distributed Java Applications*: International Workshop, FIDJI 2002, Luxembourg, November 28–29, 2002. Revised Papers. Berlin, Heidelberg, Springer, 2003, vol. 2604, pp. 122–131. doi: 10.1007/3-540-36520-6_11

14. Keane T.M., Naughton T.J. DSEARCH: sensitive database searching using distributed computing. *Bioinformatics* , 2005, vol. 21, iss. 8, pp. 1705–1706. doi:10.1093/bioinformatics/bti163

15. Livny M., Melman M. Load balancing in homogeneous broadcast distributed systems. *ACM SIGMETRICS Per-formance Evaluation Review*, 1982, vol. 11, iss. 1, pp. 47–55. doi: 10.1145/1010631.801689

16. Pitt E., McNiff K. *Java.rmi: The Remote method invocation guide*. Boston, Massachusetts, USA, Addison-Wesley Longman, 2001. 320 p.