

МЕТОД ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ИСПОЛНЕНИЯ МОДЕЛЕЙ С МАССИРОВАННЫМ ПОТОКОМ СОБЫТИЙ

К. В. Ткачёв

Институт Вычислительной математики и математической геофизики СО РАН, 630090, Новосибирск

УДК 519.876.5

На основе экспериментальных данных было выявлено, что при использовании различных алгоритмов управления можно существенно сократить время выполнения модели и количество затрачиваемой памяти, увеличить отзывчивость модели для управляющих событий. Используя новый предложенный вариант «календаря событий», можно значительно сократить время, необходимое для проверки модели на работоспособность. Использование агента во время исполнения, позволит нам наблюдать, как меняется сама модель и её поведение. **Ключевые слова:** Имитационное моделирование, исполнение больших программ, агент, календарь событий, процессорное время, наблюдаемые параметры.

Введение

Развитие имитационного моделирования как метода научного исследования применяется, с середины прошлого века. Современные информационные технологии сделали возможным моделирование больших систем, состоящих из многих компонентов, реагирующих на множество различных событий. Рассмотрим одно из направлений развития распределенного имитационного моделирования. Это цельные системы моделирования [4].

Для создания новых высокоэффективных «цельных» систем моделирования, в которых поддерживается параллельное дискретно-событийное моделирование (PDES-Parallel Discrete Event Simulation) используют сопутствующие языки, библиотеки и инструментальные средства. Примерами этого подхода служат такие системы как TeD/GTW, SPEEDES, Task-Kit и др. [1]. В России тоже ведутся работы над подобными системами (Мера (Новосибирск), Диана (МГУ)).

Для синхронизации событий в имитационной модели требуется наличие календаря событий, другими словами, упорядоченного по неубыванию списка уведомлений о будущих событиях.

Календарь событий один из самых главных элементов имитационной модели, предназначенный для управления процессом появления событий в системе с целью обеспечения необходимой причинно-следственной связи между ними. Основные функции календаря событий [2]:

- Ранжирование по времени запланированных событий, т.е. составление упорядоченной временной последовательности с учётом вида события и обработчика, в котором оно будет обрабатываться.
- Запуск обработчика в момент наступления соответствующего события.
- Хранение и передача информации о обработчиках в нужные моменты времени.

При большой размерности модели возникает проблема увеличения длины этого списка и, соответственно, увеличению затрат на его обработку. В этом случае затраты начинают преобладать над затратами на отображение поведения системы [3].

Рассмотрим два способа сокращения времени моделирования.

Работа выполнена при частичной финансовой поддержке проекта Российского фонда фундаментальных исследований (код проекта № 17-07-00775 А) и проекта программы фундаментальных исследований президиума РАН.

1 Методы

В этой статье будут предложены несколько вариантов оптимизации. Первым вариантом является разделение календаря на две части, на отвечающую за «важные» события и за остальные события. Основываясь на предыдущих исследованиях, разделим события на влияющие на логику поведения модели и не оказывающие особого влияния. Информацию о важных событиях необходимо хранить и восстанавливать, а события, поток которых носит чисто случайных характер, можно не хранить, а просто генерировать датчиком случайных чисел [5].

Основным плюсом нового варианта является то, что при исполнении только первой части календаря с большой долей вероятности можем утверждать, что модель работоспособна и правильно функционирует. При этом, степень игнорирования определяется экспериментально для каждой конкретной модели и зависит от интенсивности потоков событий различных классов. При использовании «нового» календаря событий существует вероятность выйти за пределы допустимой погрешности. Из этого и вытекает необходимость определять степень игнорирования для каждой конкретной модели. Ещё одним плюсом является то, что он позволяет частично избежать проблемы синхронизации при распараллеливании, так как необходимо перестраивать только календарь с важными событиями.

Для подтверждения этой идеи была создана модель, позволяющая тестировать различные алгоритмы управления массированным потоком событий в имитационной модели и анализировать её устойчивости по отношению к потере некоторых событий при сбоях в работе оборудования или системы управления процессами.

Рассмотрим немного изменённую модель. Она представляет собой имитационную модель проведения ремонтных работ в электрических сетях. Событие представляет собой ремонт определенного объекта. Общие характеристики модели:

- Количество объектов ремонта не менее 1000.
- Ремонт требует затрат условных ресурсов.
- Поломка случается через случайное время.
- В наличии 10 ремонтных бригад.
- Некоторые поломки связаны между собой.
- Есть рейтинг важности объектов.
- Есть коэффициент, отвечающий за время (время года, день месяца).

В рассматриваемой модели выбор «важных» событий представлял собой алгоритм анализа системы, зависящий от связанности объектов между собой, от изначального значения важности и от коэффициента времени. В определённые моменты, у некоторых событий меняется статус важности [3].

Первоначально важными событиями были ремонтные работы в муниципальных учреждениях. При дальнейшем исследовании было установлено, что необходимо расширить границу важности, дополнив её объектами, связанными между собой.

Далее, в таблице 1 представлено сравнение старого вида календаря с новым (по процессорному времени, измеряется в секундах). В первом столбце приведён процент количества важных событий от всех. В данном примере отсутствуют связанные события.

В первом столбце таблицы 2 приведён процент количества цепочек от всех троек событий (разбиваются все события на тройки, один раз). Во втором и третьем столбцах указаны процентные соотношения времени и количества ресурсов при запусках с цепочками событий и без. По таблице 2 можно заметить зависимость процессорного времени от количества связанных событий (для нового вида календаря). Проблемой данного примера является то, что связанные события могут находиться в разных частях календаря. В этом случае мы встречаем проблемы связанных событий. Увеличение количества таких событий ведет к значительному росту количества запросов необходимых для синхронизации первой части календаря с другой. Решением данной проблемы является объединение таких событий в новый класс событий, входящий в важные события. Начальные данные: $N=32768$ (объекты), $P=16777216$ (поломки), $W=838860$ (количество «важных» событий). Все значения в таблице были получены экспериментально.

Таблица 1: Сравнение по процессорному времени

Важные события	Старый вариант	Новый вариант
25%	2568.12(сек)	623.78(сек)
75%	2689.26(сек)	1991.16(сек)

Таблица 2: Связанные события

Кол-во цепочек событий	Время моделирования	Затраченные ресурсы
5%	108%	113%
10%	117%	122%
25%	145%	153%

Следующим вариантом оптимизации является создание агента наблюдателя, который будет следить и изменять некоторые параметры. Осуществлять изменения он должен во время выполнения модели (динамически). Назовем их наблюдаемыми и управляемыми параметрами. На рис. 1 изображена схема взаимодействия агента с моделью.



Рис. 1: Схема взаимодействия агента

В рассмотренной модели наблюдаемые параметры:

- Сколько ресурсов и времени необходимо бригаде на данном ремонте.
- Сколько не работает в данный момент объектов.
- Связан ли данный ремонт с другим.
- Время года и день месяца.
- Сколько важных объектов не работает.

Управляемые параметры:

- Граница важности объектов.
- Общая стоимость ремонтных работ за определённый период.

- Допустимое количество неработающих объектов.
- Ограничения времени ремонта.

В таблице 3 указано время моделирования с использованием агента, изменяющего параметры во время выполнения модели с различными ограничениями.

Таблица 3: Условия изменения

Условия изменения	Без агента	С агентом
Изменение времени года	3564,64(сек)	3258,15(сек)
Ограничения ресурсов	3489,87(сек)	2499,69(сек)
Минимизация ущерба	3698,15(сек)	2954,99(сек)

Заключение

Эксперименты показали, что при использовании различных алгоритмов управления можно существенно сократить время выполнения модели, уменьшить количество затрачиваемой памяти и увеличить отзывчивость модели для управляющих событий. Хотя модель предназначена для отладки алгоритмов управления процессом имитации, она достаточно правдоподобна и может использоваться для принятия решений в работе ремонтных бригад.

Благодаря использованию нового варианта «календаря событий», можно значительно сократить время, необходимое для проверки модели на работоспособность. Степень игнорирования определяется экспериментально для каждой конкретной модели и зависит от интенсивности потоков событий различных классов.

Использование агента во время исполнения, позволяет нам наблюдать, как меняется сама модель и её поведение. Изменении некоторых параметров ведет к существенному сокращению времени моделирования.

Список литературы

- [1] Fujimoto R. M. Parallel Discrete Event Simulation, Communications of the ACM, Vol. 33, No. 10, 1990 С. 30–53.
- [2] Окольнішников В. В. Представление времени в имитационном моделировании //Вычислительные технологии. Сибирское отделение РАН. 2005. № 5. С. 57–80.
- [3] Ткачёв К.В. Экспериментальное исследование метода управления массированного потока событий // Материалы 53-й международной научной студенческой конференции. Новосибирский гос. ун-т: Новосибирск, 2015.
- [4] Шеннон Р. Имитационное моделирование систем искусство и наука. М.: Мир, 1978.
- [5] Korniss, G., Toroczkai, Z., Novotny, M.A., Rikvold, P.A., From Massively Parallel Algorithms and Fluctuating Time Horizons to Nonequilibrium Surface Growth // Physical Review Letters. 84. 2000. P. 1351–1354.

*Кирилл Валерьевич Ткачёв — инженер Институт Вычислительной математики и математической геофизики СО РАН;
e-mail: tkachev@sscc.ru, tkachyovkv@yandex.ru
Дата поступления — 31 мая 2017 г.*