

Д.т.н., проф. Н.Б. Парамонов, к.т.н. Ю.В. Морозов (ОАО «ИНЭУМ им. И.С. Брука»),
И.В. Минин (ЗАО «МЦСТ»), Д.А. Токарев (УПМИ СП),
Ю.Н. Парамонов (ОАО «ИНЭУМ им. И.С. Брука»)

N. Paramonov, Y. Morozov, I. Minin, D. Tokarev, Y. Paramonov

МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ НА РАЗНЕСЁННОМ СТЕНДЕ

MODELING OF INFORMATION SYSTEMS ON THE DIVERSE STAND

Моделирование является основным инструментом анализа сложных технических систем. В статье показан подход к реализации моделей технических систем на основе вычислительных средств, включающих процессоры разного типа. Даны описания стенда отладки и оптимизации моделирующих программ.

Modeling is a key tool in the analysis of complex technical systems. The article describes the approach to implementing models of technical systems on computational tools, including different types of processors. Given the description of debugging and optimizing booth modeling programs.

Ключевые слова: моделирование, испытания, сложные технические системы.

Keywords: modeling, tests, big technical systems.

Введение

Моделирование является основным, а иногда и единственным, инструментом анализа сложных технических систем. В ходе создания моделирующих программ требуется решить целый спектр задач, одними из которых являются отладка моделей и оптимизация их использования на конкретных вычислительных средствах.

1. Моделирование сложных технических систем

Множество способов описания компонент информационных систем позволяет формировать соответствующее множество версий моделирующих программ.

Ограничимся классом математических моделей, создаваемых на принципах функционального подобия анализируемой системы путем функционального объединения моделей элементов и подсистем в единый комплекс программно-реализованных алгоритмов. С помощью таких моделей можно осуществить имитацию процессов для всего многообразия входных условий и текущих состояний реальной системы [1]. Для использования подобных моделей чрезвычайно важным является наличие вычислительных систем высокой производительности. Причем реализацию модели или ее существенной компоненты на новых вычислительных средствах можно рассматривать как получение новой версии модели, а возможности вычислительных средств во многом определяют возможности проводимого вычислительного эксперимента на этой модели.

Согласно обобщенной схеме (по Самарскому) вычислительный эксперимент включает следующие этапы моделирования системы (рис. 1):

- *1 этап* – построение качественной, например физической, модели изучаемой проблемы, т.е. описания (задания) исследуемого объекта на естественнонаучном языке. Функциональное описание объекта задает возможные допущения и ограничения при его анализе;

- *2 этап* – построение математических моделей систем на общем уровне. Это построение некоторого оператора $A: X \Rightarrow Y$ преобразования множества входных данных X в множество результатов Y . Построение математической модели (оператора A) включает выполнение определенных действий:

- выбор уровня детализации для проведения моделирования;
- выбор метода математического описания предметной области;
- построение математической модели;
- проверку корректности математической модели;



Рис. 1

Вычислительный эксперимент с накоплением знаний

• *3 этап* – построение дискретной модели (моделей) объекта. На этом этапе осуществляется разработка численных методов и алгоритмов, позволяющих формализовать математические модели в виде имитационных моделей. При этом осуществляются:

- разработка алгоритмов;
- исследование разработанных алгоритмов с целью определения их адекватности математической модели;
- реализация алгоритмов в виде комплекса моделей;

• *4 этап* – программная реализация алгоритмов с учетом особенностей вычислительной среды. Результатом этого этапа является моделирующая программа (комплекс моделирующих программ) позволяющая проводить исследования с использованием вычислительных средств. На этом этапе осуществляются отладка программ и проверка их корректности;

• *5 этап* – проведение расчетов. На этом этапе выявляются закономерности в поведении исследуемого объекта. Задачи, реализуемые при проведении расчетов, включают:

- планирование вычислительного эксперимента;

- проведение расчетов с использованием комплексов моделирующих программ;
- обработку результатов вычислений;
- *6 этап* – полученные результаты сравниваются с результатами натуральных экспериментов, и осуществляется калибровка моделей.

Все этапы вычислительного эксперимента, как и весь эксперимент в целом, проводятся итерационно с последовательным уточнением результатов. Преемственность предметной области позволяет использовать созданные ранее модели и их законченные компоненты для очередных шагов исследований с учетом накопления знаний как об исследуемом объекте, так и об особенностях вычислительной среды для проведения вычислительного эксперимента.

При постановке и выборе методов решения подобных задач важно знать полную группу ошибок, возникающих при оценке показателей эффективности реальных систем с помощью математических моделей. Исходя из физического смысла, ошибками моделирования являются:

- ошибки, возникающие при упрощении исходного моделирующего алгоритма;
- ошибки неточной дискретной реализации рабочего моделирующего алгоритма на средствах используемой вычислительной техники;
- ошибки, являющиеся результатом неточного задания исходных данных о параметрах модели и распределений входных данных;
- случайные ошибки, обусловленные ограниченностью статистики, которую получают при проведении натуральных экспериментов и статистических испытаниях на модели.

По характеру рассматриваемые ошибки чаще всего представляют собой сумму случайных и методических составляющих. Поскольку возможности аналитического их изучения ограничены, то комплексную оценку величины этих ошибок обычно дают в результате проверки статистической совместимости результатов моделирования и натуральных испытаний. Для сложных информационных систем решение этих задач получают с исполь-

зованием методов статистической проверки гипотез. При формировании гипотез исходят из конечных целей моделирования, состоящих в доказательстве тождественности законов распределения моделируемых и реальных значений выходных показателей системы. В зависимости от качества априорной информации и объема реальной статистики конкретные выражения для проверяемых гипотез будут получаться различными. Однако смысловое содержание этих исследований одно и то же и состоит в следующем:

а) если мера $\mu \{x_n^p, z_m^p\}$, выбранная для оценки близости законов распределения выборок x_n^p, z_m^p , полученных соответственно при натуральных испытаниях и моделировании, не превосходит некоторых критических значений Δ , то рассматриваемый вариант построения модели следует признать адекватным реальной системе;

б) если справедливо соотношение $\mu \{x_n^p, z_m^p\} > 0$, то необходимо провести анализ модели и установить наиболее вероятные причины, которые привели к отклонению гипотезы о статистической совместимости результатов моделирования и натуральных испытаний.

Под моделирующими программами информационных систем будем понимать программную реализацию агрегативной модели сложной системы. Будем считать, что комплекс моделирующих программ состоит из совокупности программных компонент и удовлетворяет требованиям, характеризующим моделирующий комплекс как сложную систему:

- моделирующая система может быть расчленена (не обязательно единственным образом) на конечное число частей, называемых подсистемами; каждая подсистема в свою очередь может быть расчленена на конечное число более мелких подсистем и т.д. до получения в результате конечного числа шагов – таких частей, называемых элементами сложной системы, относительно которых имеется договоренность, что в условиях данной задачи они не подлежат дальнейшему расчленению на части;

- элементы сложной системы функционируют не изолированно друг от друга, а во взаимодействии, при котором свойства одного зависят от условий, определяемых поведе-

нием других элементов;

- свойства сложной системы в целом определяются не только свойствами элементов, но и характером взаимодействия между элементами.

Рассмотрение моделирующих компонент в рамках приведенных ограничений позволяет опереться на математический аппарат системного анализа. Для того чтобы задать комплекс моделирующих программ как систему, необходимо и достаточно задать описание всех входящих в нее элементов и описать взаимодействие между элементами. Элементами описываемой системы являются программные моделирующие компоненты. Элементы в общем случае могут представлять задание агрегата (в агрегативной модели динамической системы) в виде его программной реализации в соответствующей вычислительной среде. Связи между компонентами должны удовлетворять требованиям, обусловленным информационными потоками в комплексе моделирующих программ, описывающих соответствующие состояния информационной системы.

С учетом перечисленных ограничений на класс рассматриваемых программ корректным будет следующее описание.

Пусть A – некоторое непустое множество; $\{j^n_1, j^n_2, j^n_3, \dots, j^n_s, \dots\}$ – система n -отношений на множестве A . С каждым n -отношением j^n на множестве A можно сопоставить n -местную логическую функцию (предикат) $P^n: A^n \rightarrow \{И, Л\}$ так, что $P^n(a_{i1}, a_{i2}, a_{i3}, \dots, a_{in}) = И$ тогда и только тогда, когда выполняется n -отношение $j^n(a_{i1}, a_{i2}, a_{i3}, \dots, a_{in})$, где И, Л – логические значения истины и лжи соответственно, $(a_{i1}, a_{i2}, a_{i3}, \dots, a_{in}) \in A$. Множество предикатов $P^n: A^n$ определяет условия активизации соответствующих компонент.

Такой метод описания моделирующих программ обоснования требований к техническим системам позволяет опереться на формальный аппарат алгоритмических алгебр В.М. Глушкова.

Отличие описываемой системы задания взаимодействующих компонент через функ-

ции предикатов на непустом множестве от традиционных методов описания моделирующих программ состоит в том, что:

- не накладываются ограничения на полное задание логической функции предиката (частичная определенность);

- существует возможность задания множества логических функций, адекватно описывающих отображение $P^n: A^n - \{И,Л\}$ (многоверсионное задание предиката).

Моделью будем называть систему, состоящую из множества A и определенной на данном множестве совокупности предикатов $\Pi = \{ p^n_s \mid s = 1, 2, \dots \}$.

Множество A будем называть основным множеством данной модели; предикаты, принадлежащие к Π , – её основными предикатами; последовательность $n_1, n_2, \dots, n_s, \dots$ будем называть типом модели, а $\Pi = \{ p^n_s \mid s = 1, 2, \dots \}$ – сигнатурой.

Модель $M_{A1} = \langle A1; \Pi1 \rangle$ (где $A1$ – непустое подмножество A ; $p1^n \in \Pi1$ – предикат на множестве $A1$, индуцированный предикатом $p^n \in \Pi$) будем называть подмоделью модели $M_A = \langle A; \Pi \rangle$.

Множество однотипных подмоделей образуют класс моделей по данному типу. Учет возможности проведения классификации подмоделей по данному типу (по допустимой последовательности применения различных отношений на A) позволяет определять границы применимости разрабатываемых методов создания и переиспользования компонентов моделирующих программ и ограничиваться рассмотрением комплексов моделей одного класса. В рамках однотипных классов моделей может быть рассмотрен вопрос об отношениях между моделями.

Однотипные модели $M_A = \langle A; \Pi \rangle$ и $M_B = \langle B; \Pi1 \rangle$ (где $\Pi = \{ p^n_s \mid s = 1, 2, \dots \}$, $\Pi1 = \{ q^n_s \mid s = 1, 2, \dots \}$) будем называть изоморфными, если существует взаимно однозначное отображение G множества A на множество B такое, что для любого $s = 1, 2, \dots$ предикат $p^n_s(a_1, a_2, a_3, \dots, a_{ns})$ справедлив тогда и только тогда, когда выпол-

няется предикат $q^n_s((a_1)G, (a_2)G, \dots, (a_{ns})G)$ для произвольных $a_1, a_2, a_3, \dots, a_{ns} \in A$. Отображение G в этом случае соответствует изоморфизмам $M_A = \langle A; \Pi \rangle$ и $M_B = \langle B; \Pi \rangle$.

Для моделирующих программ сложных информационных систем наличие изоморфных подмоделей соответствует режиму избыточности и возможности переиспользования соответствующих фрагментов. Построение модели на основе версий компонент сводится к формированию цепочек преобразований.

Построение функциональных отношений для взаимодействующих компонент моделирующих программ производится в процессе синтеза программы. Порождаемая программа может быть рассмотрена как расширяемое множество компонент, исходное множество которых определяется множеством начальных конструкций языка программирования. Цепочки операторов над исходным множеством элементов формируют расширяемое множество компонент с их частично определенными функциональными отображениями. Цепочки операторов можно рассматривать как компоненты программно-реализованной модели, допускающей их автономное вычисление, для вычислительных средств, в составе которых имеются разнородные процессоры и распределенная система коммутации [2].

Назначение компоненты программно-реализованной модели на соответствующий процессор (процессорное ядро) требует затрат определенных ресурсов на организацию вычислений. Для однотипных процессоров в рамках одной системы на кристалле [3] с подсистемой памяти типа NUMA применяются методы параллельной обработки данных, учитывающие конфликты в общей памяти. Использование вычислительных средств, включающих сигнальные и графические процессоры, требует более детального анализа трафика исполнения моделей.

2. Стенд отладки и оптимизации моделирующих программ

В ОАО «ИНЭУМ им. И.С. Брука» разработан стенд для отладки и оптимизации мо-

делирующих программ. С помощью стенда могут быть решены следующие основные задачи:

- управление вычислительным процессом за счет изменения назначения функциональных задач на соответствующие вычислители и микропроцессорные ядра с использованием механизмов операционных систем типа Unix и QNX. Правила, по которым осуществляется организация вычислительного процесса (назначения задачи на соответствующий процессор), формируются на основе ожидаемого времени реакции системы и загрузки отдельных процессоров;

- оценка реализуемости типовых программно-аппаратных решений при обеспечении управления вычислительным процессом в режиме реального времени.

Структура стенда приведена на рис. 2. В его состав входят компоненты:

- функциональной обработки;
- оценки состояния вычислительной среды;
- управления реконfigurацией вычислительной среды;
- визуализации и формирования отчетной документации.



Рис. 2

Структура и состав стенда отладки и оптимизации моделирующих программ

Стенд обеспечивает использование базовой технологии создания системы автоматизации проектирования программного обеспечения для решения задач моделирования сложными радиотехническими комплексами реального времени, в том числе:

- проверку количества универсальных процессоров в вычислительных комплексах (ВК) функциональных задач;
- проверку количества графических процессоров в ВК функциональных задач;
- проверку количества микропроцессорных ядер на процессоре в ВК функциональных задач;
- проверку частоты переключения режимов функциональной обработки;
- проверку объёма кэш-памяти 2-го уровня;
- проверку объёма оперативной памяти, доступной для системы;
- возможность системы программирования в части автоматического распараллеливания компилируемых программ для исполнения их в параллельном режиме на многоядерном микропроцессоре;
- выполнение модулей программ компонент функциональной обработки под управлением ОС Unix и QNX;
- выполнение модулей программ оценки состояния вычислительной среды под управлением ОС Windows;
- выполнение модулей управления реконфигурацией вычислительной среды;
- выполнение модулей визуализации и формирования отчетной документации.

Технология оптимизации расчетов опирается на возможности управления вычислительным процессом [4] и оценки характеристик вычислительных процессов, выполняемых на процессорах многоядерной архитектуры, и ориентирована на аппаратные платформы вычислительных средств, на которые могут быть установлены операционные системы типа Linux-intel, Sparc-Solaris и Sparc-Linux. Технология включает:

- управление вычислительными системами, созданными на базе многоядерных микропроцессоров за счет изменения приоритетов задач. Это осуществляется путем формирования нового конфигурационного файла для новой реализации трафика задачи реального времени;

- обеспечение устойчивой работы информационных систем при обработке нестационарных информационных потоков и реконфигурации вычислительных компонент на основе замеров загрузки микропроцессоров и времени реакции на изменение интенсивности потоков задач, а также на изменение конфигурационных файлов и приоритетов.

Технология использовалась в ходе обоснования облика и испытаний сложной информационной системы реального времени.

Заключение

В статье рассмотрены подходы к моделированию с целью обоснования облика и испытаний сложных информационных систем. Показана возможность использования результатов ранее накопленных знаний в виде многоверсионных моделирующих программ. Описан стенд отладки и оптимизации моделирующих программ. Полученные результаты могут быть использованы при оценке технических решений и испытаниях технических систем, применяющих в своем составе вычислительные средства ряда «Эльбрус».

Работа выполнена в ОАО «ИНЭУМ им. И.С. Брука» в рамках ОКР «Расчет-ВП» по заказу Минпромторга России.

Литература

1. Бусленко Н.П. Моделирование сложных систем. М., Наука, 1978.
2. Богданов А.Ю., Перекатов В.И., Фельдман В.М. Коммуникационные интерфейсы межмашинных связей вычислительных средств семейства «Эльбрус». – «Вопросы радиоэлектроники», сер. ЭВТ, 2013, вып. 3, с. 72-84.

3. Волконский В.Ю., Грабежной А.В., Муханов Л.Е., Нейман-заде М.И. Исследование влияния подсистемы памяти на производительность параллельных программ. – «Вопросы радиоэлектроники», сер. ЭВТ, 2013, вып. 3, с. 22-38.

4. Парамонов Н.Б. Испытания при замене вычислительных средств сложных технических систем. – «Вопросы радиоэлектроники», сер. ЭВТ, 2011, вып. 3, с. 161-172.