

Частичная автоматизация расчетов на гибридных супер-ЭВМ при моделировании переноса заряда в квазиодномерных молекулярных цепочках

Пятков М.И., Зайцев А.Ю., Ольшевец М.М., Фиалко Н.С.

ИМПБ РАН – филиал ИПМ им. М.В. Келдыша РАН

fialka@gmail.com

Задача моделирования динамики избыточного заряда в квазиодномерной молекулярной цепочке сводится к расчету траектории системы, т.е. к численному интегрированию системы ОДУ, размерность которой зависит от длины цепочки. Учет температурных флуктуаций приводит к необходимости расчета множества траекторий с последующим осреднением. Точность средних зависит от числа реализаций K как \sqrt{K} . Для решения таких задач за разумные времена требуются гибридные супер-ЭВМ. При проведении расчетов на кластере в суперкомпьютерном центре коллективного пользования появляются технические проблемы, связанные с политикой администрирования кластера, которые требуют постоянного внимания исследователя. В работе описана частичная автоматизация расчетов на гибридных супер-ЭВМ. Показана эффективность разработанных процедур при моделировании переноса заряда в биополимерах в широком диапазоне температур термостата. Разработанный комплекс программ применяется для серийных расчетов при исследованиях переноса заряда в ДНК на гибридных вычислительных комплексах K-60 и K-100 в ИПМ им. М.В. Келдыша РАН.

Ключевые слова: автоматизация вычислений, воспроизводимые вычислительные эксперименты.

Implementation of partially automated calculations on hybrid supercomputers in the modelling of charge transfer in quasi-one-dimensional molecular chains

Ryatkov M.I., Zaytsev A.Y., Olshevets M.M., Fialko N.S.

IMPB RAS – Branch of KIAM RAS

The modeling of dynamics of excess charge in a quasi-one-dimensional molecular chain is reduced to calculating the trajectory of the system, i.e. to the numerical integration of ODE system with its dimension depending on the length of the chain. To take into account the temperature fluctuations, it is necessary to calculate the set of trajectories and then to average them over the ensemble. The accuracy of the mean values depends on the number of realizations K as \sqrt{K} . To solve such problems within a reasonable time, hybrid supercomputers are required. When using a supercomputer center for collective use, technical problems, associating with the cluster administration policy, arise and require the continual attention of the researcher. In this paper, we describe the partial automation of calculations for hybrid supercomputers. The effectiveness of the developed procedures is shown for modeling the charge transfer in biopolymers over a wide range of thermostat temperatures. The developed package is used on HPC clusters K-60 and K-100 in the Keldysh Institute of Applied Mathematics for numerical experiments on the charge transfer in DNA.

Key words: programmable automation, reproducible numerical experiments.

1. Введение

Важным инструментом в изучении процессов переноса энергии или заряда в молекулярных цепочках является построение моделей и исследование с помощью вычислительных экспериментов. Математическое моделирование процессов переноса заряда в биологических

системах связано с использованием дискретных моделей, в которых рассматриваются пути переноса возбуждения в макромолекулах [1–5].

Интерес к изучению возможных механизмов переноса заряда вдоль биомacroмолекулы ДНК связан с возможностью применения ДНК в нанобиоэлектронике [6–8]. Особое внимание уделяется вопросу о подвижности носителей тока в ДНК в связи с перспективой использования ДНК-

проводов в нанoeлектронике. Знание термодинамических характеристик, таких как полная энергия, электронная часть теплоемкости, параметр делокализации и т.п., также существенно для понимания фундаментальных свойств таких систем.

2. О постановке вычислительной задачи

Моделирование динамики переноса заряда по цепочке сайтов (для ДНК сайт – это комплементарная пара оснований) основано на гамильтониане Холстейна в полуклассическом приближении. Система уравнений движения в безразмерной форме имеет вид:

$$i\dot{b}_n = \eta_{n,n-1}b_{n-1} + \eta_{n,n}b_n + \eta_{n,n+1}b_{n+1} + \chi u_n b_n, \quad (1)$$

$$\ddot{u}_n = -\omega^2 u_n - \chi |b_n|^2 + \gamma \dot{u}_n + \xi Z_n(t). \quad (2)$$

Здесь $n = 1, \dots, N$, N – число сайтов в цепочке. Переменные: b_n – комплексные (амплитуда вероятности нахождения заряда на n -ом сайте цепочки) и действительные u_n (классические смещения сайта из невозмущенного положения). Значения параметров η_{nk} , ω , χ зависят от типа цепочки и выбираются в соответствии с постановкой задачи, например – по экспериментальным данным об энергетических и динамических характеристиках отдельных нуклеотидов. Температура термостата T имитируется добавкой в (2) члена с трением γ и белого шума $Z_n(t)$ с коэффициентом, $\xi^2 = 2k_B T \gamma$. Подробнее о математической модели и о величинах параметров, характерных для фрагментов ДНК, см., например, [7, 9–12].

Соответствующая задаче (1),(2) система ОДУ первого порядка со случайной правой частью состоит из $4N$ действительных уравнений. Вычислительный эксперимент: при заданной температуре T проводится расчет множества реализаций (траекторий системы (1), (2) из разных начальных данных и с разными временными «случайными» последовательностями), и считаются средние по реализациям временные зависимости. Результатом являются усредненные по реализациям траектории движения, временные зависимости полной энергии, вероятности распределения заряда по сайтам цепочки и т.п.

Параллельная версия программы написана на языке C с использованием MPI, применяется естественное распараллеливание «один процесс – одна реализация», управляющая машина опрашивает рабочие узлы, собирает результаты, рассчитывает зависящие от переменных функции, усредняет и записывает результаты в файлы.

Точность среднего значения по выборке улучшается как квадратный корень из количества реализаций. (Для оценки средних пока нам достаточно 100 реализаций). Время расчета до прихода системы к термодинамическому равновесию может быть очень велико, например, из

начального поляронного состояния при малых температурах термостата, или в модели «донор–мостик–акцептор», когда в начальный момент заряд локализован на сайте донора с глубокой энергетической ямой. Для выполнения таких расчетов может потребоваться значительный период (несколько дней) машинного времени на каждую реализацию. Такие большие машинные ресурсы можно получить на кластерных суперкомпьютерах, например, в ИПМ им. М.В. Келдыша РАН на вычислительных комплексах К-100 или К-60.

В случае расчетов на суперкомпьютере общее время выполнения задачи зависит от политики администрирования кластера. Если кластер загружен, то место задачи в очереди определяется шкалой приоритетов, учитывающей запрошенное количество процессоров, время исполнения и другие параметры.

3. Автоматизация вычислительных экспериментов

В системах пакетной обработки данных очередь заданий спроектирована таким образом, что задачи с наименьшим временем счета выполняются как можно быстрее, в том числе по причине того, что они могут быть просто отладочными. Это позволяет оптимально распределить ресурсы операционной системы, чтобы исключить простои в работе. Тем не менее, вводятся естественные ограничения для конечного пользователя, которые не дают занять «короткими» задачами всю очередь заданий. Обычно число одновременно запускаемых задач ограничивается 5–7, что, в случае, когда требуется провести множество независимых, но однотипных вычислительных экспериментов, требует от пользователя дополнительных усилий для отслеживания и запуска новых задач.

Время исполнения задачи и запрашиваемое количество процессоров наиболее важные параметры, влияющие на приоритет задачи в очереди. Поскольку приоритет – совокупная характеристика, то чем меньше мы требуем либо одного ресурса, либо другого, тем быстрее мы получаем результат. В рамках данной работы было выявлено, что декомпозиция по количеству запрашиваемых процессоров (1 процессор = 1 реализация) приводила к тому, что совокупное количество задач выполнялось быстрее по отношению к одной большой пакетной задаче. То есть, например, вместо одной большой задачи на 96 процессоров, мы проводили расчет 8 задач на 12 процессорах. Такая схема весьма эффективна лишь в том случае, если очередь всегда заполнена большими пакетными заданиями и есть небольшое количество неиспользуемых узлов. Если очередь пуста, то всегда можно провести декомпозицию таким образом, что каждая задача будет занимать сразу несколько узлов.

Выполнение такого «дробления по реализациям»

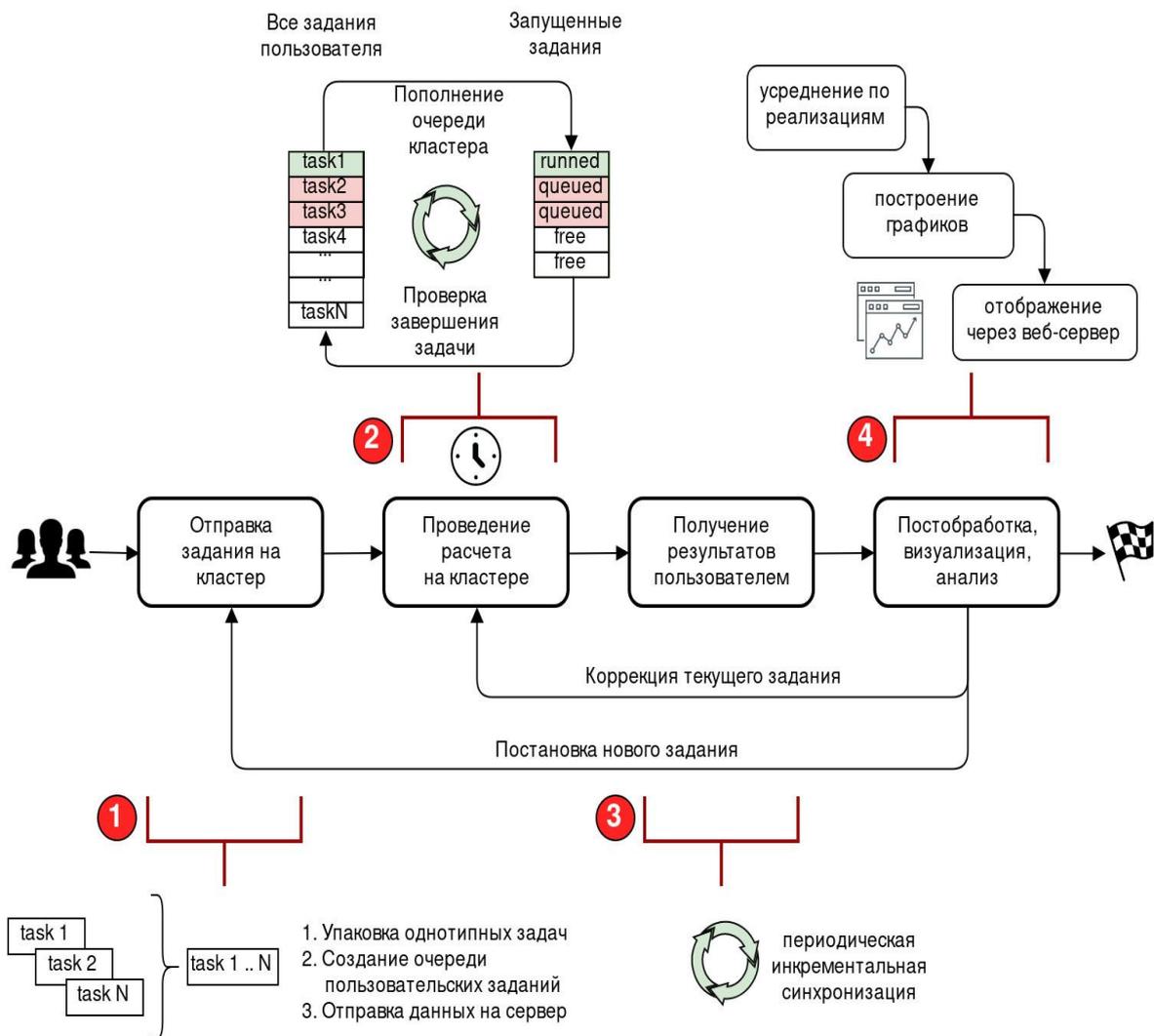


Рис. 1. Общая схема работы пользователя при расчете на кластере. Маркером помечены пункты, которые были автоматизированы. (1) – для управления однотипными заданиями, которые отличаются только параметрами, оформляется конфигурационный файл, в котором параметры описываются диапазонами. По данному конфигурационному файлу создается требуемое дерево директорий, каждая из которых отвечает за отдельное задание. Также создается файл очереди, который пересылается на вычислительный кластер. (2) – наиболее длительный этап, на котором требуется отслеживать выполнение задач и проверять условия их завершения. Очередь заданий сервера пополняется автоматически до тех пор, пока не будут выполнены все задания. Условия завершения задачи формулируются на предыдущем этапе. (3) – синхронизация результатов проводится периодически, вне зависимости от статуса задачи. То есть, если задача выполнена не полностью, то результаты все равно пересылаются на пользовательский компьютер. Инкрементальность в данном случае означает, что при следующей синхронизации будут выкачиваться только изменившиеся файлы. (4) – над промежуточными результатами проводится препроцессинг, в данном случае, поскольку производится декомпозиция по количеству реализации (или процессоров), то данные усредняются, визуализируются с помощью библиотеку matplotlib и автоматически становятся доступны онлайн через веб-интерфейс.

в совокупности с ограничениями, которые на нас накладывает очередь заданий, приводит нас к тому, что требуется более высокий уровень автоматизации. В ручном режиме такие вещи как перезапуск с контрольных точек, расчет средних из усредненных по малым выборкам, проверка выхода на равновесное состояние и т.п. превращаются в весьма трудоемкую и монотонную задачу.

В рамках данной работы было разработано несколько вспомогательных программ, которые позволили существенно ускорить получение

результатов. Во-первых, дополнительно к уже имеющейся очереди заданий сервера, была реализована дополнительная очередь, в которую помещались абсолютно все задания пользователя. Обслуживающая программа на сервере периодически проводила опрос данной локальной очереди на наличие заданий и очереди заданий сервера на предмет возможности произвести запуск задачи. Интервал периодичности был подобран эмпирически, поскольку задачи по времени выполнения могли существенно отличаться друг от

друга. Во-вторых, было принято решение не делать на сервере никаких дополнительных действий, таких как архивирование полученных результатов, усреднение по реализациям и т.д. Для этого была реализована периодическая инкрементальная синхронизация получаемых данных между кластером и клиентским компьютером штатными средствами Linux (rsync).

На стороне клиента происходит постобработка результатов, усреднение в зависимости от количества начальных условий и заказанных процессоров, расчет отклонения равномерной траектории от полярона, создание архивных копий. Для полученных данных автоматически визуализируются и отображаются онлайн. Подробная схема отображена на рисунке 1.

В настоящий момент широко распространены фреймворки планирования заданий, позволяющие автоматизировать вычисления, проводимые на кластерах [13–16]. Одним из важных свойств, которые ими предлагаются, является репродуцируемость вычислений.

В нашем случае мы также применили подобные техники для удобства повторения расчетов. Поскольку все задания были однотипными, но выполнялись с различными набором параметров, таких как температура, количество начальных условий, предполагаемое время счета, количество процессоров, выделяемых для одного задания и т.д., был разработан конфигурационный файл с декларативным описанием переменных. Данный конфигурационный файл являлся входным для программы, создающей образ задачи, отправляемой на сервер. Таким образом, набор входных данных и конфигурационный файл полностью определяют пакет заданий.

3. Заключительные замечания

С использованием описанной автоматизации была исследована задача расчета термодинамически равновесных состояний в однородной цепочке с дефектом [17] в зависимости от температуры и параметров сайта-ловушки. На рисунке 2 приведен пример одного расчета: задан перебор по температуре T (остальные параметры фиксированы), для каждой заданной T два набора начальных данных подсистемы (1), соответствующих наибольшей и наименьшей энергии системы без случайной силы, для каждого набора расчет 50 реализаций.

В ручном режиме ситуация (рис. 2) соответствует запуску и периодическому отслеживанию 14 параллельных задач, что довольно затратно по времени. Минус подобной автоматизации довольно очевиден: возможен «избыточный» расчет траектории после выхода к равновесному состоянию (например, на рис. 2 графики для температур $T = 20$ и $T = 25$). Это вопрос дальнейшей доработки.

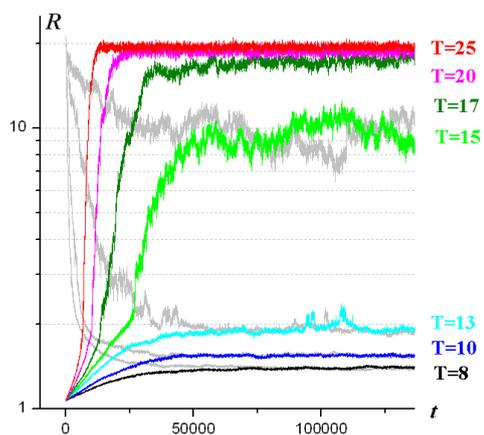


Рис. 2. Один расчет с использованием автоматизации. Серым цветом – результаты расчетов из начальных состояний с наибольшей энергией при $T = 0$, цветные – из начальных состояний с минимальной энергией при $T = 0$. Среднее значение функции в термодинамически равновесном состоянии можно получить дополнительным осреднением по времени после момента, когда кривые станут близки.

Были проведены простые тесты для определения эффективности в зависимости от числа запрашиваемых узлов. При постановке задачи в очередь заказывалось машинное время 30 минут и разное число узлов, и сравнивалась длина средних траекторий (другими словами, тест на скорость обмена данными между узлами кластера). Результаты для K-100: при заказе узлов 1:2:4:8 отношение длин траекторий 1:0.97:0.91:0.43. В таких же тестах для K-60 отношение длин траекторий всегда примерно единица, однако сама траектория в два раза короче, чем полученная на одном узле K-100.

Для рассматриваемой задачи выгодно проводить декомпозицию, например, вместо одной большой задачи на 96 процессоров выполнять расчет 8 задач на 12 процессорах.

Подобная автоматизация может быть полезной и в других задачах с похожими свойствами, где есть распараллеливание по реализациям без большого обмена данными, и требуется длительный расчет.

4. Благодарности

Работа выполнена при частичной поддержке Российского фонда фундаментальных исследований, проекты 16-07-00305-а и 17-07-00801-а.

5. Список литературы

1. Holstein T. Studies of polaron motion: Part I. The molecular-crystal model. *Ann. Phys.* 1959. V. 8. № 3. P. 325–342. doi: [10.1016/0003-4916\(59\)-90002-8](https://doi.org/10.1016/0003-4916(59)-90002-8).
2. Давыдов А.С. Солитоны в квазиодномерных молекулярных структурах. *УФН*. 1982. Т. 138.

- № 4. С. 603–643. doi: [10.3367/UFNr.0138.198212c.0603](https://doi.org/10.3367/UFNr.0138.198212c.0603).
3. Якушевич Л.В. *Методы теоретической физики в исследованиях свойств биополимеров*. Пушчино: ОНТИ НЦБИ АН СССР, 1990. 125 с.
 4. Scott A. Davydov's soliton. *Phys. Rep.* 1992. V. 217. № 1. P. 1–67. doi: [10.1016/0370-1573\(92\)90093-F](https://doi.org/10.1016/0370-1573(92)90093-F).
 5. *Modern Methods for Theoretical Physical Chemistry of Biopolymers*. Eds. Starikov E.B., Tanaka S., Lewis J.P. Amsterdam: Elsevier Scientific, 2006. 461 p.
 6. Ventra M., Zwolak M. DNA electronics. In: *Encyclopedia of Nanoscience and Nanotechnology*. Eds. Nalwa H.S., Stevenson R. CA: American Scientific Publishers, 2004. V. 2. P. 475–493.
 7. Lakhno V.D. DNA nanobioelectronics. *Int. J. Quant. Chem.* 2008. V. 108. № 11. P. 1970–1981. doi: [10.1002/qua.21717](https://doi.org/10.1002/qua.21717).
 8. *Nanobioelectronics – for Electronics, Biology, and Medicine*. Eds. Offenhausser A., Rinaldi R. New York: Springer, 2009. 337 p.
 9. Фиалко Н.С., Соболев Е.В., Лахно В.Д. О расчетах термодинамических величин в модели Холстейна для однородных полинуклеотидов. *ЖЭТФ*. 2017. Т. 151. № 4. С. 744–751. doi: [10.7868/S0044451017040000](https://doi.org/10.7868/S0044451017040000).
 10. Starikov E. Electron-phonon coupling in DNA: a systematic study. *Philos. Mag.* 2005. V. 85. № 29. P. 3435–3462. doi: [10.1080/14786430500157110](https://doi.org/10.1080/14786430500157110).
 11. Lewis F.D., Wu Ya. Dynamics of superexchange photoinduced electron transfer in duplex DNA. *J. Photochem. Photobiol. C*. 2001. V. 2. № 1. P. 1–16. doi: [10.1016/S1389-5567\(01\)00008-9](https://doi.org/10.1016/S1389-5567(01)00008-9).
 12. Lakhno V.D., Sultanov V.B., Pettitt B.M. Combined hopping-superexchange model of a hole transfer in DNA. *Chem. Phys. Lett.* 2004. V. 400. P. 47–53. doi: [10.1016/j.cplett.2004.10.077](https://doi.org/10.1016/j.cplett.2004.10.077).
 13. Monat C., Tranchant-Dubreuil C., Kougbeadjo A., Farcy C., Ortega-Abboud E., Amanzougarene S., Orjuela-Bouniol J., Summo M., Sabot F. TOGGLE: toolbox for generic NGS analyses. *BMC Bioinformatics*. 2015. V. 16. P. 374. doi: [10.1186/s12859-015-0795-6](https://doi.org/10.1186/s12859-015-0795-6).
 14. Pulvermacher C., Keck T., Feindt M., Heck M., Kuhr T. An automated framework for hierarchical reconstruction of B mesons at the Belle II experiment. *J. Phys.: Conf. Ser.* 2015. V. 608 P. 012048. doi: [10.1088/1742-6596/608/1/012048](https://doi.org/10.1088/1742-6596/608/1/012048).
 15. Schorderet P. NEAT: a framework for building fully automated NGS pipelines and analyses. *BMC Bioinformatics*. 2016 V. 17 № 53. P. 1–9. doi: [10.1186/s12859-016-0902-3](https://doi.org/10.1186/s12859-016-0902-3).
 16. Ko G., Kim P.G, Yoon J., Han G., Park S.J., Song W., Lee B. Closha: bioinformatics workflow system for the analysis of massive sequencing data. *BMC Bioinformatics*. 2018. V. 19. № 19. P. 43. doi: [10.1186/s12859-018-2019-3](https://doi.org/10.1186/s12859-018-2019-3).
 17. Fialko N., Pyatkov M., Lakhno V. On the Thermodynamic Equilibrium Distribution of a Charge in a Homogeneous Chain with a Defect. In: *MMCP 2017*. Eds. Adam G., Busa J., Hnatic M., Podgany D. *EPJ Web of Conferences*. 2018. V. 173. Article No. 06004. P. 4. doi: [10.1051/epjconf/201817306004](https://doi.org/10.1051/epjconf/201817306004).