

# Современная практика использования распределенного моделирования и управления временем

Т. Ю. Дьякова

Финансовый университет при Правительстве Российской Федерации (Финуниверситет), Financial University  
tanyadyakova@inbox.ru

**Аннотация.** Прогресс в области информационных технологий и рост требований, исходящих от научно – исследовательских приложений, объясняют формирование новейших вычислительных инфраструктур, направленных на решение ряда различных задач, которые не предполагают тесную взаимосвязь между процессами, происходящими параллельно. Для эффективного применения высокопроизводительной вычислительной техники необходимо привлечение высококвалифицированных кадров, которые могут эти средства грамотно применять для решения сложных задач.

**Ключевые слова:** *вероятность; подход; информация; моделирование; оптимизация*

Исследование систем, состояние которых зачастую зависит от случайных событий, взаимосвязей между компонентами системы и воздействия внешней среды. В таком случае исследование системы аналитическим методом представляется крайне сложным. По этой причине исследователи прибегают к методам имитационного моделирования. Использование методов статистики требует большого количества машинного времени и вычислительных ресурсов. Чтобы устранить эту сложность, можно использовать суперкомпьютеры, в том числе и кластерные архитектуры. Зачастую это означает применение методов распределенного имитационного моделирования, которое подразумевает процесс осуществления исследований в рамках локальных и глобальных вычислительных сетей. В распределенном моделировании особенно важными являются взаимосвязи и взаимодействия между различными программными средствами и аппаратами, выполнение сложных вычислительных процессов.

В данной работе рассматриваются понятия распределенного моделирования, его методы, преимущества и недостатки, обозначены особенности функционирования и реализации распределенных систем, а также описаны технологии, используемые в процессе распределенного имитационного моделирования.

## I. ПОНЯТИЕ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ

Термин система распределенного моделирования означает совокупность независимых компьютеров. Пользователи рассматривают их как целостную

объединенную систему. Из этого определения вытекают две особенности распределенной системы:

- Все машины имеют автономность;
- Сложность и гетерогенность аппарата, на котором основывается распределенная система является скрытой.

Процесс организации распределенных систем аккумулирует в себе добавочный уровень программного обеспечения, располагающийся между двумя уровнями – верхним (пользователи и набор их приложений) и нижним (совокупность операционных систем). Именно поэтому такое программное обеспечение зовут промежуточным.

Ряд исследователей определяют распределенную систему как совокупность автономных компьютеров, процессоров, находящихся во взаимосвязи. Компьютеры и процессоры являются узлами этой распределенной системы. Обладая автономностью, узлы должны иметь свой собственный блок управления. По этой причине параллельный компьютер, обладающий лишь одним потоком управления и не одним потоком данных, нельзя называть распределенной системой. Так как узлам приписывается такое свойство, как взаимосвязь между друг другом, эти узлы должны обладать способностью обмена информацией. Процессы также могут являться узлами системы, поэтому ее понятие может трактоваться как набор взаимосвязанных процессов, находящихся в постоянном взаимодействии.

Таким образом, под распределенной системой можно понимать сети локального и глобального характера, вычислительные системы с множеством процессоров и процессы, находящиеся во взаимодействии и выполняющиеся на одном компьютере. Любой вид распределенной системы имеет свои собственные черты, которые необходимо учитывать в процессе разработки программного обеспечения. К примеру, глобальные и локальные сети различаются в таких аспектах, как надежность, время коммуникации, однородность вычислительных узлов.

- Параметры надежности – это уровень надежности процесса передачи данных в рамках компьютерной сети. В глобальной сети существует высокая вероятность утраты сообщения. Однако

определенные алгоритмы должны уметь предотвращать или компенсировать такие информационные потери. Если говорить о локальных сетях, то они обладают большей надежностью, и их алгоритмы в основном рассчитаны на абсолютную надежность передачи данных.

- Время, необходимое для передачи сообщения. В рамках глобальной сети тратится большее количество времени для передачи данных, чем в локальной сети. Временем обработки передаваемых данных практически всегда есть возможность пренебречь, если сравнивать его со временем, затрачиваемым на передачу.
- Однородность. Несмотря на тот факт, что вычислительные узлы локальных сетей порой имеют отличия в степени своей производительности, представляется возможным использование единого программного обеспечения и единых протоколов. В рамках же глобальных сетей задействуются протоколы и программное обеспечение различных видов. По этой причине возникает необходимость в учете совместимости видов программного обеспечения и протоколов в процессе преобразования данных.
- Безопасность. Вычислительные узлы являются довольно подверженными внешним факторам, в том числе и негативным. Именно поэтому в глобальных сетях нужно разрабатывать различные средства защиты от атак подобного вида.

## II. РАСПРЕДЕЛЕННЫЕ И ПАРАЛЛЕЛЬНЫЕ СИСТЕМЫ

Выше говорится о том, что процесс распределения относится к ключевым современным тенденциям. Она отражается на базах данных, искусственном интеллекте, операционных системах. Становление распределенных систем имитационного моделирования является одним из проявлений ранее упомянутой тенденции.

Еще одним ее проявлением является появление параллельных вычислительных систем. Их зачастую отождествляют с понятием распределенных вычислительных систем. Однако между ними существует ряд различий, который нужно обозначить во избежание неопределенности.

Термин «параллельные системы» обычно подразумевает суперкомпьютеры, как бы подчеркивая задействование многопроцессорной архитектуры.

К самым распространенным видами архитектур параллельных компьютеров относятся:

- MPP – параллельные системы массового характера;
- SMP – мультипроцессорные системы симметричного характера;
- NUMA – системы, имеющие неоднородный доступ к памяти;

- PVP – векторные системы параллельного характера;
- Кластеры – бюджетная альтернатива MPP.

В рамках параллельных систем вычислительными узлами являются серверы, рабочие станции, а также персональные компьютеры. Процесс кластеризации проходит на различных уровнях системы компьютера, таких как аппаратное обеспечение, операционные системы, программы – утилиты, системы управления и приложения.

## III. РАСПАРАЛЛЕЛИВАНИЕ ВЫЧИСЛЕНИЙ В РАСПРЕДЕЛЁННЫХ СИСТЕМАХ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Процесс распараллеливания вычислений является одним из отличительных особенностей функционирования любой распределенной системы. Каким же образом протекает распределения вычислений между узлами? Как возможно добиться сокращения времени, затрачиваемого на осуществление имитационных экспериментов?

Некоторые исследователи говорят о том, что наравне с процессом экспериментирования следует проводить специальные функции, такие как генерация случайных чисел, управление списком событий, которые произойдут в будущем времени, сбор статистических данных. Однако экономия времени при этом является не слишком значительной.

Декомпозиция модели, находящейся в иерархичном порядке, осуществляется путем разложения этой модели на ряд подсобытий. Их стоит выполнять параллельным путем. На степень экономии времени при процессе разложения модели влияет вид и сущность модели. К примеру, для таких моделей, как СМО (система массового обслуживания) процесс декомпозиции является довольно несложным. Любая очередь такой модели может быть выполнена на отдельном компьютере. Для многих других моделях разложение протекает сложно.

## IV. УПРАВЛЕНИЕ ВРЕМЕНЕМ В РАСПРЕДЕЛЕННЫХ ИМИТАЦИОННЫХ СИСТЕМАХ

Огромное значение для имитационного моделирования имеет фактор времени, ведь имитационное моделирование, в том числе и распределенное, направлено на изучение динамической системы путем замены реальной системы, меняющей свое состояние с течением времени, на модель. На общее состояние системы непосредственно влияет состояние ее компонентов, каждый из которых имеет свой собственный ряд особенностей. Однако следует понимать, какой смысл имеет время в моделировании. Обычно выделяют три вида времени: физическое, модельное и процессорное. Рассмотрим их подробнее:

1. Физическое время ( $T_p$ ) – это время, используемое в реальной системе, которая является объектом моделирования. К примеру, создается модель рабочего дня фирмы с 9:00 до 18:00;
2. Модельное время ( $T_s$ ) – отражение физического времени в модели. То есть работу предприятия

можно представить, как отрезок времени [9.00, 18.00], за единицу модельного времени (h) можно взять любое количество времени (минута, полчаса, час и так далее). Таким образом,  $T_s = T_s/h$ ;

3. Процессорное время ( $T_w$ ) – время работы симулятора на процессоре или компьютере. Например, процесс создания модели предприятия может потребовать час работы на компьютере. Порой (при использовании тренажеров) продвижение модельного времени необходимо согласовать с временем процессорным. Это называют моделированием в реальном времени. То есть если человек использует тренажеры, он попадает в виртуальную обстановку, которая должна быть максимально похожей на реальную. С другой стороны, модельное время идет быстрее, чем процессорное, т.е. процесс моделирования должен проходить как можно скорее. К примеру, работа определенного реального процесса длится несколько дней, за единицу модельного времени берется 1 час, а моделирование на компьютере осуществляется на протяжении 30 минут.

Таким образом, одной из главных задач имитационной системы является продвижение модельного времени. Программа, которая управляет процессом и осуществляет действия в последовательном моделировании, называется симулятором. В рамках распределенного моделирования данный процесс называется алгоритмом синхронизации. Этот алгоритм проводит синхронизацию между элементами моделирования. Этими элементами обычно являются логические процессы (LP). Логические процессы (LP – logical processes) отражают поведенческие характеристики физических процессов (PP – physical processes).

В процессе реализации распределенных систем пользователь чаще всего прибегает к двум технологиям: «точка – точка» (peer-to-peer) и «клиент – сервер» (client-server). Первая технология применяется на различных вычислительных узлах, которые могут быть объединены глобальной сетью. Клиент – сервер используется в процессе реализации имитационной системы на некотором числе процессоров, которые объединены локальной сетью. Сетевые пользователи имеют удаленный доступ к серверу, в рамках которого происходит моделирование. В основном, это используется в распределенном моделировании, применяемом в играх.

## V. МЕТОДЫ РАСПРЕДЕЛЕННОГО МОДЕЛИРОВАНИЯ, МЕХАНИЗМЫ ПЕРЕДАЧИ ДАННЫХ

На текущий момент времени существуют различные методы и средства распределенного моделирования. А также такие известные системы, как Недис – Р, Мера, Диана, SLX и другие. К сожалению, их использование для моделирования распределенных систем усложняется, так как существует необходимость представления и изучения добавочного уровня описания системы, являющейся объектом моделирования, - модели ее предметной области. По этой причине есть потребность в том, чтобы

имитационные системы, обмен данными и синхронизация времени между частями имитационной модели происходили в определенной коммуникационной среде. Существует целый ряд механизмов и методов организации вычислений распределенного характера:

1. PVM (Parallel Virtual Machine) – это библиотека для обмена сообщениями и данными, находящаяся в открытом доступе. Она создает единую виртуальную машину, которая аккумулирует в себе вычислительные узлы системы.

К достоинствам PVM можно отнести формирование такой виртуальной машины, которая задействует кластер рабочих станций; интеграция вычислительных узлов гетерогенного характера и динамическое присоединение узлов в процессе выполнения.

2. MPI (Message Passing Interface) – это библиотека низкого уровня, созданная для обмена данными высокопроизводительных вычислений, осуществляющихся параллельно. Если сравнивать ее с Parallel Virtual Machine, то Message Passing Interface не является виртуальной машиной, а лишь содержит в себе свойства библиотеки.

Преимуществами MPI являются: гибкость; огромное количество методов, направленных на передачу сообщений; высокий уровень производительности; поддержание данных специфического характера.

3. COBRA (Common Object Request Broker Architecture) – это технологический стандарт. Он был спроектирован консорциумом OMG. Целью Common Object Request Broker Architecture является интеграция программ, которые были написаны на различных языках программирования и функционируют в рамках и локальных, и глобальных сетей.

К достоинствам COBRA относят: обеспечение взаимодействия процессов и явлений; разграничение между интерфейсами декларативного характера и процессом их реализации; поддержка различных языков (C, C++, ADA, SmallTalk, Java); приведение данных, возможность масштабирования.

4. RMI (Remote Method Invocation). Данная система была создана Sunsoft и базировалась на JDK (Java Development Kit). Основной задачей RMI является устранение разницы между вызовами локальных и удаленных процедур. RMI имеет такие достоинства, как создание абстракции путем тривиального вызова методологии объектов; изоляция элементов передачи данных через сеть; использование понятий передачи объектов, а не передачи байтов (к примеру, в socket сетях); упрощенность процесса взаимодействия клиента с сервером за счет того, что нет потребности в реализации специализированного протокола.

Однако все методы, указанные выше, имеют главное несовершенство – они не имеют функций, необходимых для процесса синхронизации модельного времени между компонентами модели. А отсутствие таких функций считается определенным ограничением для процесса организации распределенного моделирования.

Процесс развития распределенного моделирования происходит согласно двум следующим направлениям:

1. Параллельное дискретное событийно – ориентированное моделирование PDES (Parallel Discrete Event Simulation). Согласно этому подходу необходимо задействование специальных процессоров, поддерживаемых определенными языками, библиотек и инструментов с целью проектирования новейших систем моделирования высокой эффективности (к примеру, TeD/GTW, SPEEDES, Task – Kit и другие). На территории России также проводятся работы с целью организации таких распределенных имитационных систем, как Мера (Новосибирск), Диана (МГУ) и других. Имитационные модели в большой степени зависят от среды, для которой они были созданы. Это объясняет трудности, возникающие после смены среды (к примеру, перенос системы с одной платформы на другую).

2. Объединение разнородных систем моделирования. При этом подходе сегментами имитационного моделирования являются имитационные модели, а не объекты (как в последовательном моделировании) или логические процессы (параллельное дискретное событийно – ориентированное моделирование). Следовательно, целью данных проектов является формирование среды для взаимодействия уже существующих имитационных моделей. Это применяется в таких системах, как DIS (Distributed Interactive Simulation), ALSP (Aggregate Level Simulation Protocol), HLA (High Level Architecture или «Архитектура высокого уровня»). На данный момент очень активно используется High Level Education. Составляющими имитационного моделирования здесь являются федераты или их объединения (федерации). Федераты, находящиеся в одной федерации, могут различаться по своей сути. Передача сообщений от одного федерата другому и исполнение федератов в одном времени осуществляется в рамках оболочки программы RTL.

## VI. ТЕХНОЛОГИЯ HIGH LEVEL ARCHITECTURE

Иным подходом к созданию имитационных распределенных систем моделирования является применение HLA (Архитектуры высокого уровня). Она была спроектирована службой имитационного моделирования DSMO (Defense Modeling and Simulation Office – Оборотный отдел моделирования и имитации) Министерства обороны США. К ключевым составляющим HLA относят процесс реализации RTI (Run Time Infrastructure – Инфраструктура времени выполнения). Она аккумулирует в себе серверы, клиентские части и библиотеку классов JavaBinding, созданную с целью реализации взаимодействия моделей, написанными на языке программирования Java. HLA обладает несколькими достоинствами, которых лишены технологии, указанные ранее:

- она поддерживает несколько механизмов реализации времени;
- имеет универсальную специфику, используется и в параллельном, и в распределенном моделировании;

Из Российских систем, поддерживающих стандарты HLA, можно привести в пример Меру и AnyLogic.

Среди недочетов High Level Architecture выделяют:

- сложность стандарта;
- существование ограниченности в пределе масштабируемости у распределенных систем, основанных на HLA;
- необходимость в создании добавочных средств, обеспечивающих сохранение работоспособности моделирования;
- невозможность взаимодействия моделей при применении разных версий RTI.

Опираясь на анализ, описанный выше, стоит обратить внимание на тот факт, что с целью взаимодействия распределенных частей стоит рассматривать различные технологии, такие как CORBA, RMI, коммуникационные библиотеки MPI, PVM и HLA. Однако данные технологии не способны решить все существующие проблемы проведения распределенных вычислений, не всегда несут в себе простую реализацию, отладку, использование модели и пока что не могут полностью предоставить все условия для создания распределенной среды имитационного моделирования.

Таким образом, в связи с такой относительно новой возможностью технологий, как объединение ряда компьютеров в единую сеть или процессоров в распределенную вычислительную среду, возник такой вид имитационного моделирования, как распределенное моделирование. В настоящее время распределенная обработка применяется во многих областях, к примеру, в процессе оптимизации и проектирования баз данных. В целом, распределенное моделирование является прогрессивным видом имитационного моделирования, предлагающим довольно эффективные решения сложных проблем, которые возникают в рамках реальных систем и процессов, однако имеющим некоторые недостатки в алгоритме своей реализации и в программных средствах, отвечающих за этот процесс реализации.

## СПИСОК ЛИТЕРАТУРЫ

- [1] Замятина Е.В., Осмехин К.А. Принципы проектирования алгоритма динамической балансировки в распределенной системе имитационного моделирования. Рецензируемый научно-технический журнал «Известия белорусской инженерной академии», 2005. №1(19)/2 2005.
- [2] Походун А.И. Экспериментальные методы исследований. Погрешности и неопределенности измерений: Учебное пособие / А.И. Походун. СПб: СПбГУ ИТМО, 2006. 112 с.
- [3] Ефимова Н.Ю. Оценка неопределенности в измерениях: практическое пособие / Н.Ю. Ефимова. Мн.: БелГИМ, 2003. 50 с.
- [4] Zvyagin, L.S. Problems of implementing system analysis in the target management. Proceedings of 2017 IEEE 2nd International Conference on Control in Technical Systems, CTS 2017. St. Petersburg; Russian Federation; 2017, pp. 267-272. DOI: 10.1109/CTSUS.2017.8109549.
- [5] Zvyagin, L.S. System analysis in optimization and decision making. Proceedings of 2017 IEEE 2nd International Conference on Control in Technical Systems, CTS 2017. St. Petersburg; Russian Federation; 2017, pp. 161-164. DOI: 10.1109/CTSUS.2017.8109515.