

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

СОГЛАСОВАНО
Генеральный директор
ЗАО «АйТи»
Бакиев О.Р.
2011 г.

УТВЕРЖДАЮ
Ректор НИУ ИТМО
Васильев В.Н.
2011 г.

МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-
ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ
И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ
ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE

ПРИКЛАДНОЙ СЕРВИС МОДЕЛИРОВАНИЯ МОЛЕКУЛЯРНОЙ ДИНАМИКИ С
ИСПОЛЬЗОВАНИЕМ ПАРАМЕТРОВ, ОПРЕДЕЛЯЕМЫХ КВАНТОВО-
ХИМИЧЕСКИМИ МЕТОДАМИ

ОПИСАНИЕ ПРОГРАММЫ

ЛИСТ УТВЕРЖДЕНИЯ

RU.СНАБ.80066-06 13 54-ЛУ

Инв.№ подл.	Подп. и дата	Взам.инв.№	Инв.№ дубл.	Подп. и дата

Организации-разработчика

Руководитель разработки,
профессор НИУ ИТМО

Бухановский А.В.
"22" сентября 2011 г.

Ответственный исполнитель,
с.н.с. НИУ ИТМО

Луценко А.Е.
"22" сентября 2011 г.

Нормоконтролер
ведущий инженер НИУ ИТМО

Позднякова Л.Г.
"22" сентября 2011 г.

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

УТВЕРЖДЕН
RU.СНАБ.80066-06 13 54-ЛУ

**МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-
ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ
И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ
ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE**

**ПРИКЛАДНОЙ СЕРВИС МОДЕЛИРОВАНИЯ МОЛЕКУЛЯРНОЙ ДИНАМИКИ С
ИСПОЛЬЗОВАНИЕМ ПАРАМЕТРОВ, ОПРЕДЕЛЯЕМЫХ КВАНТОВО-
ХИМИЧЕСКИМИ МЕТОДАМИ**

ОПИСАНИЕ ПРОГРАММЫ

RU.СНАБ.80066-06 13 54

ЛИСТОВ 40

Ине.№ подл.	Подп. и дата	Взам.инв.№	Ине.№ дубл.	Подп. и дата

2011

АННОТАЦИЯ

Документ содержит описание прикладного сервиса моделирования молекулярной динамики с использованием параметров, определяемых квантово-химическими методами. Прикладной сервис реализует композитное приложение для многоцелевой инструментально-технологической платформы (МИТП) CLAVIRE, которое определяет процесс последовательного исполнения пакетов GAMESS, реализующего квантово-химические методы, и NAMD реализующего метод МД. Прикладной сервис разработан в ходе выполнения проекта «Создание распределенной вычислительной среды на базе облачной архитектуры для построения и эксплуатации высокопроизводительных композитных приложений» (Договор № 21057 от 15 июля 2010 г., шифр 2010-218-01-209) в рамках реализации постановления Правительства РФ № 218 «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства».

СОДЕРЖАНИЕ

1.	ОБЩИЕ СВЕДЕНИЯ	4
2.	ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ	4
2.1.	Область применения	4
2.2.	Функциональное назначение	5
2.3.	Ограничения на применение	6
3.	ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	6
3.1.	Структура композитного приложения	6
3.2.	Характеристики модулей композитного приложения	9
3.3.	Описание модулей композитного приложения в МИТП	15
4.	ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА	29
5.	ВЫЗОВ И ЗАГРУЗКА	29
6.	ВХОДНЫЕ ДАННЫЕ	32
7.	ВЫХОДНЫЕ ДАННЫЕ	35
	ПЕРЕЧЕНЬ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	38
	ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	39

1. ОБЩИЕ СВЕДЕНИЯ

Прикладной сервис (ПС) моделирование молекулярной динамики (МД) с использованием параметров, определяемых квантово-химическими методами. RU.СНАБ.80066-06 01 54 реализует композитное приложение, которое позволяет проводить моделирование МД, используя параметры атом-атомного взаимодействия, определяемые квантово-химическими методами. ПС разработан в ходе выполнения проекта «Создание распределенной вычислительной среды на базе облачной архитектуры для построения и эксплуатации высокопроизводительных композитных приложений» (Договор № 21057 от 15 июля 2010 г., шифр 2010-218-01-209) в рамках реализации постановления Правительства РФ № 218 «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства».

ПС функционирует в рамках распределенной среды облачных вычислений под управлением многофункциональной инструментально-технологической платформы (МИТП) CLAVIRE RU.СНАБ.80066-06. Он разработан на предметно-ориентированном языке EasyFlow описания композитных приложений.

ПС использует следующие пакеты прикладных программ, доступные в распределенной среде под управлением МИТП (см. также раздел 3.2):

- GAMESS - прикладной пакет, предназначенный для моделирования средствами квантовой химии;
- NAMD - прикладной пакет, предназначенный для моделирования методом молекулярной динамики;

Перечисленные пакеты описываются на языке EasyFlow и регистрируются в базе пакетов МИТП.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

2.1. Область применения

ПС предназначен для моделирования методом молекулярной динамики с использованием параметров межатомного взаимодействия, вычисленными методами квантовой химии. Поэтому область применения ПС совпадает с областью применения метода МД и включает в себя исследование больших молекулярных систем при условии

корректности приближения классической механики, в частности, в отсутствии протекания химических реакций. Примерами применения метода МД являются задачи молекулярной биологии, задачи по расчету наносистем и наноконплексов.

В основе работы сервиса лежит тот факт, что квантово-химические методы позволяют весьма точно рассчитывать частоты колебаний молекул. Это дает возможность получать значения силовых постоянных, которые могут быть использованы в МД расчетах. В связи с тем, что подобные квантово-химические расчеты весьма трудоемки, в рамках данного сервиса они проводятся не для самого исследуемого объекта, а над некоторым задаваемым пользователем модельным объектом, наиболее близко отражающим особенности строения изучаемой системы. В данном ПС квантово-химический расчет силовых постоянных осуществляется с помощью пакета GAMESS, а МД моделирование – с использованием пакета NAMM.

2.2. Функциональное назначение

В методе МД траектории атомов определяются на основе знаний о параметрах их взаимодействия с другими атомами по законам классической механики. Поэтому точность, с которой известны параметры атом-атомного взаимодействия, в МД эксперименте имеют решающее значение для его достоверности. Таким образом, метод МД на практике оказывается применим только для систем, параметры взаимодействия для всех частиц которой известны. Данный прикладной сервис позволяет преодолеть указанное ограничение за счет вычисления параметров межатомного взаимодействия при помощи методов квантовой химии.

Данный сервис является альтернативой использованию и известных значений параметров взаимодействия, имея преимущество, поскольку позволяет вычислять их, учитывая особенности конкретной МД модели. Считается, что данные о потенциалах взаимодействия не являются жестко связанными с задачей. Однако, при применении метода МД к решению задач из области нанотехнологий данный тезис может нарушаться: одни и те же химические соединения могут использоваться при конструировании различных нанообъектов, внутри которых их геометрия может не соответствовать равновесной для изолированной молекулы. Поскольку табличные значения параметров взаимодействия известны только для равновесных состояний, возникает несоответствие модели, для которой параметры взаимодействия известны точно, и моделируемой структуры. Данный сервис позволяет избежать вызванных этим систематических ошибок за счет соответствующего выбора модельной молекулы.

Таким образом, прикладной сервис позволяет корректно проводить МД моделирование в случаях когда:

- неизвестны параметры взаимодействия для некоторых или всех типов атомов;
- параметры взаимодействия требуют уточнения из-за особенностей структуры моделируемой системы.

2.3. Ограничения на применение

Результатом работы ПС являются результаты МД моделирования исследуемой системы, корректность которых зависит от применимости метода МД как такового к моделированию происходящих в системе процессов, т. е. при условии корректности приближения классической механики.

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Структура композитного приложения

Композитное приложение описывает следующий процесс, реализуемый средствами МИТП-Ц. В данной реализации ПС предполагается, что расчеты для модельного объекта проводятся методом DFT/B3LYP/6-31G. Работа осуществляется в 4 этапа.

1. На 1-м этапе решается обычная задача оптимизации геометрии модельного объекта с применением указанных выше пакета и метода.
2. На 2-м этапе в качестве исходных данных задаются координаты атомов модельной системы, полученные на 1-м этапе, и определяется избыточная система внутренних координат. В результате решения задачи на этом этапе из выходного файла программы GAMESS находится число таких координат, а также их полный список, что необходимо для формирования входного файла для следующего этапа. Из этого же выходного файла определяется список тех внутренних координат, для которых должны быть определены значения (длины связей, валентные углы) и силовые постоянные. Для этого используется файл, содержащий информацию о соответствии между номерами атомов исследуемой и модельной систем.
3. На 3-м этапе в качестве исходных данных во входном файле должен присутствовать полный список всех внутренних координат, составляющих избыточную систему. В качестве результата выдается обычная для подобных

задач информация о частотах и формах нормальных колебаний, а также матрица гессиана в системе заданных внутренних координат. Для этого используется метод преобразования координат, описанный в работе [1]. В этой матрице ищутся силовые постоянные, номера которых определены на предыдущем шаге. Значения равновесных положений для соответствующих внутренних координат также считываются из этого же выходного файла. Значения тех параметров, которые рассматриваются как эквивалентные, усредняются.

4. На 4-м этапе осуществляется запуск программы NAMD для исследуемой системы. При этом для части параметров используются значения, полученные на предыдущем шаге, а часть параметров задается как обычно через входной файл параметров взаимодействия NAMD.

Структура композитного приложения приведена на рис. 3.1.

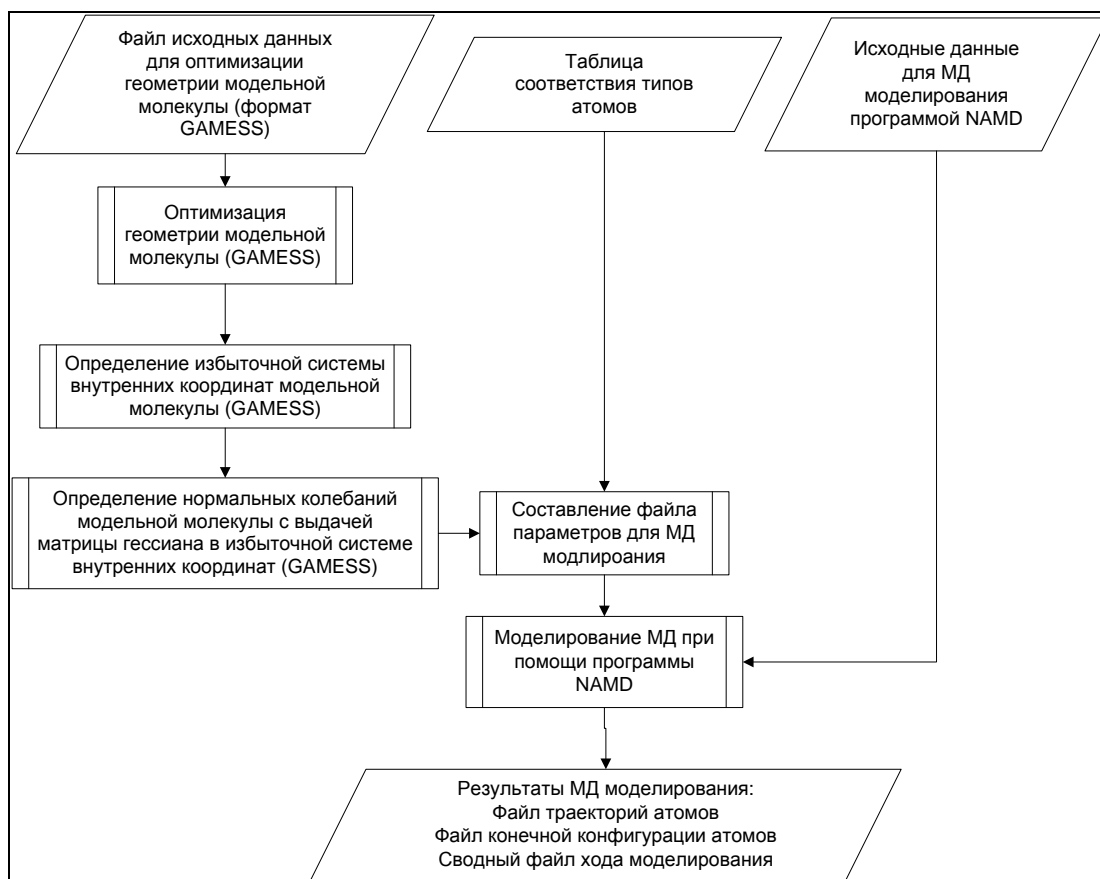


Рисунок 3.1 - Структура композитного приложения

Для реализации композитного приложения в соответствии со структурой на рис. 3.1, используются следующие прикладные пакеты:

- GAMESS - прикладной пакет, предназначенный для моделирования средствами квантовой химии;
- NAMD - прикладной пакет, предназначенный для моделирования методом молекулярной динамики

В листинге 3.1 приведен скрипт композитного приложения на языке EasyFlow, а на рис. 3.2 представлен результат его интерпретации в МИТП, который создает очередь из четырех связанных процессов запуска пакетов в виде AWF.

Листинг 3.1 Описание композитного приложения на языке EasyFlow

```
require gamess_inp, coordinates_inp, structure_inp, parameters_inp, namd_config,
cmp_table;
[mode = @meta]

step Step1 runs GAMESS_EXT
(
  gamess_input = games_input
);

step Step2 runs GAMESS_EXT
(
  molecule_xyz = Step1.Result.outs["atoms_xyz_file.txt"],
  basis = "6-31G",
  maxit = 0,
  calc_method = "DFT",
  use_DIRSCF = false,
wf_mode = "Standart",
  task = "gradient",
  useIChange = false,
  hs_send = false,
  ZMAT = "auto",
  FORCE = "PRTIFC=.T."
);

step Step3 runs GAMESS_EXT
(
  wf_mode = "Standart",
  in_bonds_index_file = Step2.Result.outs["bonds_index.txt"],
  molecule_xyz = Step1.Result.outs["atoms_xyz_file.txt"],
  basis = "6-31G",
  maxit = 0,
  calc_method = "DFT",
  use_DIRSCF = false,
  task = "hessian",
  useIChange = false,
  hs_send = false,
  ZMAT = "table",
  FORCE = "PRTIFC=.T. DECOMP=.T."
);

step Step4 runs NAMD_EXT
(
  coordinates = coordinates_inp,
  structure = structure_inp,
  namd_config = namd_config,
  length_trace_file = Step3.Result.outs["length_trace.txt"],
  hessian_trace_file = Step3.Result.outs[ "hessian_trace.txt" ],
  index_name_file = Step3.Result.outs[ "index_name.txt" ],
  bound_index_file = Step3.Result.outs[ "bonds_index.txt" ],
  cmpl_parameters_file = parameters_inp,
  av_table_file = cmp_table,
);
```

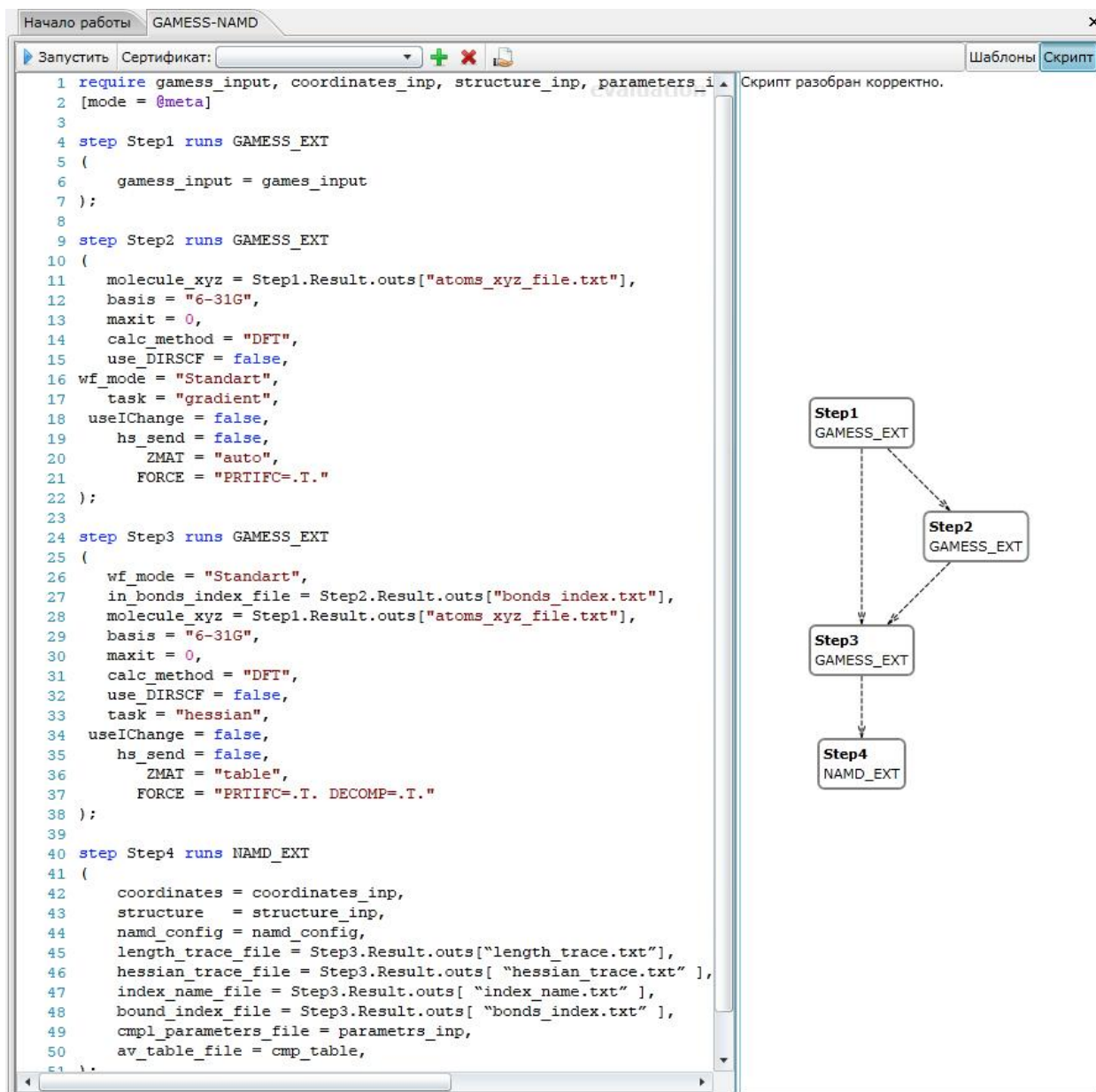


Рисунок 3.2 - Подготовка композитного приложения к запуску на выполнение в МИТП

3.2. Характеристики модулей композитного приложения

3.2.1. Прикладной пакет GAMESS моделирования методом квантовой химии

Прикладной пакет GAMESS разработан членами исследовательской группы Гордона в Государственном Университете штата Айова. Он представляет собой открытый проект для моделирования средствами квантовой химии. GAMESS имеет в своем арсенале широкий спектр методов и позволяет, в частности, рассчитывать колебательные частоты произвольной атомно-молекулярной системы путем прямого вычисления матрицы гессиана. В случае данного ПС основная цель применения этого пакета состоит в вычислении силовых постоянных для модельной молекулы.

Пакет GAMESS способен производить расчеты на базе набора высокоточных методов квантовой химии, позволяющих моделировать как основное, так и возбужденные состояния произвольных атомно-молекулярных систем, включая:

- метод Хартри–Фока, теорию функционала плотности,
- многоконfigurационное самосогласованное поле,
- обобщенный метод валентных связей,
- учет электронной корреляции по методу Меллера–Плессета второго порядка,
- конфигурационное взаимодействие позволяет провести коррекцию энергии электронной корреляции после процедуры самосогласования,
- учет различных поправок и эффектов.

Пакет представляет собой набор программных модулей, написанных на языке Fortran. Основные программные модули GAMESS поддерживают параллельный режим счета как на многопроцессорных компьютерах, так и в кластерах рабочих станций UNIX. Основой параллелизации кода является язык посылки сообщений.

Для более детального описания функционального назначения и структуры пакета можно воспользоваться документацией из источника [3].

Требования к аппаратной совместимости: персональный компьютер или узел кластера на базе процессора не менее Pentium IV, свободное дисковое пространство – не менее 500 МБ, объем оперативной памяти – не менее 512 МБ.

Требования к информационной и программной совместимости: программа предназначена для функционирования на персональном компьютере POSIX-совместимых операционных системах, OS X и Windows, также ряд других платформ Sun Solaris UltraSPARC/x86/x86-64, Linux x86/x86-64 и т.д.

Для вызова пакета GAMESS требуется в консольном приложении перейти в директорию, в которой находится приложение, и выполнить команду:

```
GAMESS Входной_файл.in > Выходной_файл.out.
```

В случае удачного запуска и отработки приложения в директории появится файл Выходной_файл.out. с выходными данными.

Входным параметром пакета является конфигурационный файл задания. Описание расчетной задачи обычно включает в себя описание основных химических параметров, параметров настройки ресурсов исполнения, параметров базиса. После описания расчетной задачи следует описание рассчитываемой молекулярной системы. Описание системы определяется следующими возможными основными типами координат – декартовые, координаты Хильдербрандта, Z-матрицы в стандарте MOPAC и GAUSSIAN,

собственный способ задания Z -матрицы. Декартовое описание представляет собой массив из пяти столбцов, каждая строка которого описывает один из атомов молекулы. В первом столбце задается тип атома, во втором номер в периодической таблице, далее – координаты по x , y и z (см. листинг на рис. 3.2.1). Более подробно это изложено в пользовательской инструкции [2].

```
$SYSTEM MWORDS=10 $END
$BASIS GBASIS=N31 NGAUSS=6 $END
$CONTRL SCFTYP=RHF RUNTYP=OPTIMIZE DFTTYP=B3LYP $END
$STATPT NSTEP=300 $END
$DATA
Benzene
C1
C1      6      -1.619      -0.745      0.000
C2      6      -1.619       0.655      0.000
C3      6      -0.406       1.355      0.000
C4      6       0.806       0.655      0.000
C5      6       0.806      -0.745      0.000
C6      6      -0.406      -1.445      0.000
H7      1      -2.554      -1.285      0.000
H8      1      -2.554       1.195      0.000
H9      1      -0.406       2.435      0.000
H10     1       1.741       1.195      0.000
H11     1       1.741      -1.285      0.000
H12     1      -0.406      -2.525      0.000
$END
```

Рисунок 3.2.1 - Входной файл, предназначенный для работы с программой GAMESS на 1-м этапе работы композитного приложения.

Результатом работы пакета является текстовый файл, состоящий из блоков, каждый из которых включает группу связанных данных. Данные, получающиеся в результате расчета, можно классифицировать по решаемой задаче.

3.2.1. Прикладной пакет GAMESS моделирования методом квантовой химии

Прикладной пакет NAMD разработан совместно группой теоретической и вычислительной биофизики и лабораторией параллельного программирования из Иллинойского университета в Урбане и Шампейне. представляет собой бесплатно-распространяемую программу с открытым кодом для выполнения расчета методом молекулярной динамики. Разнообразие реализованных в программе технологий параллельных вычислений обеспечивает эффективную масштабируемость и позволяет моделировать молекулярные системы отвечающих современным требованиям размеров. Возможность применения NAMD для конкретной задачи зависит от наличия знаний о молекулярном строении системы и параметрах взаимодействия для ее элементов. В данном случае параметры взаимодействия могут быть введены как входные данные, если

они известны заранее, или вычислены методами квантовой химии на предшествующем молекулярному моделированию этапе.

Пакет NAMD предназначен для моделирования систем методом молекулярной динамики в рамках нескольких следующих постановок задач:

- классическая молекулярная динамика (моделирование систем вблизи состояния равновесия),
- неравновесная молекулярная динамика (моделирования процессов релаксации, моделирование систем при наличии внешних сил),
- расчет изменения свободной энергии вдоль заданных множеств фазовых траекторий,
- минимизация энергии системы.

Программа NAMD написана на языке C++ и построена на основе библиотек Charm++. Поддерживает параллельный режим на многопроцессорных компьютерах, в кластерах рабочих станций UNIX (и под управлением других систем), на основе технологии CUDA.

Для более детального описания функционального назначения и структуры пакета можно воспользоваться документацией [4].

Требования к аппаратной совместимости: персональный компьютер или узел кластера на базе процессора не менее Pentium IV, свободное дисковое пространство – не менее 500 МБ, объем оперативной памяти – не менее 512 МБ.

Требования к информационной и программной совместимости: программа предназначена для функционирования на персональном компьютере POSIX-совместимых операционных системах, OS X и Windows, также ряд других платформ Sun Solaris UltraSPARC/x86/x86-64, Linux x86/x86-64 и т.д.

Запуск программы NAMD производится из командной строки. Команда запуска зависит от платформы и используемых для параллельного исполнения программно-аппаратных средств. При этом общим требованием является указание входного конфигурационного файла и выходного файла, куда будет помещена сводная информация о ходе выполнения расчета. В общем виде команда запуска из директории в которой находится пакет имеет вид:

```
[описание параллельного запуска] namd2 [ключ с указанием количества узлов] входной_файл > выходной_файл
```

Подробно синтаксис команд для различных платформ и режимов исполнения приведен в [5].

В случае удачного запуска приложения в выходном файле появятся данные о ходе выполнения и доклады о произошедших при выполнении ошибках.

Входным параметром пакета является конфигурационный файл задания. В нем содержится описание типа решаемой задачи, термодинамическое описание моделируемой системы, параметры расчета, имена файлов входных и выходных данных, описание внешних сил и полей.

Термодинамическое описание системы обычно включает в себя параметры термостата, баростата и граничные условия. Основными параметрами расчета являются шаг интегрирования, метод интегрирования, количество итераций.

Набор файлов входных данных зависит от задачи и базово включает в себя:

- файл описания начальных координат атомов в формате PDB;
- файл молекулярной структуры в формате X-PLOR PSF;
- файл, содержащий параметры взаимодействия для элементов системы в формате CHARMM.

Синтаксис конфигурационного файла имеет вид:

имя_параметра значение_параметра

Значение параметра может быть строкой, числом или таблицей чисел. Кроме того конфигурационный файл поддерживает синтаксис языка TCL. Более подробно информация изложена в пользовательской инструкции [6]. Пример содержания файла приведен на рис. 3.2.2.

```
structure          src/structure.psf
coordinates        src/coordinates.pdb
set temperature    300
set outputname    result
set outdir        output
paraTypeCharmm    on
parameters        src/parameters.inp
temperature        $temperature
cellBasisVector1  40.0    0.    0.
cellBasisVector2  0.    40.0    0.
cellBasisVector3  0.    0    140.0
cellOrigin        0.    0.    0.
wrapAll           on
exclude           scaled1-4
1-4scaling        1.0
cutoff            12.
switching         on
switchdist        10.
pairlistdist      14.5
timestep          0.2
rigidBonds        none
nonbondedFreq     10
fullElectFrequency 10
stepspercycle     10
PME               no
```

PMEGridSpacing	1.0
useGroupPressure	yes
useFlexibleCell	yes
useConstantArea	yes
langevinPiston	off
langevinPistonTarget	1.01325
langevinPistonPeriod	100.0
langevinPistonDecay	50.0
langevinPistonTemp	\$temperature
langevin	on
langevinDamping	5
langevinTemp	\$temperature
langevinHydrogen	no
outputName	\$outputname
restartname	\$outdir/\$outputname
restartfreq	1000
DCDfile	\$outdir/coord_dcd_\$outputname.dcd
DCDfreq	\$out_freq
velDCDfile	\$outdir/vel_dcd_\$outputname.dcd
velDCDfreq	\$out_freq
outputEnergies	100
outputPressure	100
reinitvels	\$temperature
run	40000

Рисунок 3.2.2 - Входной файл, предназначенный для работы с программой GAMESS на 1-м этапе работы композитного приложения.

Формат файла описания начальных координат соответствует формату секции координат файла типа Protein Data Bank PDB. Описание формата приведено в [7].

Формат файла описания структуры молекул приведен в [8].

Формат файла, содержащего параметры взаимодействия, зависит от выбранного потенциала из набора: CHARMM, AMBER или GROMACS. Полное описание форматов содержится в [9], [10] и [11] для приведенного списка соответственно.

Набор дополнительных входных файлов, требуемых при использовании некоторых постановок задач, и описание их формата приведено в руководстве пользователя NAMD [4].

Набор выходных данных определяется пользователем в конфигурационном файле и зависит от постановки задачи. Кроме выходного файла со сводной информацией о ходе моделирования, при моделировании классической молекулярной динамики (см. пункт 2.1) формируется набор файлов, приведенный в Таблице 3.2.1.

Таблица 3.2.1

Набор выходных файлов при моделировании классической молекулярной динамики

<i>Название</i>	<i>Тип</i>	<i>Содержание</i>
Лог-файл	Текстовый	Содержит сводную информацию о ходе моделирования
Restart coordinates	Текстовый,	Файл, содержит координаты атомов на

file	формат PDB или бинарный	последней итерации и информацию, необходимую для перезапуска расчета
Restart velocities file	Текстовый, формат PDB или бинарный	Файл, содержит скорости атомов на последней итерации и информацию, необходимую для перезапуска расчета
DCD coordinates file	Бинарный, формат DCD Charmm	Файл содержит траектории атомов в координатной плоскости

Информация о выходных файлах для расчета в других постановках задач находится в руководстве пользователя [4].

3.3. Описание модулей композитного приложения в МИТП

3.3.1. Прикладной пакет GAMESS генерации комплексной сети

Для интеграции пакета GAMESS в МИТП необходимо описать его взаимодействия с платформой, опираясь на формат представления входных и выходных данных, на языке EasyPackage. Платформенный скрипт описания пакета GAMESS позволяет определить уровень абстракции и интерпретации в работе с входными и выходными параметрами, что обеспечивает гибкость и упрощение процедуры взаимодействия пользователя с платформой на этапе запуска задания. Фрагмент платформенного скрипта описания пакета GAMESS определяется в соответствии с листингом 3.2.

Листинг 3.2. Фрагмент скрипта описания пакета GAMESS

```

name "GAMESS_EXT"
display_as "GAMESS"
vendor "Iowa State University"
url "http://www.msg.ameslab.gov/gamess/"
license "GPLv3"
description "The General Atomic and Molecular Electronic Structure System (GAMESS) is
a general ab initio quantum chemistry package."

inputs {
  meta param{
    name "calc_method"
    display_as "Method"
    #type string
    type enum ["TDDFT", "DFT"] #,"MP2"]
    default "DFT"
    required
  }

  meta param{
    name "package_method"
    display_as "package method"
    type string
    evaluator lambda { |ctx|
      if (ctx.calc_method == "DFT")
        return "RHF"
      elsif (ctx.calc_method == "TDDFT")
        return "RHF TDDFT=EXCITE"
      end
    }
  }
}

```



```

meta param{
  name "functional"
  ontology "escience.ifmo.ru/onto/functional"
  display_as "Functional"
  #type string
  type enum ["B3LYP"]
  default "B3LYP"
}

meta param{
  name "task"
  required
  display_as "Task"
  #type string
  type enum ["geometry optimization","single
             point","energy","gradient","hessian"]
  default "single point"
}

meta param{
  name "package_task"
  display_as "package task"
  type string
  evaluator lambda { |ctx|
    if (ctx.task == "single point") || (ctx.task == "energy")
      return "ENERGY"
    elsif (ctx.task == "geometry optimization")
      return "OPTIMIZE"
    elsif (ctx.task == "gradient")
      return "GRADIENT"
    elsif (ctx.task == "hessian")
      return "HESSIAN"
    end
  }
}

meta param{
  name "basis"
  required
  display_as "basis"
  #type string
  type enum ["6-31G","STO-3G","6-311++G(2d,p)","SBKJC","pc-3", "am-1"]
  default "6-31G"
}

...
auto file {
  name "gamess_input"
  filename "gamess.inp"
  place "/"
  assembler erb_template rtext("gamess_template.erb")
}

auto file {
  name "in_bonds_index_file"
  filename "in_bonds_index_file.txt"
  place "/"
  extractor BondIndexExtractor.new
}

...
cmdline lambda { |ctx| "{0} gamess.inp gamess.out" }
}
outputs {
  auto file {
    name "gamess_out"
    filename "gamess.out"
    place "/"
    extractor TotalEnergyGamessExtractor.new
  }

  meta param {
    name "atoms_xyz"
    type string
    default ""
  }
}

```

```
meta param {
  name "index_name"
  type string
  default ""
  validator lambda { |val, ctx| true }
}

meta file {
  name "index_name_file"
  filename "index_name.txt"
  place "/"
  assembler ObjectToSAssembler.new("index_name")
}

meta param {
  name "bonds_index"
  type string
  default ""
  validator lambda { |val, ctx| true }
}

meta file {
  name "bonds_index_file"
  filename "bonds_index.txt"
  place "/"
  assembler ObjectToSAssembler.new("bonds_index")
}

meta param {
  name "hessian_trace"
  type string
}

meta file {
  name "hessian_trace_file"
  filename "hessian_trace.txt"
  place "/"
  assembler ObjectToSAssembler.new("hessian_trace")
}

meta param {
  name "length_trace"
  type string
  default ""
  validator lambda { |val, ctx| true }
}

meta file {
  name "length_trace_file"
  filename "length_trace.txt"
  place "/"
  assembler ObjectToSAssembler.new("length_trace")
}

meta file {
  name "atoms_xyz_file"
  filename "atoms_xyz_file.txt"
  place "/"
  assembler ObjectToSAssembler.new("atoms_xyz")
}

...
}
```

В блоке `inputs` описаны входные данные. Параметры типа `file` определяют входные файлы, передаваемые программе GAMESS при запуске. В блоке `outputs` описаны выходные файлы, формируемые программой GAMESS, и извлекаемые из них параметры.

Входные и выходные файлы программы в МИТП описаны со следующими дополнительными атрибутами:

- «тип файла» определяет спецификатор `raw` или `auto`. Спецификатор `raw` при этом обозначает файлы, которые передаются напрямую от пользователя без изменений. Спецификатор `auto` предполагает дополнительно к прямой передаче файла возможность его автоматической генерации на базе других входных данных. Так, например, файл `games_input`, содержащий конфигурацию запуска GAMESS, может передаваться напрямую или может быть автоматически составлен на основе параметров (см. ниже), инициализированных пользователем при помощи скрипта на языке EasyFlow.
- «логическое имя файла» (`name`) – определяет имя файла через которое осуществляется его связывание с входными данными в скрипте на языке EasyFlow.
- «физическое имя файла» (`filename`) – определяет физическое имя файла.
- «местоположение файла» (`place`) – определяет физическое местоположение файла относительно директории, из которой будет произведен запуск пакета.
- «необходимый» `required` – данный атрибут отражает необходимость присутствия параметра при запуске.
- «экстракторы и ассемблеры» – атрибуты файлов, необходимые для нахождения какой-либо информации (конкретного значения, строки результата) внутри файла и создания новых файлов на основе указанных значений параметров и описанных шаблонов. Экстракторы и ассемблеры составляются в соответствии с синтаксисом языка Ruby.

Входные параметры программы в МИТП описаны со следующими дополнительными атрибутами:

- «информация о параметре» (`display_as`) – данный атрибут отражает текстовую информацию о параметре в интерфейсе пользователя;
- «значение по умолчанию» (`default`) – определяет значение по умолчанию. К примеру, для параметра `timestep` – установлено 2.
- «тип» (`type`) – необходим для отсеивания низкоуровневых ошибок, связанных с типизацией данных;
- «валидатор» (`validator`) – атрибут параметров, необходимый для отсеивания низкоуровневых ошибок, связанных с некорректными значениями параметров, валидаторы составляются в соответствии с синтаксисом языка Ruby;
- «эвалюатор» - атрибут параметра, позволяющие присваивать ему значение, вычисленное в соответствии с определенным в эвалюаторе алгоритмом на основе

значений других параметров; эвалюаторы составляются в соответствии с синтаксисом языка Ruby.

На листинге 3.3.2 приведен фрагмент экстрактора, выделяющего необходимое содержимое из выходного файла GAMESS в выходные параметры GAMESS. В числе таких параметров `hessian_trace`, `length_trace`, `length_trace`, `num_bonds`, `index_name` – предназначенные для передачи в этап по моделированию МД и для взаимодействия между собой запусков GAMESS при выполнении квантово-химических расчетов в рамках ПС (`atoms_xyz`).

Листинг 3.2 Фрагмент скрипта описания пакета GAMESS

```
class TotalEnergyGamessextractor < TextExtractor
  @format = "xyz"

  def extract(str)
    params = {"total_energy"=>"", "molecula_name"=>"", "atoms_xyz"=>"", "index_name"
=> "", "hessian_trace" => "", "length_trace"=>"", "bonds_index"=>"", "num_bonds"=>"",
"summary"=>""}
    xyz = str.to_s
    lines = xyz.split("\n")
    mode="searchPattern"
    index_name = []
    hessian = [[]]
    linesToMiss = 8;
    pr = false;
    pr1 = true;
    count = 0
    # $log.Trace("Starting to find total energy")

    # FSM
    lines.each { |line|
      line.strip!
      if (mode == "searchPattern")
        foundEnergyBlock = line.scan("ENERGY COMPONENTS")
        if (foundEnergyBlock != [])
          # $log.Trace("Found ENERGY COMPONENTS")
          mode = "energyBlockFound"
        end
        boundIndex = line.scan("--- ENCODED Z MATRIX ---")
        if (boundIndex != [])
          # $log.Trace("Found --- ENCODED Z MATRIX ---")
          mode = "bondIndexFound"
        end
        atomIndex = line.scan("INTERNUCLEAR DISTANCES (ANGS.)")
        if (atomIndex != [])
          # $log.Trace("Found INTERNUCLEAR DISTANCES (ANGS.)")
          mode = "atomIndexFound"
        end
        hessianTrace = line.scan("HESSIAN MATRIX IN INTERNAL
                                COORDINATES")

        if (hessianTrace != [])
          # $log.Trace("Found HESSIAN MATRIX IN INTERNAL COORDINATES")
          mode = "hessianFound"
        end
        foundSamples = line.scan("INTERNAL COORDINATES")
        if((foundSamples[0] != nil) and pr1)
          # $log.Trace("Found INTERNAL COORDINATES")
          mode="lengthFound"
          pr1 = false
        end
        foundAtoms = line.scan(/^ATOM\s+ATOMIC\s+COORDINATES\s+[A-
                                Z\(\)]+$/)

        if(foundAtoms!=[])
          # $log.Trace("Found ATOMS")
          mode = "atomsFound"
        end
      end
    }
  end
end
```

```

foundEqGeom = line.scan("***** EQUILIBRIUM GEOMETRY LOCATED
                        *****")
if(foundEqGeom != [])
#       $log.Trace("Found EQUILIBRIUM GEOMETRY LOCATED")
mode = "foundEqGeom"
end
foundMolName = line.scan("RUN TITLE")
if(foundMolName != [])
#       $log.Trace("Found RUNRUN TITLE")
mode = "foundMolName"
next
end
foundMolName = line.scan("SUMMARY OF TDDFT RESULTS")
if(foundMolName != [])
$log.Trace("SUMMARY OF TDDFT RESULTS")
mode = "foundSummaryTDDFT"
next
end
elseif (mode == "energyBlockFound")
#$log.Trace("Missing line")
linesToMiss -= 1
if (linesToMiss == 0)
mode = "totalEnergyLine"
end
elseif (mode == "totalEnergyLine")
#$log.Trace("Found Total Energy line")
foundSample = line.scan(/([\d\-\.]*)\s*$/i)
params["total_energy"] = foundSample.at(0).at(0)
mode = "searchPattern"
elseif (mode == "bondIndexFound")
#$log.Trace("Found boundIndex Line")
foundSample=line.scan(/^\d+\s+(\d+\s+\d+\s+\d+\s*\d*\s*\d*)$/);
if(foundSample[0] != nil)
params["bonds_index"] += foundSample[0][0].to_s + "\n"
pr = true
count+=1
elseif(pr)
mode = "searchPattern"
params["bonds_index"] = params["bonds_index"].chop
params["num_bonds"] = count.to_s
pr = false
end
elseif (mode == "atomIndexFound")
#$log.Trace("Found atomIndex Line")
foundSample=line.scan(/^\d+\s+([A-Z\d+])\s*[\d\s*\.\.]+$$/);
if(foundSample[0] != nil)
index_name[foundSample[0][0].to_f]=foundSample[0][1]
pr = true
elseif(pr)
params["index_name"]=index_name
mode = "searchPattern"
pr = false
end
elseif (mode == "hessianFound")
foundSample=line.scan(/^\d+\s+(\d\.\-)+\s+(\d\.\-
)*\s*(\d\.\-)*\s*(\d\.\-)*\s*(\d\.\-)*$/);
if(foundSample[0] != nil)
test_sample = line.scan(/^\d\s+$/i)
if(test_sample[0]!=nil)
next
end
if(hessian[foundSample[0][0].to_i] != nil )
hessian[foundSample[0][0].to_i] =
hessian[foundSample[0][0].to_i] + foundSample[0]
hessian[foundSample[0][0].to_i].delete_at(-6)
else
hessian[foundSample[0][0].to_i] = foundSample[0]
hessian[foundSample[0][0].to_i].delete_at(0)
end
pr = true
elseif(pr)
foundSamples = line.scan("-----")
if(foundSamples[0] != nil)
mode="searchPattern"
pr = false;
i=0;
while (i<hessian.size())

```

```

        params["hessian_trace"]+=hessian[i][i-
            1].to_s + " "
        i+=1
    end
    params["hessian_trace"] = params["hessian_trace"].strip
end
end
elsif(mode == "atomsFound")
    foundSample = line.scan(/^[A-Z\d]+\s+[\d\.\-]+\s+[\d\.\-
        ]+\s+[\d\.\-]+\s+[\d\.\-]+$/)
    if(foundSample[0]!=nil)
        pr = true
    elsif(pr)
        mode = "searchPattern"
        pr = false
    end
end
elsif (mode == "foundEqGeom")
    foundSample = line.scan(/^[A-Z\d]+\s+[\d\.\-]+\s+[\d\.\-
        ]+\s+[\d\.\-]+\s+[\d\.\-]+$/)
    if(foundSample[0]!=nil)
        params["atoms_xyz"] += foundSample[0] + "\n"
        pr=true
    elsif(pr)
        params["atoms_xyz"] = params["atoms_xyz"].to_s
        mode = "searchPattern"
        pr = false
    end
end
elsif (mode == "foundMolName")
    foundSamples = line.scan("-----")
    if(foundSamples[0]!=nil)
        next
    else
        foundSamples = line.scan(/^[A-Za-z]+$/)
        params["molecula_name"] += foundSamples[0].to_s
        mode = "searchPattern"
        next
    end
end
elsif (mode=="lengthFound")
    #log.Trace("Found bondLength Line")
    foundSamples = line.scan(/^\d+\s+[A-Z]+\s+(\d+\s+)+[\d\.\-
        ]+\s+(\d\.\-+)$/)
    if(foundSamples[0]!= nil)
        params["length_trace"] += foundSamples[0][-1] + " "
        pr = true
    elsif(pr)
        mode = "searchPattern"
        pr = false
    end
end
...
}
puts params
return params
end
end

```

На листинге 3.3.3 приведены два параметра, второй из которых составляется посредством эвалюатора в зависимости от первого. Данный эвалюатор используется при автоматическом составлении конфигурационного файла GAMESS на третьем этапе (см. листинг 3.1) с использованием избыточной системы внутренних координат, экстрактированной из файла `in_bonds_index_file` в параметр `in_bonds_index`.

На листинге 3.3.3 приведены два параметра, второй из которых составляется посредством эвалюатора в зависимости от первого. Данный эвалюатор используется при автоматическом составлении конфигурационного файла GAMESS на третьем этапе (см. листинг 3.1) с использованием избыточной системы внутренних координат, экстрактированной из файла `in_bonds_index_file` в параметр `in_bonds_index`.

Листинг 3.3. Описание параметра GAMESS с эвалюатором используется при автоматическом составлении конфигурационного файла GAMESS

```

meta param{
  name "ZMAT"
  display_as "type ZMAT"
  type enum ["none","auto","table"]
  default "none"
}

meta param{
  name "package_ZMAT"
  display_as "package_ZMAT"
  type string
  default ""
  evaluator lambda{ |ctx|
    if ctx.ZMAT=="auto"
      return "$ZMAT DLC=.T. AUTO=.T. $END"
    elsif(ctx.ZMAT=="table")
      return "$ZMAT IZMAT(1)= \n" +
        ctx.in_bonds_index + "\n$END"
    else
      return ""
    end
  }
}

```

Помимо представленных параметров на примере вышеприведенных скриптов определяются также другие параметры пакета GAMESS.

3.3.2 Платформенное описание прикладного пакета NAMD как модуля композитного приложения

Для интеграции пакета NAMD в МИТП необходимо описать его взаимодействия с платформой, опираясь на формат представления входных и выходных данных, на языке EasyPackage. Платформенный скрипт описания пакета NAMD позволяет определить уровень абстракции и интерпретации в работе с входными и выходными параметрами, что обеспечивает гибкость и упрощение процедуры взаимодействия пользователя с платформой на этапе запуска задания. Фрагмент платформенного скрипта описания пакета NAMD определяется в соответствии с листингом 3.2.

Листинг 3.4 Фрагмент описания пакета NAMD в МИТП

```

name "NAMD_EXT"
display_as "NAMD"
license "GPLv3"

inputs {
  auto file {
    name "namd_config"
    required
    place "/"
    filename "namd_config.conf"
    assembler erb_template rtext("namd_config_template.erb")
  }
}

raw file{
  name "structure"
  required
}

```

```

    place "/src/"
    filename "structure.psf"
}

raw file{
    name "coordinates"
    required
    place "/src/"
    filename "coordinates.pdb"
    extractor CellBasisExtractor.new
}

auto file{
    name "parameters"
    required
    place "/src/"
    filename "parameters.inp"
    assembler erb_template rtext("par_nanotubes_template.erb")
}

auto file{
    name "cml_parameters_file"
    display_as "complimented parameters file"
    filename "parameters_0.inp"
    extractor ParametersExtractor.new
}

...

meta param{
    name "num_of_iter"
    display_as "Number of iterations for dynamics simulation"
    type int
    default 5000
    validator lambda { |val, ctx| val > 0 }
}

meta param{
    name "timestep"
    type float
    default 2.0
    display_as "Time step, fs"
}

meta param{
    name "temperature"
    type float
    default 300.0
    validator lambda { |val, ctx| val > 0 }
    display_as "Temperature of system in K"
}

meta param{
    name "termostat_on_off"
    type enum ["on","off"]
    default "on"
    display_as "Termostat on/off"
}

...
}

cmdline lambda { |ctx| "{0} " }

outputs {
    raw file{
        name "logfile"
        display_as "Logfile"
        filename "log.log"
        place "/"
    }

    raw file{
        name "coord"
        display_as "Coordinate file"
        filename "result.coord"
        place "/output/"
    }
}

```



```
raw file{
name "vel"
display_as "velocity file"
filename "result.vel"
place "/output/"
}

raw file{
name "dcd_coor"
display_as "DCD coordinate file"
filename "coor_dcd_result.dcd"
place "/output/"
}

raw file{
name "dcd_vel"
display_as "DCD velocity file"
filename "vel_dcd_result.dcd"
place "/output/"
}
}
```

В блоке `inputs` описаны входные данные. Параметры типа `file` определяют входные файлы, передаваемые программе NAMD при запуске. В блоке `outputs` описаны выходные файлы, формируемые программой NAMD.

Входные и выходные файлы программы в МИТП описаны со следующими дополнительными атрибутами:

- «тип файла» определяет спецификатор `raw` или `auto`. Спецификатор `raw` при этом обозначает файлы, которые передаются напрямую от пользователя без изменений. Спецификатор `auto` предполагает дополнительно к прямой передачи файла возможность его автоматической генерации на базе других входных данных. Так, например, файл `parameters`, содержащий параметры взаимодействия атомов при стандартном запуске NAMD передается напрямую, а при работе в прикладном сервисе генерируется автоматически на основе результатов расчетов методами квантовой химии. Как `auto` также обозначен конфигурационный файл NAMD `namd_config`, который может быть автоматически составлен на основе параметров (см. ниже), инициализированных пользователем при помощи скрипта на языке EasyFlow.
- «логическое имя файла» (`name`) – определяет имя файла через которое осуществляется его связывание с входными данными в скрипте на языке EasyFlow.
- «физическое имя файла» (`filename`) – определяет физическое имя файла.
- «местоположение файла» (`place`) – определяет физическое местоположение файла относительно директории запуска пакета.
- «необходимый» `required` – данный атрибут отражает необходимость присутствия параметра при запуске.

- «экстракторы и ассемблеры» – атрибуты файлов, необходимые для нахождения какой-либо информации (конкретного значения, строки результата) внутри файла и создания новых файлов на основе указанных значений параметров и описанных шаблонов. Экстракторы и ассемблеры составляются в соответствии с синтаксисом языка Ruby.

Входные параметры программы в МИТП описаны со следующими дополнительными атрибутами:

- «информация о параметре» (`display_as`) – данный атрибут отражает текстовую информацию о параметре в интерфейсе пользователя;
- «значение по умолчанию» (`default`) – определяет значение по умолчанию, например, для параметра `timestep` – установлено 2;
- «тип» (`type`) – необходим для отсеивания низкоуровневых ошибок, связанных с типизацией данных;
- «валидатор» (`validator`) – атрибут параметров, необходимый для отсеивания низкоуровневых ошибок, связанных с некорректными значениями параметров, валидаторы составляются в соответствии с синтаксисом языка Ruby;
- «эвалюатор» – атрибут параметра, позволяющие присваивать ему значение, вычисленное в соответствии с определенным в эвалюаторе алгоритмом на основе значений других параметров, эвалюаторы составляются в соответствии с синтаксисом языка Ruby.

На листинге 3.5 приведены экстракторы копирующие содержимое передаваемых из квантово-химической части расчета файлов в параметры описания программы NAMD для функционирования в рамках ПС.

Листинг 3.5. Два экстрактора, копирующих содержимое передаваемых из квантово-химической части расчета файлов в параметры описания программы NAMD

```
class ParametersExtractor < TextExtractor
  def extract(str)
    Params = {}
    Params["parameters_string"] = str
    puts Params
    return Params
  end
end

class HessianTraceExtractor < TextExtractor
  def extract(str)
    Params = {}
    Params["hessian_trace"] = str
    puts Params
    return Params
  end
end
```

```

class LengthTraceExtractor < TextExtractor
  def extract(str)
    Params = {}
    Params["length_trace"] = str
    puts Params
    return Params
  end
end

class IndexNameExtractor < TextExtractor
  def extract(str)
    Params = {}
    Params["index_name"] = str
    puts Params
    return Params
  end
end

class BoundIndexExtractor < TextExtractor
  def extract(str)
    Params = {}
    Params["bound_index"] = str
    puts Params
    return Params
  end
end

class AvTableExtractor < TextExtractor
  def extract(str)
    Params = {}
    Params["av_table"] = str
    puts Params
    return Params
  end
end

```

На листинге 3.6 приведен параметр с эвалюатором, в котором производится расчет параметров атом-атомного взаимодействия на основе данных квантово-химического расчета.

Листинг 3.6. Описания параметра NAMD с эвалюатором, в котором производится расчет параметров атом-атомного взаимодействия на основе данных квантово-химического расчета в рамках ПС

```

meta param {
  name "atom_table"
  type string
  default ""
  evaluator lambda { |ctx|
    av_table = []
    scale_const = [2239.378502, 0.19102249, 0.19102249]
    gamess = ctx.av_table.to_s
    gamess.delete! "\r"
    lines = gamess.split("\n")
    fs="";
    lines.each { |line|
      line.strip!
      fs=line.scan(/([A-Za-z\s\d]+):\s*([A-Za-z\s\d]+);$/);
      if (fs[0] != nil)
        av_table.push(fs[0])
      end
    }
    av_table.each { |line|
      line[1].split("\s").each { |arr|
        line.push(arr) }
      line[1]=line.size-2
    }
  }
}

```

```

hissian_trace_str = ctx.length_trace
length_trace_str = ctx.hessian_trace

index_name_str = ctx.index_name
bound_index_str = ctx.bound_index
index_name_map = {}
name_index_map = {}
module_map = {}
finalTable = {}
k = 0;

for molclass in av_table

  for i in 2..molclass.length - 1
    currMol = molclass[i];
    module_map[currMol.to_s] = molclass[0]
    index_name_str = index_name_str.delete("[\\"")#!!!!
    lines = index_name_str.split(", ")

    for j in 1..lines.length - 1
      if (currMol == lines[j])
        name_index_map[currMol] = j
        index_name_map[j] = currMol
      end
    end
  end
end

length_lines = length_trace_str.split(" ")
hissian_lines = hissiian_trace_str.split(" ")

links = {}
count = 0;
lines = bound_index_str.split("\n")
lines.each { |link|
  count+=1
  link.insert(0, count.to_s+ " ")
}
puts lines
lines.each { |link|
  fs=link.scan(/([\d]+)/);
  if (fs != [])
    if (fs[1][0].to_s == "1")
      if (index_name_map[fs[2][0].to_i] != nil && index_name_map[fs[3][0].to_i]
!= nil)
        links[fs[0][0]] = [fs[1][0],
          module_map[index_name_map[fs[2][0].to_i]],
          module_map[index_name_map[fs[3][0].to_i]],
          nil, length_lines[fs[0][0].to_i - 1],
          hissiian_lines[fs[0][0].to_i - 1]]
        end
      elsif (fs[1][0].to_s == "2")
        if (index_name_map[fs[2][0].to_i] != nil &&
          index_name_map[fs[3][0].to_i] != nil &&
          index_name_map[fs[4][0].to_i] != nil)
          links[fs[0][0]] = [fs[1][0],
            module_map[index_name_map[fs[2][0].to_i]],
            module_map[index_name_map[fs[3][0].to_i]],
            module_map[index_name_map[fs[4][0].to_i]],
            length_lines[fs[0][0].to_i - 1],
            hissiian_lines[fs[0][0].to_i - 1]]
          end
        end
      end
    end
  }
}
puts links
trash = {}
count_map = {}
count = 0
for tLine_out in links
  if (trash[tLine_out.to_s] == nil)
    for tLine_in in links
      if (tLine_in.to_s != tLine_out.to_s && trash[tLine_in.to_s] == nil)
        if (tLine_out[1][1] == tLine_in[1][1] && tLine_out[1][2] ==
tLine_in[1][2] && tLine_out[1][3] == tLine_in[1][3]) ||

```

```

        (tLine_out[1][1] == tLine_in[1][1] && tLine_out[1][2] ==
tLine_in[1][3] && tLine_out[1][3] == tLine_in[1][2]) ||
        (tLine_out[1][1] == tLine_in[1][2] && tLine_out[1][2] ==
tLine_in[1][3] && tLine_out[1][3] == tLine_in[1][1]) ||
        (tLine_out[1][1] == tLine_in[1][3] && tLine_out[1][2] ==
tLine_in[1][1] && tLine_out[1][3] == tLine_in[1][2])
        tLine_out[1][4] = tLine_out[1][4].to_f +
tLine_in[1][4].to_f
        tLine_out[1][5] = tLine_out[1][5].to_f +
tLine_in[1][5].to_f
        count+=1
        if (count_map[tLine_out.to_s] == nil)
            count_map[tLine_out.to_s] = 0
        end
        count_map[tLine_out.to_s] += 1

        trash[tLine_in.to_s] = tLine_in
    end
end
end
tLine_out[1][5] = tLine_out[1][5].to_f/(count+1)
tLine_out[1][4] = tLine_out[1][4].to_f/(count+1)
count = 0
end
end

result = [[],[],[ ]]
i = 0
puts links
for tLine_out in links
    if (trash[tLine_out.to_s] == nil)
        ppos1 = tLine_out[1][4].to_s.index(".")+4
        ppos2 = tLine_out[1][5].to_s.index(".")+4
        if (tLine_out[1][2] == nil)
            result[tLine_out[1][0].to_i-1][i] = tLine_out[1][1].to_s + " " +
tLine_out[1][2].to_s + " " + tLine_out[1][4].to_s + " " +
tLine_out[1][5].to_s
        elsif
            result[tLine_out[1][0].to_i-1][i] = tLine_out[1][1].to_s + "\t"
tLine_out[1][2].to_s + "\t" + tLine_out[1][3].to_s + "\t" +
(tLine_out[1][4].to_f*scale_const[tLine_out[1][0].to_i-1]/2.0).to_s[0..ppos1]
"\t" + tLine_out[1][5].to_s[0..ppos2]
            result[tLine_out[1][0].to_i-1].delete(nil)
        end
        i += 1
    end
end
end

str_result = ["" , "" , "" ]
count=0

result.each{|type|
    type.each{|bond|
        str_result[count] += bond.to_s + "\n"
    }
    str_result[count] += ";"
    count+=1
}
$log.Trace(str_result)
return str_result.join(" ")
}

```

Помимо представленных параметров на примере вышеприведенных скриптов определяются также другие параметры пакета NAMD.

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

ПС функционирует в рамках распределенной среды облачных вычислений под управлением многофункциональной инструментально-технологической платформы (МИТП) CLAVIRE RU.СНАБ.80066-06. Для использования PS необходима рабочая станция с подключением к Интернет, со следующими минимальными характеристиками:

- архитектура процессора – x86, x86_64, IA64;
- объем оперативной памяти – 1 ГБ;
- объем свободного пространства на жестком диске – 1 ГБ;
- тактовая частота процессора – 1 ГГц.

Для работы с композитным приложением необходимо использовать браузеры Mozilla FireFox (версия 3.0 и выше), Google Chrome (версия 13 и выше), Opera (версия 9.0 и выше) и Internet Explorer (версия 7.0 и выше).

5. ВЫЗОВ И ЗАГРУЗКА

Исполнение PS выполняется средствами МИТП CLAVIRE. При запуске композитное приложение описывает следующий процесс, реализуемый в среде облачных вычислений средствами МИТП:

1. По заданному перечню параметров запуска (в тексте программы на языке EasyFlow) определяются входные данные пакетов;
2. На основе мониторинга доступных вычислительных ресурсов МИТП строит оптимальное (с точки зрения минимизации общего времени выполнения) расписание исполнения цепочки запусков;
3. Экземпляр пакета моделирования при помощи квантово-химических методов GAMESS запускается на исполнение в режиме оптимизации геометрии на выбранном вычислительном ресурсе;
4. По окончании расчета экземпляра пакета GAMESS выходные данные (в форме файлов) передаются на вход пакету GAMESS, сконфигурированному для поиска избыточной системы внутренних координат;
5. По окончании расчета экземпляра пакета GAMESS выходные данные (в форме файлов) вместе с выходными данными первого расчета GAMESS, передаются на вход пакету GAMESS сконфигурированному для расчета частот, форм нормальных колебаний и матрицы гесса в избыточной системе координат;

6. После завершения работы пакета GAMESS выходные файлы передаются пакету NAMD;
7. После окончания работы приложения выходные данные передаются в хранилище данных, и пользователь уведомляется средствами МИТП об успешном выполнении задания.

Загрузка файлов в систему производится в левой панели интерфейса. Для этого надо нажать правую кнопку мыши на пункте "Файлы" и в контекстном меню выбрать пункт "добавить файл", как показано на рис. 5.1

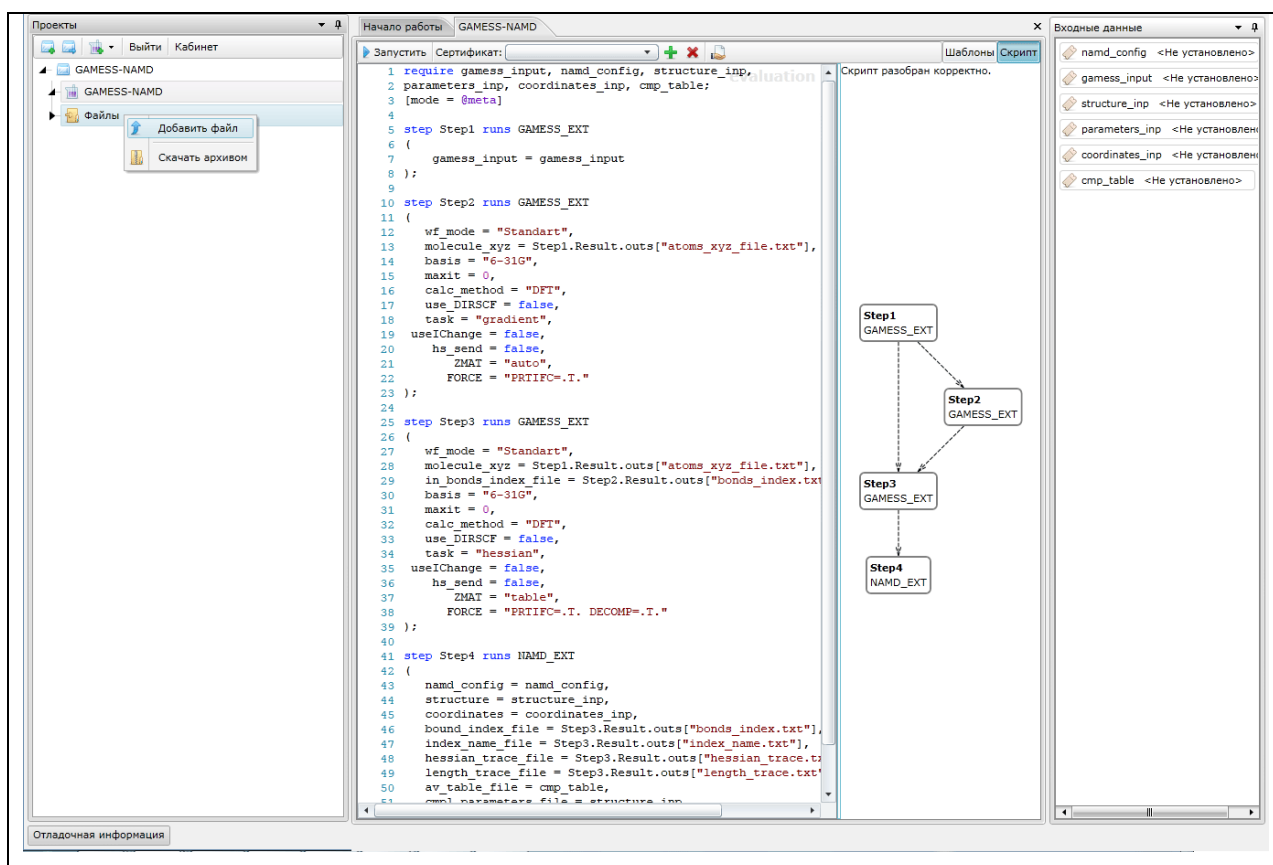


Рисунок 5.1 - Внешний вид и процесс загрузки входных файлов в интерфейса

МИТП-Ц

После загрузки всех необходимых входных файлов (см. ниже) следует установить соответствие их с логическими именами, используемым в описании ПС на языке EasyFlow. Они перечислены после ключевого слова `require` и отображают на панели справа. Соответствие необходимо установить, как это показано в таблице 5.1.

Таблица 5.1

Порядок установления соответствия логических имен входных файлов

Логическое имя файла	Содержание файла
gamess_input	конфигурационный файл GAMESS для

	оптимизации геометрии модельной молекулы
namd_config	конфигурационный файл NAMD
coordinates_inp	файл описания начальных координат атомов в формате PDB
structure_inp	файл описания молекулярной структуры в формате X-PLOR PSF
parameters_inp	файл, содержащий известные (но не вычисляемые на квантово-химическом этапе) параметры взаимодействия для элементов системы в формате CHARMM
cmp_table	файл, определяющий соответствие типов атомов в МД расчете и атомов в модельной молекуле

Для запуска расчета необходимо нажать на кнопка “Запустить”. На экране появится информация о проведения инициализации задания и далее о его ходе выполнения в соответствии со схемой ПС. На рисунке 5.2 показан CWF в процессе исполнения композитного приложения под управлением МИТП.

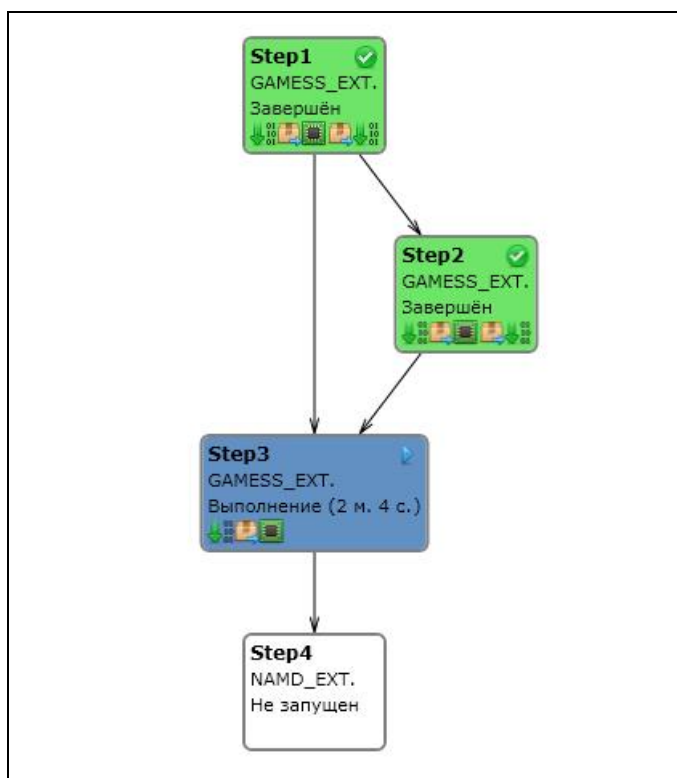


Рисунок 5.2 - CWF в процессе исполнения композитного приложения под управлением МИТП

После завершения работы задания набор выходных файлов, полученных с каждого этапа можно увидеть в интерфейсе МИТП. Для этого следует развернуть пункт меню соответствующий выполненному запуску в левой панели интерфейса как показано на рис. 5.3 и открыть в нем пункт “Результаты”.

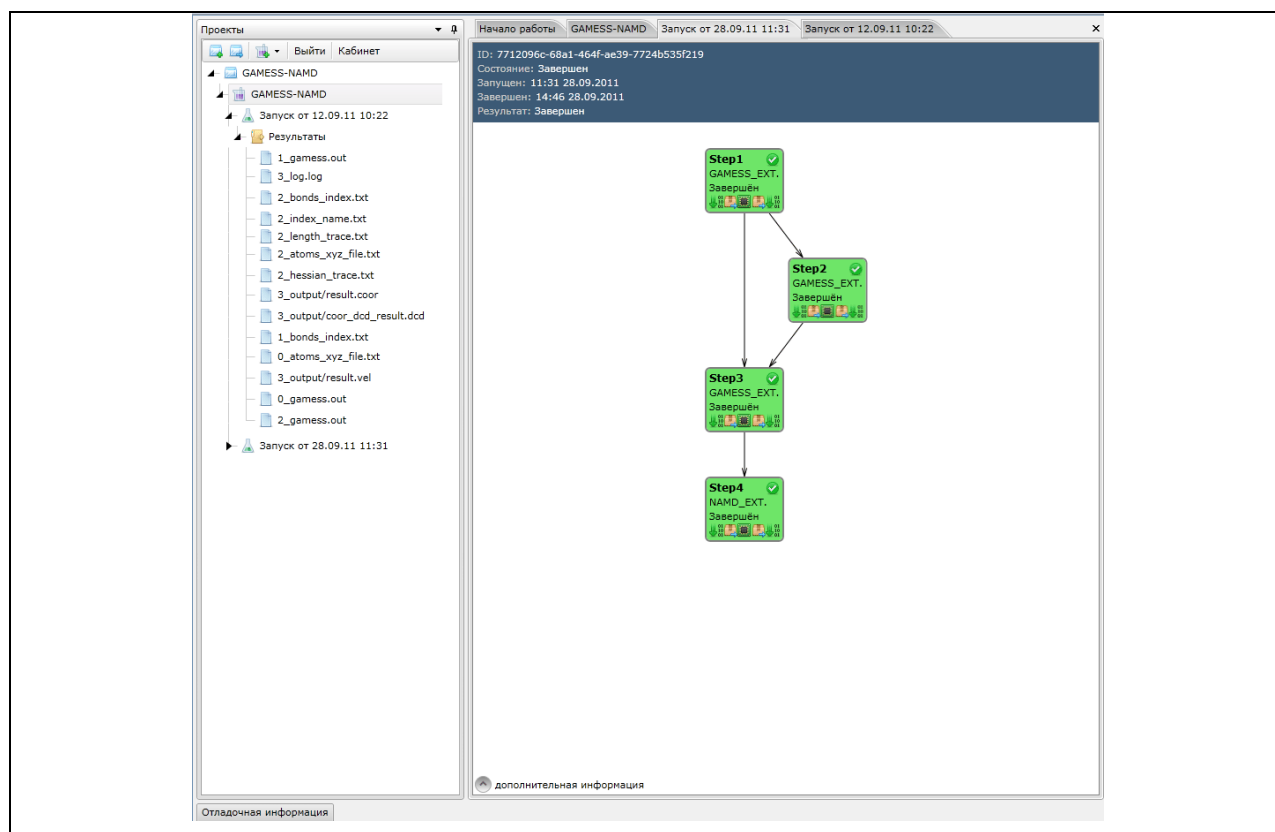


Рисунок 5.3 - Завершенное композитного приложения под управлением МИТП

В появившемся списке файлов префикс у имени указывает на номер этапа, в результате работы которого данный выходной файл был получен. Нумерация производится с 0. Таким образом в списке выходных файлов можно увидеть три сводных файла расчета с именем gamess.out, соответствующие этапам расчета с использованием прикладного пакета GAMESS, набор файлов, служащих для передачи данных между этапами, и набор выходных файлов программы NAMD. Последний являются конечным продуктом работы ПС и включают в себя текстовый сводный файл процесса МД моделирования log.log; бинарные файл конечной конформации системы для координат и скоростей атомов: result.coor и result.vel соответственно; бинарный файл траектории движения атомов coor_dcd_result.dcd. Указанные бинарные файлы могут быть визуализированным при помощи любого подходящего молекулярного визуализатора или редактора. К примеру, рекомендуемой разработчиками NAMD программы VMD [12].

6. ВХОДНЫЕ ДАННЫЕ

Управление композитным приложением в контрольном примере осуществляется посредством набора входных файлов, содержащих все необходимые параметры,

определяющие выполнение расчета. Входные данные контрольного примера состоят из приведенного ниже набора файлов:

- Конфигурационный файл GAMESS для оптимизации геометрии модельной молекулы. Пример его содержания приведен на рис. 3.2.1. Формат входного файла программы GAMESS описан в пункте 3.2 и в [2]. При перечислении атомов модельной молекулы необходимо использовать имена со сквозной нумерацией, т. к. именно такая нумерация предполагается для составления файла сопоставления типов атомов в МД расчете и атомов в модельной молекуле (см. ниже).
- Конфигурационный файл NAMD для этапа моделирования МД. Пример его содержания приведен на рис. 3.2.2. Формат файла описан в пункте 3.2 и в [6].
- Файл описания начальных координат атомов в формате PDB. Формат файла описан в [7]. Типы атомов, используемые в файле должны соответствовать типам, перечисленным в файле описания молекулярной структуры и в файле параметров взаимодействия (см. ниже).
- Файл описания молекулярной структуры в формате. Формат файла описан в [8].
- Файл, содержащий известные параметры взаимодействия для элементов систем. При работу композитного приложения возможно использование только формата CHARMM. Формат описан в [9]. При этом отличие от стандартного формата при работе с ПС состоит в описании не всех необходимых параметров взаимодействия, а только тех, что известны и не предполагаются к расчету методами квантовой химии. Пример такого файла приведен на рис. 6.1.

```
* This is a reduced version of the CHARMM22 parameter file for
* a nanotube simulation in water
*

BONDS
!V(bond) = Kb(b - b0)**2
!
!atom type Kb          b0

HT  HT      0.000      1.5139
OT  HT      450.000     0.9572

ANGLES
!V(angle) = Ktheta(Theta - Theta0)**2
!V(Urey-Bradley) = Kub(S - S0)**2
!
!atom types      Ktheta  Theta0  Kub      S0
HT  OT  HT      55.000   104.5200

DIHEDRALS
!V(dihedral) = Kchi(1 + cos(n(chi) - delta))
!
!atom types      Kchi    n    delta
CA  CA  CA  CA    3.1000  2   180.00
```

```

NONBONDED nbxmod 5 atom cdie1 shift vatom vdistance vswitch -
cutnb 14.0 ctofnb 12.0 ctonnb 10.0 eps 1.0 e14fac 1.0 wmin 1.5
!V(Lennard-Jones) = Eps,i,j[(Rmin,i,j/ri,j)**12 - 2(Rmin,i,j/ri,j)**6]
!
!atom ignored epsilon Rmin/2
CA 0.000000 -0.070000 1.992400
HT 0.000000 -0.046000 0.224500
OT 0.000000 -0.152100 1.768200

HBOND CUTNB 0.5

END

```

Рисунок 6.1 - Пример входного файла параметров взаимодействия для работы с ПС

В предполагаемой системе присутствуют валентные связи между атомами с типом CA. Однако, для таких связей указан только потенциал двугранных углов в разделе DIHEDRALS и отсутствует информация о потенциале валентных связей и валентных углов в разделах BONDS и ANGLES соответственно. Это означает, что параметры для данных потенциалов будут рассчитаны в ходе выполнения ПС, после чего файл параметров будет автоматически дополнен. Формирование данного файла связано с файлом, описанном в следующем пункте.

- Файл, определяющий соответствие типов атомов в МД расчете и атомов в модельной молекуле. Пример файла приведен на рис. 6.2.

```
CA1: C1 C2 C3 C4 C5 C6;
```

Рисунок 6.2 - Пример файла соответствия квантово-химического и МД расчетов

Каждой строчке соответствует один тип атомов в МД расчете. В файл потенциалов взаимодействия (рис. 6.1) будут записаны потенциалы для связей из числа перестановок по 2 для линейных связей, по 3 для связей валентных углов и по 4 для связей двугранных углов (торсионных) при условии, что эти связи встречаются в модельной молекуле. В случае, если одному типу связи соответствует несколько связей в модельной молекуле, их параметры будут усреднены. Лексемы после двоеточия, перечисленные через пробел, соответствуют меткам атомов во входном файле GAMESS. Лексема перед двоеточием соответствует названию типа атома в файле описания потенциалов взаимодействия в формате CHARMM.

Указанные входные файлы для запуска необходимо загрузить в систему МИТП и установить соответствие с логическими именами входных данных ПС, как показано в таблице 5.1.

7. ВЫХОДНЫЕ ДАННЫЕ

В результате выполнения расчета при помощи композитного приложения окончательный набор выходных файлов формируется в соответствии с таблицей 3.2.1 и представляет собой результаты молекулярно-динамического расчета для определенной во входных данных системой с параметрами взаимодействия, рассчитанными методами квантовой химии.

О ходе моделирования можно судить по сводному текстовому лог-файлу. Фрагмент лог-файла, содержащий сводную информацию о начале расчета и о его корректном завершении, приведен в листинге. 7.1.

Листинг 7.1. Выходной лог-файл расчета контрольного примера

```
Info: *****
Info: STRUCTURE SUMMARY:
Info: 20136 ATOMS
Info: 15894 BONDS
Info: 14592 ANGLES
Info: 17670 DIHEDRALS
Info: 0 IMPROPERs
Info: 0 CROSTERMS
Info: 0 EXCLUSIONS
Info: 60408 DEGREES OF FREEDOM
Info: 8712 HYDROGEN GROUPS
Info: 3 ATOMS IN LARGEST HYDROGEN GROUP
Info: 8712 MIGRATION GROUPS
Info: 3 ATOMS IN LARGEST MIGRATION GROUP
Info: TOTAL MASS = 138937 amu
Info: TOTAL CHARGE = 0 e
Info: MASS DENSITY = 1.02998 g/cm^3
Info: ATOM DENSITY = 0.0898929 atoms/A^3
Info: *****
Info:
Info: Entering startup at 0.469 s, 9.85938 MB of memory in use
Info: Startup phase 0 took 0 s, 9.85938 MB of memory in use
Info: Startup phase 1 took 0.016 s, 13.6328 MB of memory in use
Info: Startup phase 2 took 0 s, 13.8516 MB of memory in use
Info: Startup phase 3 took 0 s, 13.8516 MB of memory in use
Info: PATCH GRID IS 2 (PERIODIC) BY 2 (PERIODIC) BY 8 (PERIODIC)
Info: PATCH GRID IS 1-AWAY BY 1-AWAY BY 1-AWAY
Info: REMOVING COM VELOCITY 0.0047413 -0.0336342 0.00874194
Info: LARGEST PATCH (10) HAS 663 ATOMS
Info: Startup phase 4 took 0.0309999 s, 18.3633 MB of memory in use
Info: Startup phase 5 took 0 s, 18.3633 MB of memory in use
Info: Startup phase 6 took 0 s, 18.3633 MB of memory in use
LDB: Central LB being created...
Info: Startup phase 7 took 1.219 s, 18.3633 MB of memory in use
Info: CREATING 618 COMPUTE OBJECTS
Info: useSync: 1 useProxySync: 0
Info: NONBONDED TABLE R-SQUARED SPACING: 0.0625
Info: NONBONDED TABLE SIZE: 769 POINTS
Info: ABSOLUTE IMPRECISION IN FAST TABLE ENERGY: 3.38813e-021 AT 11.8295
Info: RELATIVE IMPRECISION IN FAST TABLE ENERGY: 5.45773e-014 AT 11.9138
Info: ABSOLUTE IMPRECISION IN FAST TABLE FORCE: 8.47033e-022 AT 11.9138
Info: RELATIVE IMPRECISION IN FAST TABLE FORCE: 1.6856e-015 AT 11.9138
```

RU.СНАБ.80066-06 13 54

Info: ABSOLUTE IMPRECISION IN VDWA TABLE ENERGY: 9.86076e-032 AT 11.8295
 Info: RELATIVE IMPRECISION IN VDWA TABLE ENERGY: 6.75473e-014 AT 11.9138
 Info: ABSOLUTE IMPRECISION IN VDWA TABLE FORCE: 4.62223e-032 AT 11.9138
 Info: RELATIVE IMPRECISION IN VDWA TABLE FORCE: 2.12237e-015 AT 11.9138
 Info: ABSOLUTE IMPRECISION IN VDWB TABLE ENERGY: 4.1359e-025 AT 11.8295
 Info: RELATIVE IMPRECISION IN VDWB TABLE ENERGY: 7.56853e-015 AT 11.9138
 Info: ABSOLUTE IMPRECISION IN VDWB TABLE FORCE: 3.87741e-026 AT 11.9138
 Info: RELATIVE IMPRECISION IN VDWB TABLE FORCE: 5.97409e-016 AT 11.9138
 Info: Startup phase 8 took 0.0150001 s, 18.4609 MB of memory in use
 Info: Startup phase 9 took 0 s, 22.6719 MB of memory in use
 Info: Finished startup at 1.75 s, 22.8047 MB of memory in use

TCL: Minimizing for 500 steps

ETITLE:	TS	BOND	ANGLE	DIHED
IMPRP	ELECT	VDW	BOUNDARY	MISC
KINETIC	TOTAL	TEMP	POTENTIAL	TOTAL3
TEMPAVG	PRESSURE	GPRESSURE	VOLUME	PRESSAVG
GPRESSAVG				

ENERGY:	0	5304.3197	1635.4386	1847.2809
0.0000	-54792.9505	66601701.4000	0.0000	0.0000
0.0000	66555695.4887	0.0000	66555695.4887	66555695.4887
0.0000	82666768.6332	82671913.2333	224000.0000	82666768.6332
82671913.2333				

MINIMIZER SLOWLY MOVING 68 ATOMS WITH BAD CONTACTS DOWNHILL

ENERGY:	1	5304.3197	1635.4386	1847.2809
0.0000	-54699.3162	1190993.7452	0.0000	0.0000
0.0000	1145081.4683	0.0000	1145081.4683	1145081.4683
0.0000	1482631.0943	1487605.5126	224000.0000	1482631.0943
1487605.5126				

. . .

ENERGY:	10500	7368.8728	3439.8289	2608.1128
0.0000	-73244.7990	19398.1196	0.0000	0.0000
16132.0603	-24297.8047	268.7731	-40429.8650	-24011.6214
273.7732	624.5684	-2206.4967	224000.0000	-2300.7572
2300.6482				

WRITING EXTENDED SYSTEM TO RESTART FILE AT STEP 10500

WRITING COORDINATES TO DCD FILE AT STEP 10500

WRITING COORDINATES TO RESTART FILE AT STEP 10500

FINISHED WRITING RESTART COORDINATES

The last position output (seq=10500) takes 0.016 seconds, 67.215 MB of memory in use

WRITING VELOCITIES TO RESTART FILE AT STEP 10500

FINISHED WRITING RESTART VELOCITIES

The last velocity output (seq=10500) takes 0.000 seconds, 66.598 MB of memory in use

WRITING EXTENDED SYSTEM TO OUTPUT FILE AT STEP 10500

WRITING COORDINATES TO OUTPUT FILE AT STEP 10500

CLOSING COORDINATE DCD FILE

The last position output (seq=-2) takes 0.031 seconds, 67.168 MB of memory in use

WRITING VELOCITIES TO OUTPUT FILE AT STEP 10500

The last velocity output (seq=-2) takes 0.016 seconds, 66.598 MB of memory in use

=====

WallClock: 701.812988 CPUTime: 701.812012 Memory: 66.597656 MB

Program finished.

Информация, выводимая в лог-файл, зависит от конфигурационного файла NAMD как описано в руководстве [6].

Другие выходные файлы из таблицы 3.2.1 в данной реализации ПС выводятся в бинарном формате. Они могут быть открыты для визуализации и анализа любым подходящим молекулярно-динамическим визуализатором или молекулярным редактором.

ПЕРЕЧЕНЬ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. J.A.Boatz and M.S.Gordon, J.Phys.Chem., 93, 1819-1826 (1989)
2. http://www.msg.ameslab.gov/gamess/GAMESS_Manual/input.pdf
3. <http://www.msg.chem.iastate.edu/gamess/documentation.html>
4. <http://www.ks.uiuc.edu/Research/namd/2.8/ug/>
5. <http://www.ks.uiuc.edu/Research/namd/2.8/ug/node43.html>
6. <http://www.ks.uiuc.edu/Research/namd/2.8/ug/node11.html>
7. <http://www.wwpdb.org/documentation/format33/sect9.html>
8. <http://www.ks.uiuc.edu/Training/Tutorials/namd/namd-tutorial-unix-html/node21.html>
9. <http://www.ks.uiuc.edu/Training/Tutorials/namd/namd-tutorial-unix-html/node23.html>
10. <http://ambermd.org/#ff>
11. <http://www.gromacs.org/Documentation>
12. <http://www.ks.uiuc.edu/Research/vmd/current/ug/ug.html>

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

МД	Молекулярная динамика
МИТП	Многофункциональная инструментально-технологическая платформа
ПС	Прикладной сервис

