

Средства и методы распределенного и имитационного моделирования

А. П. Коротков

Финансовый университет при Правительстве Российской Федерации (Финуниверситет), Financial University
akorotkov@yandex.ru

Аннотация. Моделирование и анализ действий различных динамических систем во времени и пространстве являются наиболее важными сферами при изучении науки и техники. Для таких динамических систем характерна большая сложность вычислительных операций. Развитие информационных технологий и постоянно растущие требования в мире для этих систем, повлекли за собой создание истинно новых вычислительных структур, которые способны решать ещё более сложные и запутанные задачи.

Ключевые слова: имитация; анализ; система; моделирование; связи; измерения

При изучении различных систем человек сталкивается с проблемой их отображения, а также их применения в различных сферах. Объекты закрепляются разными терминами, графиками, чертежами, рисунками, с помощью математических формул, уравнений и многими другими способами. Итак, само отображение моделей называют моделированием. Модель представляет собой четкую систему, с помощью познания которой можно получить знание о совершенно другой системе или структуре.

Моделирование является очень трудоёмким процессом по ряду причин. Первой причиной служит «проектирование», которое невозможно без глубоких познаний и определенных навыков. Второй причиной можно выделить то, что сам процесс моделирования занимает очень большое количество времени, так как для сбора и обработки информации о модели потребуется приложить достаточные усилия, а затем и терпение для обработки информации, чтобы достичь соответствующих параметров. Третьей причиной является необходимость изучать огромное количество различных параметров модели, что требует значительных вычислительных ресурсов. Тем самым кажется возможным повысить скорость моделирования путем задействования большего числа вычислительных ресурсов, то есть, использования множества процессоров.

К основным понятиям имитационного моделирования можно отнести понятия реальной системы и модели. Реальная система подразумевает под собой совокупность взаимосвязанных элементов, которые принадлежат определённой части, являются объектом исследования. Таким образом, понятие системы является относительным. С одной стороны, эта совокупность элементов рассматривается как система, а с другой – как подсистема, то есть одна из составных частей системы. Моделью называется некое абстрактное описание системы, детали которой определяет исследователь, решение,

существенный это элемент или нет, принимает сам [исследователь]. Имитационная модель – логико-математическое описание системы, которое можно переделать как условие, решаемое на вычислительной технике, то есть, другими словами, это программа.

I. СИСТЕМЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Применение современной техники играет очень важную роль в исследовании свойств и изучении поведения моделей. Успешность таких исследования напрямую зависит от выбора языка для реализации данных алгоритмов модели. Именно поэтому большое значение играет правильный выбор языка моделирования или даже среды, которая включает в себя соответствующий язык. Главные преимущества языков имитационного моделирования в отличие от языков общего назначения заключаются в удобстве программирования при реализации модели систем.

Системы имитационного моделирования в отличие от универсальных языков программирования предоставляют ряд преимуществ:

- обеспечение исследователя естественной средой для создания имитационных моделей и дают такие возможности, как генерирование случайных чисел с распределением вероятности;
- продвижение модельного времени;
- обрабатывание списков текущих и предстоящих событий;
- имеют встроенные механизмы описания параллельных процессов.

Имитационные модели, которые созданы с помощью моделирования, намного проще усовершенствовать и в дальнейшем использовать.

Применение статистических методов, которые неизбежны при имитационном моделировании, нуждаются в больших затратах машинного времени и вычислительных средств. Одним из путей решения проблемы является использование суперкомпьютерных вычислительных машин, и, так называемых, «кластерных архитектур» (рис. 1).

И тут возникает вопрос об использовании параллельного или распределенного имитационного моделирования. Данные направления изучают следующим образом: использование симметричных мультипроцессоров (SMP) или массово параллельных

систем (МРР) определяется как параллельное моделирование, а исследования в локальных или глобальных вычислительных сетях чаще всего принимаются за распределенное моделирование.

Главными требованиями при выборе системы моделирования, учитывая её функционирование в данной вычислительной среде, являются её отказоустойчивость, синхронизация времени и возможность организации взаимодействия частей модели между собой в процессе моделирования. В параллельном моделировании основной вопрос – ускорение процесса решения задач. В распределенном моделировании преобладают проблемы взаимодействия между различными программно-аппаратными средствами, повторного использования библиотек и выполнения многовариантных расчетов.

Системы, которые готовы предложить такой функционал довольно трудно найти. Такие возможности, так или иначе, включают в себя системы семейства GPSS.

II. ПРИЧИНЫ ДЛЯ ПЕРЕХОДА К РАСПРЕДЕЛЕННОМУ МОДЕЛИРОВАНИЮ

Распределенное моделирование объясняется:

- способностью использовать вычислительные ресурсы несколькими процессорами (компьютерами) для осуществления имитационного эксперимента (подразумевается распределение компонентов «имитационной модели» по процессорам (компьютерам) и их совместное участие в имитационном эксперименте), преследуя цель повысить производительность системы «имитационного моделирования»;
- использованием локальной памяти процессоров (компьютеров);
- одновременным запуском нескольких повторений параллельно на нескольких компьютерах (временные затраты на эксперимент снижаются);
- объединением готовых имитационных моделей и их совместным участием в эксперименте;
- возможностью участия удаленных друг от друга географически пользователей в работе над имитационным проектом, доступность разработки модели, старта имитационного эксперимента и одновременно наблюдения за выполнением данной модели;
- усиление надежности выполнения имитационного эксперимента, так как при выходе из строя компьютера или процессора, на котором выполнялся один из экспериментов «имитационной модели», выполнение его можно продолжить на другом компьютере (процессоре).

Для выполнения распределенного моделирования необходимо определить протокол синхронизации, который служит для синхронизации модельного времени и передачи событий между логическими процессами. Передача сообщений гарантирует сохранение причинно-следственных связей между событиями в моделируемой системе. По принципу организации синхронизации

модельного времени между логическими процессами различают классический и оптимистический протокол распределенного моделирования. В контексте консервативного протокола моделирования можно выделить работы К. М. Chandy, J. Misra, D.A. Jefferson.

III. ДВА НАПРАВЛЕНИЯ В РАЗВИТИИ РАСПРЕДЕЛЕННЫХ СИСТЕМ МОДЕЛИРОВАНИЯ

Развитие распределенного имитационного моделирования разделяется на два направления. Первое – монолитные системы моделирования, второе – подготовленные системы моделирования, которые объединены особым, специально созданным программным обеспечением. Для разработки новых сверхэффективных «монолитных» систем моделирования, где возможна поддержка параллельного дискретно-событийного моделирования (PDES – Parallel Discrete Event Simulation) задействуются спецпроцессоры для параллельного моделирования, библиотеки, языки и различные инструменты. Яркими примерами могут служить следующие системы: TeD/GTW, SPEEDES, Task-Kit. В России также ведутся разработки распределенных систем имитации.

Имитационные модели вплотную контактируют со сферой, для которой были созданы, и это не может не создать проблемы при смене среды их применения, например, при использовании на другой платформе. Другим образом, который возник в распределенном моделировании – является совокупность различных систем моделирования. На этот раз составляющими частями «имитационного моделирования» выступают не объекты (в отличие от последовательного моделирования), не логические процессы (PDES), а сами имитационные модели. Задачей этих проектов является создание среды для взаимодействия разработанных ранее имитационных моделей. Данный подход характерен для следующих систем: DIS (Distributed Interactive Simulation Protocol), HLA (High Level Architecture или, другими словами, архитектура высокого уровня). На сегодняшний день самой распространенной системой является HLA. Составными частями имитационного моделирования в данном случае выступают федераты. Объединение федератов – федерация. Федераты одной федерации могут быть различными (например, тренажеры, программы для сбора данных и, так называемые, «живые участники»). С помощью программной оболочки RTI происходит обмен данными между федератами.

IV. РАСПРЕДЕЛЕННОЕ ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

На сегодня выработаны различные модели, методы и средства распределенного имитационного моделирования. В их число входят наиболее известные системы, такие как, Недис-Р, Диана, SLX, Мера. Но применение данных систем для моделирования распределенных вычислительных систем затруднено необходимостью представления и анализа вспомогательного уровня моделируемой системы, так называемой, концептуальной схемы её предметной области. Именно поэтому всё чаще и чаще возникает необходимость в коммуникационной среде, способной поддерживать высокоуровневые

системы имитационного моделирования и могла бы позволить организация обмена данными и синхронизировать время между участками имитационных моделей.

Существует множество довольно известных механизмов передачи данных, которые применяются при организации распределённых вычислений.

1. PVM (с английского Parallel Virtual Machine) – произвольно распространяемая библиотека передачи сообщений. Она позволяет использование узлов вычисления, которые находятся в сети как параллельные компьютеры с распределённой памятью, объединённые, другими словами, в виртуальную машину.

Достоинства PVM:

- группировка разнородных узлов вычисления и динамическое присоединения узлов во время выполнения;
- координирование виртуальной машины, которая использует горизонтальный кластер рабочих станций.

2. MPI (с английского Message Passing Interface) – низкоуровневая библиотека передачи данных. Разработана для высоких производительных параллельных вычислений.

По сравнению с PVM, MPI способна предоставить функции как библиотека, но она не представляет собой виртуальную машину.

Достоинства MPI:

- высокая эффективность и поддержка собственных типов данных;
- пластичность, наличие огромного числа функций передачи сообщений.

3. CORBA (с английского Common Object Request Broker Architecture) – технологичный стандарт, который проектируется и разрабатывается консорциумом OMG.

Задача CORBA: объединение вызова программ, которые написаны на различных языках, работают и на локальных, и на удалённых сетевых узлах.

Достоинства CORBA:

- поддержка различных языков, таких как C, C++, ADA, Java и SmallTalk;
- масштабируемость;
- межвзаимодействие процессов;
- отделение декларативных интерфейсов от их реализации;
- приведение типов данных.

4. RMI (с английского Remote Method Invocation) – система, которая первоначально была разработана Sunsoft, и основой являлась платформа Java Development Kit (JDK).

Задача RMI: уничтожение различий между вызовами локальный и удалённых операций.

Достоинства RMI: отсутствует необходимость в осуществлении специального протокола при взаимодействии клиента и сервера:

- Позволение разработчику абстракции на уровне простого вызова методов объектов и инкапсуляция многих частей обмена сообщений по сети;
- Оперирование суждениями передачи объектов по сравнению с передачей байтов, таких как, например, socket-сети.
- Однако, все вышеперечисленные технологии имеют один важный общий для всех недостаток:
- Отсутствуют функции синхронизации модельного времени между частями модели. Наличие данных функций является ключевым фактором, который необходим для организации распределённого моделирования.

V. УПРАВЛЕНИЕ ВРЕМЕНЕМ В РАСПРЕДЕЛЕННОМ МОДЕЛИРОВАНИИ

Управление временем в распределённом моделировании обеспечивает выполнение событий в правильном хронологическом порядке. На алгоритм синхронизации возлагается обязанность верно выполнять повторные имитационные действия. При повторяющемся моделировании пользователь должен быть уверен, что им будут получены такие же результаты, что и в первый раз, если изначальные данные останутся без изменений. Обеспечение следования событий в верном порядке и проблемы повторяющегося имитационного прогона для моделирования тренажеров не особо важно.

Имитационная модель представляет собой совокупность логических процессов, которые взаимодействуют между собой, передавая друг другу сообщения в пометкой времени. Логические процессы являются отражением физических процессов. События первых должны выполняться в хронологической последовательности. Данное требование называют ограничением локальной казуальности. Если при игнорировании события с одинаковыми временными метками и если процесс поддерживает ограничение локальной казуальности, то осуществление имитационной программы на параллельном компьютере даст абсолютно такие же результаты, что и выполнение на последовательном, где события исполняются сходно с их временными метками. Данное свойство позволяет удостовериться в том, что исполнение имитационного прогона повторяемо. Каждый логический процесс как последовательный симулятор поддерживает дискретное событийное моделирование, что означает, что каждый логический процесс включает в себя локальную информацию о состоянии объектов моделирования и содержит список событий, которые уже запланированы для данного процесса, но ещё не выполнены. Список не выполненных событий содержит локальные события и события, которые уже был запрограммированы другим процессом.

Работы симулятора состоит в том, чтоб отобрать из списка невыполненных событий такое, которое имеет минимальную временную метку для последующей

обработки. Процесс можно рассматривать как выполнение последовательности действий. Выполнению событий сопутствует изменение переменных, которыми определяется состояние моделируемого объекта. Логический процесс при выполнении очередного события способен запланировать выполнение нового события для себя или для других процессов.

Любой логический процесс имеет локальные часы. Они показывают время для выполнения самого последнего выполненного симулятором события. Время, на которое запланировано логическим процессом событие обязано быть больше, чем значение локальных часов.

VI. СОБЫТИЙНОЕ И ВРЕМЕННОЕ МОДЕЛИРОВАНИЕ

На сегодняшний день различают два вида процесса моделирования, которые различают по критерию увеличения модельного времени. В синхронной дискретной системе модельное время увеличивается на одинаковые промежутки, или, иначе, моделируемая динамическая система дискретизирована малыми промежутками времени. Выбор моделирования и промежутка времени моделирования соответствует закономерности: время одного такта должно хватать для запрашиваемой точности моделирования (чем ниже длительность одного такта, тем выше точность). Сокращение продолжительности такта повышает длительность процесса моделирования. В асинхронном моделировании явления беспорядочно разбросаны во временном промежутке, и концепции, управляемые временем, влекут за собой неэффективные алгоритмы моделирования. Событийное дискретное моделирование производит запуск системы моделирования в момент появления события. В последующем такой способ обозначается как дискретно-событийное моделирование (DES – discrete event simulation). В DES в момент исполнения последовательных повторных действий возникают события в модельном времени. Данные события лежат в упорядоченном во времени списке, в котором хранятся события, которые могут произойти в будущем. Ядро моделирования управляет процессом моделирования; постоянно извлекает начальное событие из списка, выполняет обработку действия посредством изменения состояния переменных. В процессе моделирования предполагается уничтожение устаревших событий. Моделирование выполняется до того, пока имеются запланированные события и не будет достигнуто время завершения моделирования.

При моделировании дискретно-событийной системы возможно возникновение как параллельных, так и взаимоисключающих событий. Главной отличительной чертой параллельных событий служит то, что их появление никак не связано и возможно выполнение обработки этих сообщений параллельно. А с другой стороны, для улучшения последовательного моделирования могут быть удалены. При обработке взаимоисключающих событий моделирование одного приводит к удалению другого.

VII. ОРГАНИЗАЦИЯ РАСПРЕДЕЛЁННОГО МОДЕЛИРОВАНИЯ

Общая схема процесса построения имитационной модели выглядит следующим образом: исполнительную

систему можно представить в виде двухуровневой системы управления вычислительным кластером. Верхний уровень функционирует с помощью управляющей программной надстройки, которая транслирует запросы от генератора распределённой модели в формат системы пакетной обработки (СПО). На нижнем уровне происходит управление запуском заданий модулями системы пакетной обработки (СПО). Взаимодействие осуществляется с кластером вычисления с помощью Web-интерфейса и модулем трансляции в систему пакетной обработки. Для гарантии отказоустойчивости процесса распределенного моделирования был разработан модуль создания КТ. Он предназначен для инициирования частей GPSS распределенной модели и старта системы имитационного моделирования. Также модуль отслеживает порядок исполнения сегментов модели и организывает необходимые КТ в полном соответствии с установками пользователя. Сформированные КТ позволяют отслеживать промежуточные отчеты о состоянии продленных действий всего процесса моделирования в интервале между началом и концом процесса. Коммуникационная библиотека обеспечивает взаимодействие частей распределенной GPSS-модели. Библиотека основана на клиент-серверной архитектуре и включает в себя коммуникационные сервера, Active-X-сервера и dll-библиотеки. Коммуникационный сервер можно представить как многопоточное асинхронное NET-приложение. Dll-библиотека служит для связи имитационного моделирования с ActiveX сервером.

VIII. ЗАКЛЮЧЕНИЕ

Проблема проведения распределенного имитационного моделирования с электронным элементом на сегодняшний день остается актуальной в связи со сложностью и огромной стоимостью выполнения исследований. Для проверки электронного элемента используется среда моделирования, которая способна воссоздать его работу по описанию на языке проектирования аппаратуры. Весь процесс моделирования электронного элемента допустимо рассматривать как процесс моделирования дискретно-событийной системы. Организация распределенного моделирования основывается на использовании созданных коммуникационных библиотек, которые способны обеспечить синхронизацию времени и обмен данными между частями распределенной модели.

СПИСОК ЛИТЕРАТУРЫ

- [1] Аврамчук Е.Ф., Вавилов А.А., Емельянов С.В. Теория системного моделирования. М.: Машиностроение; Берлин: Техник, 1998. С. 299-303.
- [2] Лычкина Н.Н. Системы принятия решений в задачах социально-экономического развития регионов // Компьютер. 1999. № 2(32). С. 11-18.
- [3] Звягин Л.С. Применение системно-аналитических методов в области экспертного прогнозирования// Экономика и управление: проблемы, решения. 2017. Т. 3. № 6. С. 145-148.
- [4] Звягин Л.С. Ключевые аспекты современного партнерства организаций бизнеса и сферы образования// Планирование и обеспечение подготовки кадров для промышленно-экономического комплекса региона. 2017. Т. 1. С. 126-129.
- [5] Мохов А.И. Отличие в подходе системотехники и комплексотехники к созданию технических систем // Электротехнические и информационные комплексы и системы. 2011. № 1. Т. 7. С. 41-44.