

**ВЫЧИСЛИТЕЛЬНАЯ СРЕДА ДЛЯ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ
НАНОСИСТЕМ. CASE-УПРАВЛЕНИЕ ВЫЧИСЛЕНИЯМИ**

**COMPUTATIONAL ENVIRONMENT FOR NANOSYSTEM MODELING.
CASE-BASED TASK MANAGEMENT SYSTEM**

К. М. Назаренко, Е. С. Назаренко, П. Н. Марков, Н. А. Коробов, А. Б. Надыкто

ФГБОУ ВПО МГТУ «СТАНКИН»

127055, Москва, Вадковский пер., 1, Российская Федерация

тел:(+7 499)972-55-00;

e-mail: cmr.nazy@gmail.com

Описаны принципы и средства автоматизации написания управляющих программ, основанные на встроенных утилитах GNU/Linux и CASE. Разработанная CASE-система предназначена для организации масштабных вычислений силами одной группы или подразделения без участия высокооплачиваемого IT персонала, специализирующегося в области стандартных программных решений, и операторов вычислений. Данная система адаптирована к специфике молекулярных расчетов, проста и удобна в развертывании и эксплуатации, обладает высокой отказоустойчивостью, гибкостью, скоростью реконfigurирования и низкой стоимостью владения.

Ключевые слова: математическое моделирование, наносистемы, управление заданиями, CASE, высокопроизводительные вычисления, GRID-системы, HPC, суперкомпьютеры, вычислительная среда, *ab initio*, DFT.

This paper describes the automated task management system for molecular and quantum-chemical calculations, which is based on embedded GNU/Linux utilities and CASE (Computer-Aided Software Engineering) technology. The developed CASE-based system is capable of managing large-scale computations carried out by a small group including neither highly-qualified software engineers nor operators. The present system, which was developed specifically for molecular and quantum-chemical calculations, is simple and easy to deploy and to operate, and offers high performance, flexibility and configuration speed at low costs of ownership.

Keywords: mathematical modeling, nanosystems, task management, CASE, high-performance computations, GRID-systems, HPC, supercomputers, computational environment, ab initio, DFT

Математическое моделирование наносистем имеет особенности, которые необходимо учитывать при организации масштабных вычислений. В частности, оно включает одновременное исполнение, зачастую различными программными пакетами, нескольких сотен или тысяч расчетных заданий, оперативную подготовку входных данных для запуска новых и перезапуска неудачно завершившихся файлов, их распределение по ЭВМ вычислительного комплекса, запуск прикладного ПО, мониторинг вычислений и обработку результатов [1].

На рынке представлено большое количество разнообразных систем управления заданиями, таких как gLite, Ansible Tower, Condor, программный каркас Hadoop и его производные и многие другие [2]. Они вполне удовлетворительно справляются со своей основной задачей: управлением заданиями. Однако организация вычислений в области свойств наносистем выходит далеко за рамки управления заданиями/автоматизации вычислений, предлагаемых готовыми решениями, и требует создания полноценной вычислительной среды, в полной мере учитывающей специфику выполняемых задач и используемых программных пакетов/прикладного ПО. К сожалению, готовые программные продукты не адаптированы к специфике квантово-химических вычислений, и поэтому их применение требует разработки дополнительной адаптирующей программной оболочки. Поскольку разработка такой оболочки и ее периодический апгрейд являются самостоятельными, достаточно сложными и ресурсоемкими задачами, требующими привлечения на постоянной основе высокооплачиваемого IT персонала, использование «готовых» платформ/систем управления заданиями целесообразно только в случае выполнения научных проектов общенационального уровня [3], аккумулирующих интеллектуальные и финансовые ресурсы, заведомо превышающие те, которыми обладает небольшая группа, подразделение или отдельный научный центр, для которого высокопроизводительные квантово-химические расчеты являются всего лишь одним из нескольких направлений исследований.

Необходимо заметить, что поскольку востребованность квантово-химических и молекулярных расчетов чрезвычайно велика в связи с тем, что именно на них основано атомно-молекулярное конструирование, являющееся основой большинства современных методов исследования нанообъектов и технологий разработки новых

материалов с уникальными свойствами [4-5], а количество вузов и научных центров обладающих парком в несколько десятков и более компьютеров постоянно растет, создание простой, удобной в использовании, надежной и недорогой в эксплуатации вычислительной среды для математического моделирования наносистем является весьма актуальной задачей.

Таким образом, целью данной работы является создание проблемно-ориентированной вычислительной среды для математического моделирования наносистем, обладающей определенным набором функций, свойств и эксплуатационных характеристик:

- глубокой степенью автоматизации управления заданиями;
- системой оперативного мониторинга вычислений;
- комплексом программ для подготовки исходных данных, обработки результатов вычислений и составления сводных отчетов по массивам выполненных заданий;
- простым, эффективным и удобным для пользователя интерфейсом;
- высокой отказоустойчивостью, интегральной надежностью и масштабируемостью;
- простотой развертывания, диагностики и аварийного восстановления;
- низкой стоимостью владения.

Разработанная система управления заданиями (см. рис.1) проста, эффективна и достаточно оригинальна. Она основана на иерархии вычислительных узлов и системе управляющих CASE-скриптов [6]. При ее разработке был сделан выбор в пользу встроенных инструментов автоматической обработки текстовых файлов - утилит текстового процессинга GNU операционных систем семейства UNIX, таких как `awk`, `sed`, `grep`, вместо часто используемых готовых программных решений в связи с тем, что большинство пакетов, применяемых для моделирования наносистем, используют текстовые форматы данных. Отказ от использования аппарата баз данных (БД) обусловлен сложностью их внедрения, а также невысокой интегральной надежностью, недостаточной гибкостью и сложностью эксплуатации в условиях неоднородной, динамически меняющейся структуры информационных потоков при проведении квантово-химических и молекулярных расчетов. Иерархия взаимодействующих вычислительных узлов построена при помощи встроенных протоколов SSH, RSH и SMB/CIFS. Для обработки больших наборов вычислительных заданий используется инструментарий автоматизации операций с файлами и текстовыми строками командной оболочки UNIX, файловые маски и язык регулярных выражений.

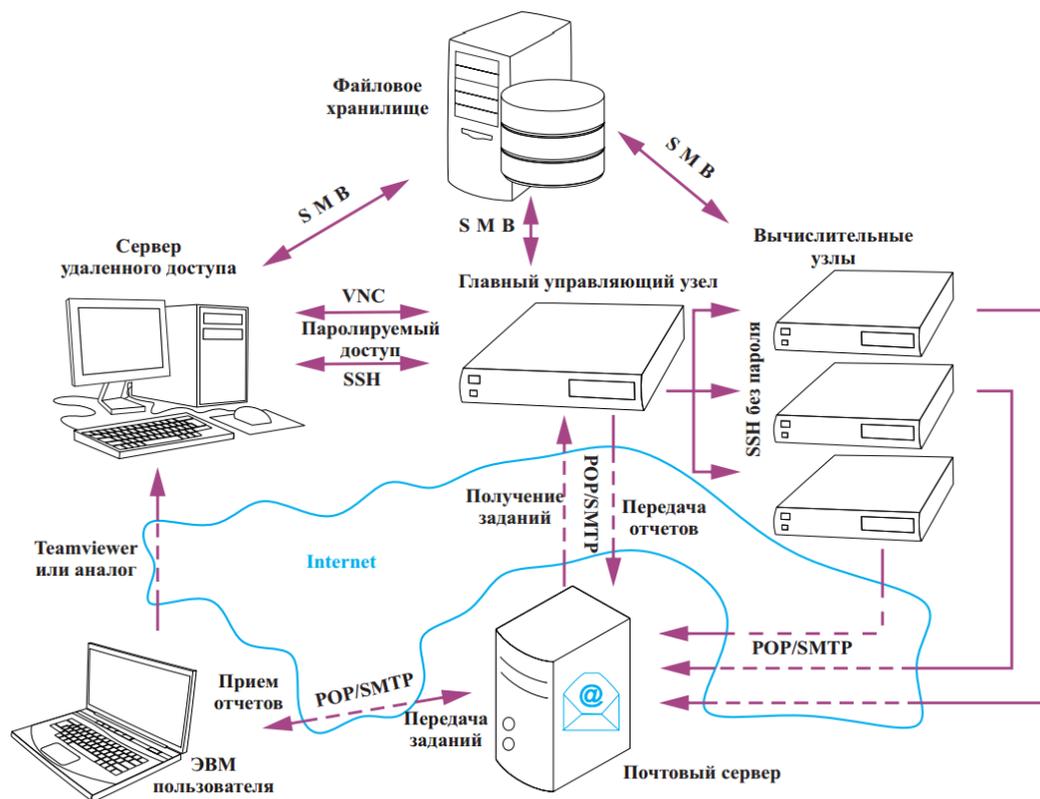


Рис. 1. Общая структура вычислительной среды.

Мониторинг состояния вычислительного комплекса и вычислительных процессов осуществляется при помощи встроенной системы работы с электронной почтой ОС семейства Unix. Удаленное управление вычислениями и администрирование производится через Интернет при помощи программы удаленного доступа TeamViewer с любого мобильного устройства в режиме реального времени.

Как видно из рис. 1, в структуре системы управления заданиями выделены управляющие узлы, передающие команды на выполнение другим (ведомым) узлам с помощью автоматической авторизации по SSH без пароля. Управляющий узел может одновременно быть и ведомым, как для самого себя (исполняя свои собственные команды через повторную авторизацию), так и для другого управляющего узла, в подчинении которого он находится. Такая схема позволяет создать сколь угодно сложную разветвленную иерархическую систему взаимодействующих узлов путем их объединения во взаимодействующие группы и тем самым обеспечивает повышенную отказоустойчивость системы с помощью передачи функций управляющего узла другому узлу при выходе управляющего узла из строя или его сетевой недоступности.

При такой схеме организации сетевого взаимодействия производится передача большого числа однотипных команд с главного управляющего узла на ведомые узлы. Данная компонента системы управления заданиями разработана с применением технологии автоматического генерирования программного кода (CASE), обеспечивающей составление сценариев, осуществляющих ввод удаленных команд. Составление управляющих сценариев также производится автоматически при помощи специализированных CASE-скриптов. Эти сценарии осуществляют составление других сценариев, которые, после вложения в них информирующих инструкций, передаются вычислительным узлам на выполнение. Их функцией является составление всех текстов программ, реализующих выполнение заданий конкретного проекта. Входной информацией для данных скриптов являются файлы с заданиями, список IP-адресов вычислительных узлов и адреса электронной почты, через которые производится информирование пользователей о начале, ходе и завершении заданий. CASE-скрипты также осуществляют автоматическое генерирование программного кода, реализующего алгоритмы изменения конфигурации вычислительной среды, управления информационными потоками и вычислительными процессами.

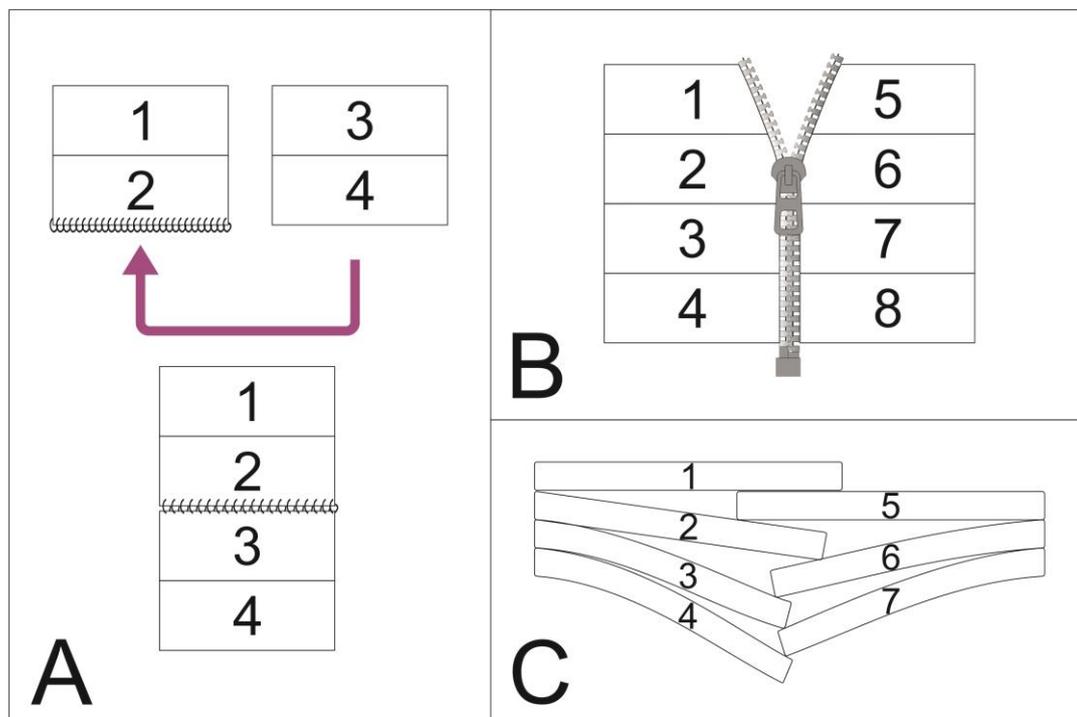


Рис. 2. Операции автоматического конструирования программного кода (вертикальная конкатенация (А), горизонтальное стыкование (В) , сэндвич-сборка (С)).

Исходными компонентами при конструировании управляющих сценариев являются шаблоны – типовые сегменты программного кода. Составление кода из этих шаблонов производится при помощи трех операций: вертикальной конкатенации, горизонтального стыкования и сэндвич-сборки, которые позволяют создавать сценарии практически любых требуемых конфигураций (см. рис. 2).

На рис. 3 показаны потоки информационных сообщений при распределении заданий на вычислительные узлы (а) и синхронном запуске на расчет (б). Управляющим вычислительным узлом составляется сценарий для анализа занятости ведомых вычислительных узлов, по результату выполнения которого создается сценарий, содержащий приказы о получении заранее подготовленных вычислительных заданий с файлового хранилища. После этого производится составление сценария-расписания запуска заданий с учетом требований к режиму их запуска (число используемых процессорных ядер, максимальная длина очереди, число параллельно выполняемых задач) и его выполнение. При начале обработки задания ведомый вычислительный узел отправляет соответствующее сообщение пользователю с помощью электронной почты. При сборе результатов производится составление сценария, содержащего приказы, управляющим узлом произвести их архивацию и передачу на файловое хранилище.

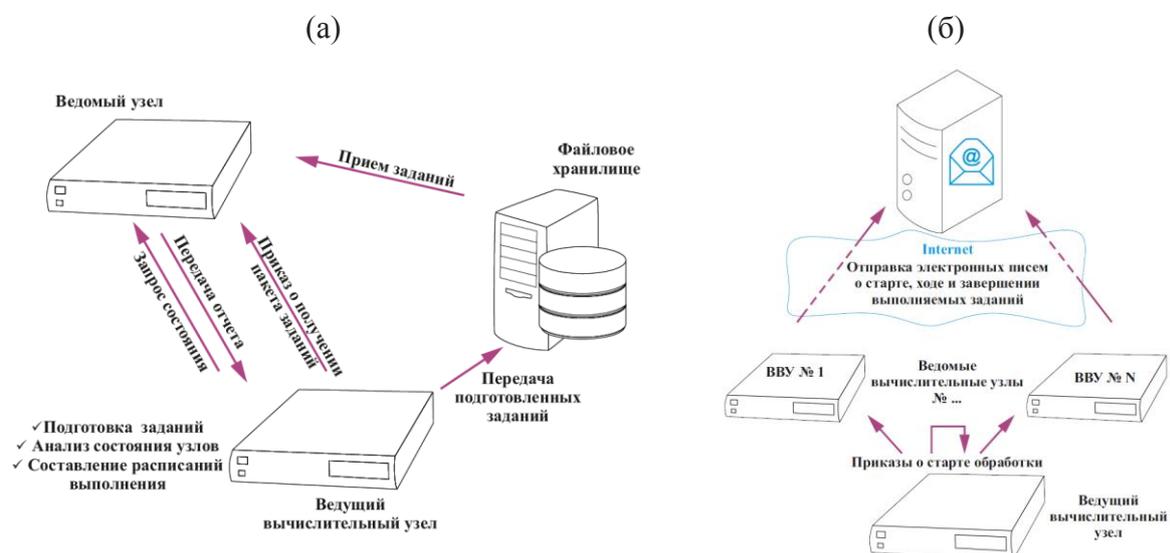


Рис. 3. Процесс размещения (а) и запуска (б) заданий на ведомых вычислительных узлах.

Реализация предложенной в данной работе схемы "Иерархия взаимодействующих узлов + система управляющих CASE-скриптов" позволила создать систему управления

заданиями, обладающую, помимо глубокой автоматизации вычислительных процессов, целым рядом достоинств, которые делают целесообразным ее применение для организации масштабных исследований наносистем силами подразделений, располагающих ограниченным бюджетом и человеческим ресурсом:

Низкая стоимость владения. Разработанная система управления заданиями позволяет осуществлять вычисления специалистом в предметной области самостоятельно. Отсутствует необходимость привлечения высокооплачиваемых специалистов по внедрению и сопровождению «готовых» решений и операторов вычислений, что позволяет значительно снизить стоимость вычислений.

Эргономичность и простота. Система управления проста и лаконична, а ее освоение не представляет никакого труда. Управление вычислениями производится непосредственно из командной оболочки ОС семейства UNIX. Для освоения принципов ее работы достаточно обладать базовыми навыками работы с командной строкой. Например, запуск заданий на расчет требует выполнения трех команд, сбор результатов и их сепарация – еще трех.

Гибкость и оперативность конфигурирования под конкретные задания. Написание новых управляющих сценариев производится на базе ранее разработанного программного кода, который отличается простотой и наглядностью, и не требует специальных знаний, выходящих за рамки базовых навыков программирования под UNIX/Linux. Библиотека скриптов содержит большое число разнообразных примеров, иллюстрирующих различные приемы и методы решения задач управления.

Простота и высокая скорость развертывания и модернизации. Для добавления нового вычислительного узла достаточно однократно внести его IP-адрес в общий список, запустить соответствующий скрипт и произвести однократную авторизацию. Процесс подключения нового узла с учетом временных затрат на установку системного и прикладного ПО занимает не более 40 минут, из которых трудозатраты персонала не превышают 10 минут. Последовательность действий, необходимых для установки нового вычислительного узла, формализована в виде набора инструкций, которые могут быть выполнены обычным пользователем.

Высокая отказоустойчивость. Восстановление работоспособности узла после аварийного отключения электроснабжения не подразумевает никаких дополнительных действий помимо выполнения процедур по автоматизированному копированию с него данных и возобновлению на нем расчетов, при помощи запуска специализированного набора скриптов.

Выводы:

(1) Система управления заданиями, разработанная на основе иерархии вычислительных узлов и технологии CASE, отвечает всем требованиям проектного задания. Таким образом, цель данной работы в части управления вычислениями достигнута.

(2) Разработанная система управления заданиями успешно эксплуатируется с 12/2014 на высокопроизводительной GRID [7]-системе, состоящей из 45 вычислительных узлов на базе 8-ядерных процессоров Intel Xeon, и лучше всего адаптирована к использованию в GRID-системах, являющихся наиболее доступными высокопроизводительными комплексами, чья производительность может быть сравнима с производительностью более дорогостоящих HPC и суперкомпьютеров в связи с низкой степенью параллелизации вычислительных алгоритмов в наиболее часто используемых программных пакетах для квантово-химического (ab initio, DFT, composite methods) и, в целом ряде случаев, молекулярного моделирования наносистем.

(3) Принципы, использованные при разработке данной системы, и сама разработанная система, достаточно универсальны по отношению к типу вычислительных комплексов, и могут успешно применяться на HPC и суперкомпьютерах.

(4) Разработанная система управления заданиями также обладает определенной универсальностью по отношению к типу вычислений и может применяться для решения других прикладных ресурсоемких задач, таких как трехмерный рендеринг, датамайнинг, криптоанализ и др.

Данная работа была проведена при поддержке Министерства Образования и Науки РФ в рамках Государственного Задания, проект номер 108 (рук. д.ф.-м.н. А.Б. Надыкто).

Литература:

[1] Titov, A. V., Ufimtsev, I. S., Luehr, N., Martinez, T. J. (2012). Generating efficient quantum chemistry codes for novel architectures. *Journal of Chemical Theory and Computation*, 9(1), 213-221.

[2] Wu, Y., Guo, W., Ren, J., Zhao, X., Zheng, W. (2014). NO₂: Speeding up Parallel Processing of Massive Compute-Intensive Tasks. *Computers, IEEE Transactions on*, 63(10), 2487-2499.

[3] Волохов, В. М., Варламов, Д. А., Волохов, А. В., Пивушков, А. В., Покатович, Г. А., Сурков, Н. Ф. (2011). Грид-сервисы в вычислительной химии: достижения и

перспективы. Вестник уфимского государственного авиационного технического университета, 15(5 (45)).

[4] Ohno, K., Esfarjani, K., Kawazoe, Y. (2012). Computational materials science: from ab initio to Monte Carlo methods (Vol. 129). Springer Science & Business Media.

[5] Britnell, L., Ribeiro, R. M., Eckmann, A., Jalil, R., Belle, B. D., Mishchenko, A., ..., Novoselov, K. S. (2013). Strong light-matter interactions in heterostructures of atomically thin films. *Science*, 340(6138), 1311-1314.

[6] Muller, H. A., Norman, R. J., Slonim, J. (Eds.). (2012). Computer Aided Software Engineering. Springer Science & Business Media.

[7] Nasmachnow, S. (2013). Parallel multiobjective evolutionary algorithms for batch scheduling in heterogeneous computing and grid systems. *Computational Optimization and Applications*, 55(2), 515-544.