

# Разработка инструментальных средств автоматизации построения и применения имитационных моделей РВС на языке Java

## Введение

Сегодня в области развития и применения высокопроизводительных вычислений наметились тенденции к организации проблемно-ориентированных распределенных вычислительных сред (РВС) различного назначения [2, 3, 5]. Как правило, такие среды обладают рядом свойств, существенно усложняющих процессы их построения, эксплуатации и исследования. К свойствам такого рода, например, относятся: организационно-функциональная разнородность, динамичность и неполнота описания интегрируемых в РВС ресурсов; наличие различных категорий пользователей, преследующих свои цели и задачи эксплуатации вычислительной среды. С точки зрения исследования РВС, важными задачами в настоящее время являются испытание таких систем и оценка их надежности [4]. Стандартизированные средства управления распределенными вычислениями, используемые в РВС, не позволяют в полной мере решить указанные выше проблемы. В этом случае требуется разработка специальных инструментальных средств, обеспечивающих возможность исследования надежности функционирования проблемно-ориентированных РВС (надежности процессов решения ресурсоемких задач в этих средах).

В настоящее время существует широкий спектр методов и средств [8] исследования различных аспектов функционирования РВС. Однако, как правило, они являются узкоспециализированными, в том смысле, что обеспечивают исследование вычислительной среды, в том числе ее программной или аппаратной надежности, путем аналитического, имитационного или полунатурного моделирования со слабой привязкой к особенностям предметной области решаемых задач.

В данной работе рассматривается подход к разработке инструментального средства для спецификации агрегированных многоуровневых моделей РВС с применением Java-технологий, обеспечивающего детальное описание всех аспектов решения задач в этих средах. Языковые средства спецификации модели, разрабатываемые в рамках данного подхода, являются развитием исследований, представленных в [6, 7].

## Проблемно-ориентированная распределенная вычислительная среда

В работе рассматриваются проблемно-ориентированные РВС, предназначенные для решения фундаментальных и прикладных вычислительных задач и характеризующиеся рядом особенностей, представленных ниже.

*Масштабность.* Общее число процессоров (ядер), суммарный объем памяти, число пользователей, а также число приложений, одновременно работающих/выполняющихся

в вычислительных средах, может быть очень большим. С ростом показателей этих параметров увеличивается сложность обеспечения эффективности проведения расчетов. С другой стороны, прирост этих показателей позволяет решать более сложные задачи.

*Распределенность.* Вычислительные узлы среды могут быть размещены на значительном расстоянии друг от друга.

*Неоднородность.* Узлы, входящие в состав вычислительной среды могут быть построены с применением различных архитектур, работать под управлением отличающихся друг от друга операционных систем, а так же поддерживать связь между собой с помощью коммуникационного оборудования, обладающего различной пропускной способностью и линий связи. В качестве узлов системы могут выступать вычислительные кластеры. Кластеры организуются на базе как выделенных, так и невыделенных вычислительных машин, и, следовательно, существенно различаются по степени надежности своих вычислительных ресурсов.

*Динамичность.* В состав вычислительной среды могут встраиваться новые узлы, которые необходимо задействовать для проведения текущих расчетов. Также узлы могут непредсказуемо отключаться, что не должно приводить к аварийному завершению вычислительного процесса.

*Различная административная принадлежность.* Узлы вычислительной среды могут принадлежать различным категориям пользователей, в их числе пользователи, нуждающиеся в высокоуровневых средствах организации вычислительного процесса решения задачи, обладать собственными режимами работы и иметь собственные настройки политик безопасности.

*Разнообразие спектра задач,* решаемых с помощью вычислительной среды.

*Выполнение заданий.* Задание пользователя представляет собой спецификацию процесса решения задач, содержащую информацию о требуемых вычислительных ресурсах, исполняемых прикладных программах, входных/выходных данных, а также другие необходимые сведения. Множество заданий пользователей рассматривается с точки зрения теории очередей и представляется в виде совокупности потоков заданий с приоритетами. Поток заданий характеризуется следующими свойствами: динамичностью, стохастичностью, неоднородностью, отсутствием обратной связи, неординарностью, стационарностью. Свободных ресурсов системы может быть недостаточно для одновременного обслуживания всех заданий, находящихся в очередях.

На рис. 1 представлена схема функционирования РВС [1]. Условно можно выделить несколько компонентов среды, взаимодействующих между собой: интегрированная кластерная система, предназначенная для выполнения заданий; администраторы и пользователи системы; мультиагентная система, служащая для оптимального распределения заданий и отправки на выполнение на кластере или группы кластеров; система мониторинга, сообщающая управляющему агенту информацию о состоянии интегрированной кластерной системы.

Администратор(ы) интегрированной кластерной системы с помощью политик управления манипулирует управляющим агентом, давая задание на выполнение на кластере. Управляющий агент распределяет задание между агентами. Каждый агент «знает» какими ресурсами он может управлять, характеристики этих ресурсов, типы задач, которые могут решаться быстрее, используя подконтрольные ему ресурсы и т. д. Информацию обо всех агентах «знает» управляющий агент. При поступлении задания от администратора системы к управляющему агенту последний, определяя тип задания, перенаправляет задание определенному агенту, который в свою очередь отправляет за-

дание на выполнение на подконтрольные ему ресурсы. Однако ресурсы могут быть частично заняты из-за использования их конечными пользователями кластера. Система мониторинга служит для информирования управляющего агента о происходящих процессах на кластерах (занятость ресурсов, процент выполнения заданий и другие важные сведения).

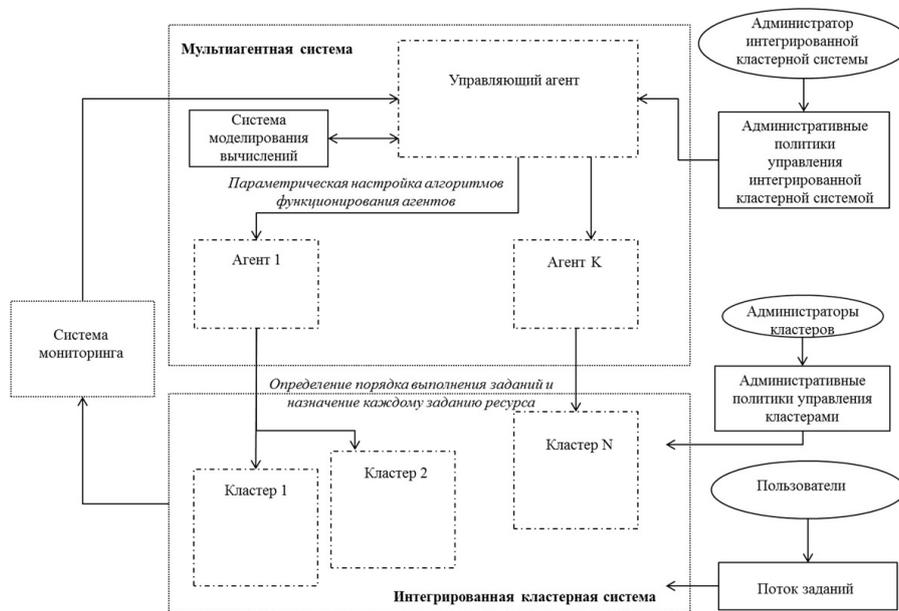


Рис. 1. Схема функционирования РВС

Для имитационного моделирования систем такого рода в ИДСТУ СО РАН разрабатывается специализированный инструментальный комплекс «SIRIUS-II», архитектура которого представлена на рис. 2.

Подсистема спецификации модели представляет собой инструментальное средство, обеспечивающее: описание имитационной модели с высокой степенью привязки к конкретной предметной области; трансляцию данного описания в термины языка SQL для получения файла базы данных; автоматическое построение пользовательского графического интерфейса для взаимодействия пользователей различных категорий с инструментальным комплексом и наполнения специфицированной модели реальными данными. Построение пользовательского графического интерфейса осуществляется с помощью системы АРМГис [9, 10].

Разрабатываемое инструментальное средство функционирует на верхнем уровне архитектуры инструментального комплекса, отвечает за подсистему спецификации модели и обеспечивает работу с различными категориями пользователей: разработчиками моделей и конечными пользователями, использующими разработанные модели.

Разработчик модели описывает имитационную модель на специальном входном языке. Затем модель транслируется в описание терминов языка SQL, по которому создается файл БД. Данный файл передается системе АРМГис для построения графического пользовательского интерфейса программы, после чего разработчик модели может наполнять ее реальными данными. Описания имитационных моделей хранятся в специальной БД – библиотеке имитационных моделей. Каждый объект модели хранится в виде класса, описанного на языке Java. Данная технология предполагает создание биб-

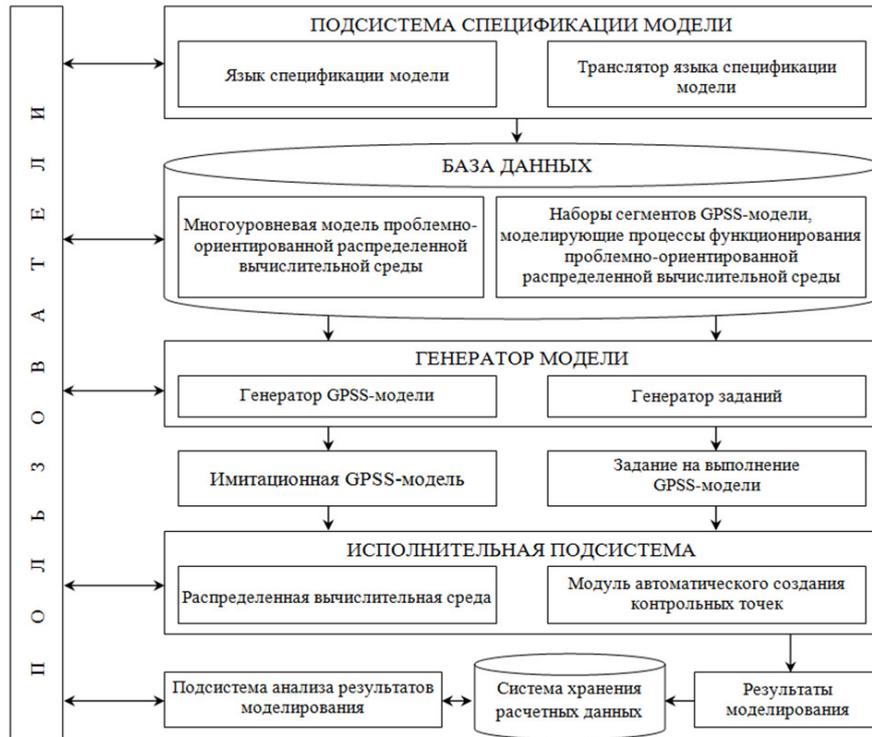


Рис. 2. Архитектура инструментального комплекса

лиотеки описаний моделей, которые впоследствии могут быть использованы другими предметниками для создания своих моделей. Существует возможность импорта данных из уже готовых моделей, обеспечивающая комплексирование по данным.

Язык спецификации модели предназначен для описания модели РВС и включает в себя набор базисных символов (символы входного алфавита, константы, имена операторов, специальные символы) и множество операторов. Операторы специфицируют операции создания, изменения и удаления элементов модели, а также задания отношений между этими элементами. Таким образом, описание модели РВС представляет собой последовательность операторов, разделенных между собой специальными символами. Ниже приведен пример описания модели РВС.

Пусть необходимо проанализировать нагрузку локальной компьютерной сети и выявить ее «узкие» места. При этом известно, что информация, запрашиваемая пользователем с некоторого компьютера, может быть ему передана только в том случае, если пользователь имеет доступ к этому компьютеру.

Опишем данную модель на языке спецификации. Для этого выделим объекты предметной области, которые будут участвовать в модели: узлы сети (объект NODES); концентраторы, к которым подключены узлы (объект HUBS); подключение концентраторов между собой (объект HUB2HUB); подключение узлов к концентраторам (объект NODESHUB); права доступа (объект ACCESS) – список пар компьютеров, определяющих право доступа одного компьютера к другому компьютеру. Текст модели на языке спецификации приведен на рис. 3.

```

new model 'C:\project\network';
create object NODES(ID_NODE INTEGER NOT NULL, NODE_NAME VARCHAR(20) NOT NULL);
create object HUBS(ID_HUB INTEGER NOT NULL, PORT_COUNT INTEGER NOT NULL);
create object HUB2HUB(HUB_NUM_IN INTEGER NOT NULL, HUB_NUM_OUT INTEGER NOT
NULL);
create object NODESHUB(ID_HUB INTEGER NOT NULL, ID_NODE INTEGER NOT NULL);
create object ACCESS(ID_NODE_ASK INTEGER NOT NULL, ID_NODE_ANSW INTEGER NOT
NULL);
create object TASK(ID_NODE_SEND INTEGER NOT NULL, ID_NODE_RECEIVE INTEGER NOT
NULL);
change object NODES setPK(ID_NODE);
change object HUBS setPK(ID_HUB);
change object HUB2HUB setUNIQUE(HUB_NUM_IN, HUB_NUM_OUT);
change object NODESHUB setUNIQUE(ID_HUB, ID_NODE);
change object ACCESS setUNIQUE(ID_NODE_ASK, ID_NODE_ANSW);
change object TASK setUNIQUE(ID_NODE_SEND, ID_NODE_RECEIVE);
change object HUB2HUB setFK(HUB_NUM_IN) REF HUBS(ID_HUB);
change object HUB2HUB setFK(HUB_NUM_OUT) REF HUBS(ID_HUB);
change object NODESHUB setFK(ID_HUB) REF HUBS(ID_HUB);
change object NODESHUB setFK(ID_NODE) REF NODES(ID_NODE);
change object ACCESS setFK(ID_NODE_ASK) REF NODES(ID_NODE);
change object ACCESS setFK(ID_NODE_ANSW) REF NODES(ID_NODE);
change object TASKsetFK(ID_NODE_SEND) REF NODES(ID_NODE);
change object TASKsetFK(ID_NODE_RECEIVE) REF NODES(ID_NODE);

```

Рис. 3. Текст на языке спецификации модели

Данное описание транслируется в термины языка SQL, после чего, полученный файл, передается системе АРМГис для построения графического пользовательского интерфейса, с помощью которого обеспечивается возможность наполнения модели данными. Пример автоматически построенного графического пользовательского интерфейса представлен на рис. 4.

ID_NODE_ASK	NODE_NAME	ID_NODE_ANSW	NODE_NAME_1
1	4-310-1	107	4-306-23
1	4-310-1	85	4-306-1
1	4-310-1	61	4-307-7
1	4-310-1	59	4-307-5
2	4-310-2	59	4-307-5
3	4-310-3	14	4-312-10
3	4-310-3	100	4-306-16
10	4-312-6	100	4-306-16
10	4-312-6	2	4-310-2
10	4-312-6	15	4-312-11
10	4-312-6	52	4-311-5
100	4-306-16	52	4-311-5
95	4-306-11	21	4-312-17
95	4-306-11	37	4-309-8
23	4-312-19	44	4-309-15

Рис. 4. Графический пользовательский интерфейс

На основе специфицированной модели РВС строится имитационная модель. В инструментальном комплексе (рис. 2) применяется каркасный подход к построению имитационных моделей. В рамках этого подхода в качестве модулей используются фрагменты имитационной модели (модули), моделирующие процессы функционирования объектов исследуемой системы. При каркасном подходе любая формируемая из модулей

пакета версия программы включает два компонента. Первый, постоянный компонент – каркас, не меняющийся от версии к версии и несущий в себе гнезда для размещения сменных модулей. Второй, переменный компонент – содержимое гнезд каркаса. Все многообразие формируемых версий достигается только за счет варьирования содержимого гнезд в разнообразных комбинациях.

Использование модульного подхода обеспечивает достаточно гибкую модификацию и «безболезненное» развитие математического и программного базиса для моделирования исследуемой РВС посредством добавления или замены модулей этого базиса новыми модулями, в том числе модулями уже разработанных библиотек шаблонов типовых объектов исследуемых РВС.

### **Заключение**

В статье рассмотрено инструментальное средство для спецификации агрегированных многоуровневых моделей РВС. Данное средство позволяет описывать имитационные модели РВС с высокой степенью привязки к предметной области; обеспечивает открытый доступ к библиотекам моделей и позволяет использовать их в качестве шаблонов для создания новых моделей. Исследование выполнено при финансовой поддержке РФФИ, проект №15-29-07955-офи\_м.

## **Список литературы / References**

- [1] BOGDANOVA V. G., BYCHKOV I. V., KORSUKOV A. S., OPARIN G. A., FEOKTISTOV A. G. Multiagent Approach to Controlling Distributed Computing in a Cluster Grid System // *Journal of Computer and Systems Sciences International*. – 2014. – Vol. 53. – № 5. – P. 713–722.
- [2] Бычков И. В., Корсуков А. С., Опарин Г. А., Феокистов Г. А. Инструментальный комплекс для организации гетерогенных распределенных вычислительных сред // *Информационные технологии и вычислительные системы*. – 2010. – № 1. – С. 45-54.
- [3] Бычков И. В., Опарин Г. А., Феокистов Г. А., Корсуков А. С. Децентрализованное управление потоками заданий в интегрированной кластерной системе // *Вестник Новосибирского государственного университета. Серия: Информационные технологии*. – 2011. – Т. 9. – Вып. 2. – С. 42-54.
- [4] Бычков И. В., Опарин Г. А., Феокистов Г. А., Корсуков А. С. Испытание и оценка надежности интегрированных кластерных систем на основе их комплексного моделирования // *Вестник компьютерных и информационных технологий*. – 2013. – № 3. – С. 3-8.
- [5] Бычков И. В., Опарин Г. А., Феокистов Г. А., Корсуков А. С. Распределение заданий в интегрированной кластерной системе на основе их классификации // *Вычислительные технологии*. – 2013. – Т. 18. – № 2. – С. 25–32.
- [6] Опарин Г. А., Феокистов Г. А., Александров А. А. Графическая инструментальная среда для описания модели распределенной вычислительной системы // *Вестник Иркутского государственного технического университета*. – 2006. – № 2 (26). – С. 30-34.
- [7] Опарин Г. А., Феокистов Г. А., Вартанян Э. К. Входной язык объектно-ориентированной базы знаний Grid-системы // *Программные продукты и системы*. – 2012. – № 1. – С. 3-6.
- [8] Самиоваров О. И., Кузюрин Н. Н., Грушин Д. А., Аветисян А. И., Михайлов Г. М., Рогов Ю. П. Проблемы моделирования GRID-систем и их реализации // *Научный сервис в сети Интернет: решение больших задач: Тр. Всерос. науч. конф. М.: МГУ, 2008. – С. 83-88.*

- [9] ФЕРЕФЕРОВ Е. С., БЫЧКОВ И. В., ХМЕЛЬНОВ А. Е. Технология разработки приложений баз данных на основе декларативных спецификаций // Вычислительные технологии. – 2014. – Т. 19. – № 5. – С. 85-100.
- [10] ФЕРЕФЕРОВ Е. С., ХМЕЛЬНОВ А. Е. Автоматизация создания пользовательского интерфейса на основе модели приложения баз данных // Вестник Бурятского государственного университета. – 2013. – № 9. – С. 100-118.