

**ВЗАИМОДЕЙСТВИЕ АГЕНТОВ В РАСПРЕДЕЛЕННОЙ ДИСКРЕТНО-СОБЫТИЙНОЙ СИСТЕМЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ
OpenGPSS**

Д. Г. Диденко (Киев)

Метод имитационного моделирования приобретает все большую популярность при проектировании различных систем. Однако, несмотря на рост производительности современных компьютеров, их мощности не хватает для моделирования задач, связанных с самолетостроением, автомобилестроением, логистикой, сборочным производством и т. п., когда имитационные прогоны моделей могут длиться часами. Одним из вариантов решения этой проблемы является использование параллельного и распределенного дискретно-событийного имитационного моделирования.

Существует много реализаций распределенных систем имитационного моделирования, например, SPEEDES [1] и PARASOL [2], и даже есть стандарт в этой области – HLA (High Level Architecture). Однако многие разработчики имитационных моделей отмечают сложность реализации этого стандарта, а также невозможность его использования для языков декларативного типа, таких, как GPSS [3] из-за объектно-ориентированной направленности HLA. В то же время язык GPSS широко используется в мире, недаром одна из последних его версий названа GPSS World [4]. Поэтому актуальной является проблема создания распределенной системы, с автоматическим распределением GPSS-программ.

Предлагается создание распределенной системы имитационного моделирования OpenGPSS [5] с использованием технологии взаимодействующих агентов и многосессионной клиент-серверной СУБД Oracle, позволяющей одновременно работать нескольким пользователям (сессиям). В такой системе пользователь сможет задавать задание системе моделирования через WEB-интерфейс, а потом получать результат – подобно тому, как это реализовано в системе WebGPSS [6].

Архитектурные особенности системы OpenGPSS

Предлагается создание системы OpenGPSS с использованием кластера моделирования, состоящего из серверов, предназначенных для хранения моделей, сущностей и проведения распределенных имитационных экспериментов.

Дадим некоторые определения, необходимые для описания архитектуры системы OpenGPSS. Под *сущностями* будем понимать множество объектов (блоков) языка GPSS [3]: одноканальные устройства (Facility), многоканальные устройства (Storage), очереди (Queue), функции (Function), логические переключатели (Logic Switch), матрицы (Matrix), сохраняемые величины (Save Value), таблицы (Table), списки пользователя (User Chain), а также, дополнительно, генераторы (Generate).

Понятие *сообщение* в системе OpenGPSS аналогично понятию транзакта в языке GPSS и используется для изменения состояния сущностей.

Под *агентом* в системе OpenGPSS будем понимать программную систему [7], обладающую свойствами автономности, реактивности, активности и коммуникативности.

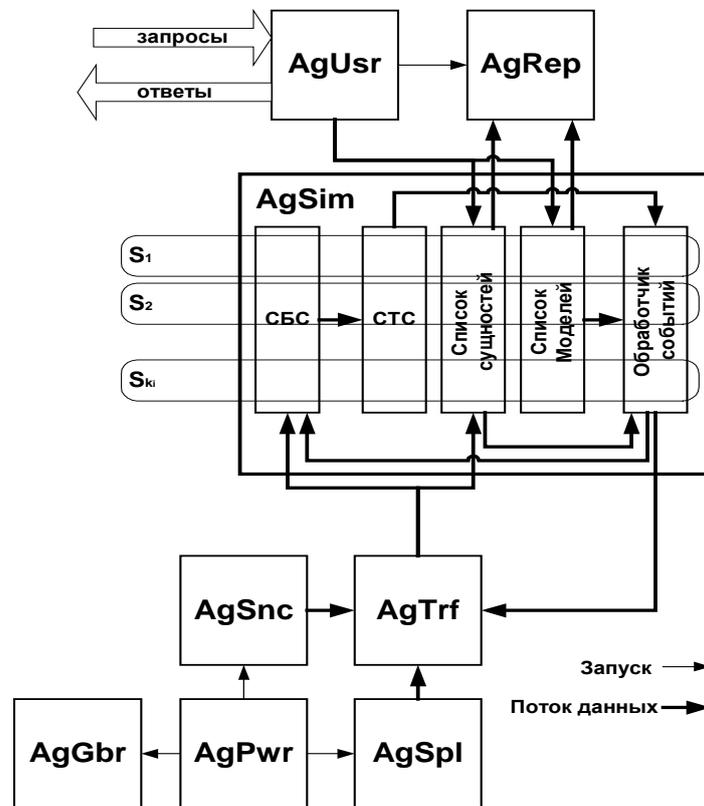
Система OpenGPSS использует технологию взаимодействующих агентов, базируется на кластере и поддерживает динамическую переконфигурацию кластера в случае отказа или при подключении нового сервера. Информация о сетевой топологии кластера используется только на системном уровне, что обеспечивает прозрачность имитации для пользователей. В GPSS-программе нет директив распараллеливания,

свойственных языкам параллельного программирования, например, таким, как HPF и OpenMP Fortran [8].

Система OpenGPSS работает в гетерогенной сети, на различных архитектурах компьютеров с различными операционными системами. Для синхронизации модельного времени используется оптимистический подход [9], который, в отличие от консервативного подхода, обеспечивает большую производительность, хотя и увеличивает сложность реализации системы. Для удобства пользователей реализован WEB-интерфейс для работы через Интернет. Обеспечение отказоустойчивости достигается с помощью механизмов репликации и других способов защиты данных.

Агентная архитектура сервера моделирования

Рассмотрим более подробно архитектуру сервера моделирования SS (рисунок), состоящую из взаимодействующих агентов: агент имитационного моделирования AgSim, агент репликации модели AgRep; агент разделения нагрузки AgSpl; агент синхронизации AgSnc; агент приёмо-передачи сообщений AgTrf; агент производительности сервера AgPwr; агент взаимодействия с пользователем AgUsr; агент сборки мусора AgGbr.



Агентная архитектура сервера моделирования

Пусть $A_i = \{ A_i^{AgSim}, A_i^{AgRep}, A_i^{AgSpl}, A_i^{AgSnc}, A_i^{AgTrf}, A_i^{AgPwr}, A_i^{AgUsr}, A_i^{AgGbr} \}$ – множество активных агентов i -го сервера, $i = \overline{1, n}$, где n – количество серверов имитационного моделирования в кластере, на языке GPSS и EML/Java.

Введем матрицу B_{iz} размерности $n \times n$, которая показывает, есть ли связь между i -м и z -м сервером:

$$\mathbf{B}_{iz} = \begin{cases} 1, & \text{если существует связь между } i\text{-м и } z\text{-м сервером;} \\ 0, & \text{в противном случае;} \end{cases}, \quad i = \overline{1, n}, \quad z = \overline{1, n}.$$

Пусть S_{ij} – j -е задание пользователя на i -ом сервере, $i = \overline{1, n}$, $j = \overline{1, k_i}$, где k_i – количество заданий на i -ом сервере. Обозначим через F_i – устаревшие задания пользователя i -го сервера. Множество модельных времен i -го сервера определим как $T_i = \{t_{i1}, t_{i2}, \dots, t_{ij}, \dots, t_{ik_i}\}$, а результаты выполнения j -го задания пользователя на i -м сервере обозначим R_{ij} . Пусть $m_{ijz} = (m_{ijz1}, m_{ijz2})$ – маркеры начала и конца обработки блока, и $M_{ij} = \bigcup_z (m_{ijz1}, m_{ijz2})$ – разделение всего GPSS-сценария на непересекающиеся части для разных серверов, при этом берутся такие i и z , для которых $\mathbf{B}_{iz} = 1$, причем $M_i = \bigcup_{j=1}^{k_i} M_{ij}$.

Обозначим через D_{ij} – очереди исходящих сообщений для j -го задания на i -м сервере, а через Q_{ij} очереди входящих сообщений на i -м сервере, причем $Q_i = \bigcup_{j=1}^{k_i} Q_{ij}$ и

$$D_i = \bigcup_{j=1}^{k_i} D_{ij}.$$

Состояние i -го сервера опишем как $P_i(A_i, S_i, F_i, T_i, R_i, M_i, Q_i, D_i)$, а состояние кластера моделирования – $P = \bigcup_n P_i(A_i, S_i, F_i, T_i, R_i, M_i, Q_i, D_i)$.

Определим агент имитационного моделирования AgSim как незамкнутую систему дискретно-событийного имитационного моделирования, состоящую из списка будущих событий (СБС), списка текущих событий (СТС), списка моделей, списка сущностей и обработчика событий. Списки логически разделяются между пользователями (сессиями) Oacle. Работа этого агента описывается оператором:

$$A_{ij}^{AgSim} : P_i(\{t_{i1}, t_{i2}, \dots, t_{ij}, \dots, t_{ik_i}\}) \rightarrow P_i(\{t_{i1}, t_{i2}, \dots, t_{ij}, \dots, t_{ik_i}\}), \text{ где } t_{ij} \leq t_{ij}^{\cdot}. \quad (1)$$

После завершения моделирования j -го задания, агент имитационного моделирования сохраняет результаты в списке результатов R_i :

$$A_{ij}^{AgSim} : P_i(A_i, R_i) \rightarrow P_i(A_i \setminus \{A_{ij}^{AgSim}\}, R_i \cup R_{ij}).$$

Агент взаимодействия с пользователем AgUsr отвечает за взаимодействие системы с пользователями. Он транслирует текст GPSS-программы [10] во внутренний формат системы (строки модели, сущности), а далее заносит в список заданий (job) сервера новую задачу проведения имитации с помощью оператора:

$$A_i^{AgUsr} : P_i(A_i, S_i, T_i) \rightarrow P_i(A_i \cup \{A_{ik_i+1}^{AgSim}\}, S_i \cup \{s_{ik_i+1}\}, T_i \cup \{t_{ik_i+1}\})$$

Любое изменение структуры GPSS-модели (перегрузки блоков) приводит к «устареванию» старой задачи и созданию на ее основе новой.

При удалении j -го задания, $1 \leq j \leq k_i$ этот агент переносит задание из списка активных заданий S_i в список устаревших заданий F_i и очищает все списки i -го сервера от j -го задания с помощью оператора:

$$A_i^{AgUsr} : P_i(A_i, S_i, F_i, T_i) \rightarrow P_i(A_i \setminus \{A_{ij}^{AgSim}\}, S_i \setminus \{s_{ij}\}, F_i \cup \{s_{ij}\}, T_i \setminus \{t_{ij}\} | \forall i, z, \mathbf{B}_{iz} = 1).$$

Агент синхронизации AgSnc определяет окончание моделирования, когда входящие списки агентов приёмо-передачи сообщений для j -го задания, списки СБС и СТС пусты, и проводит распределенное подтверждение – синхронизирует сущности на серверах. После этого, по запросу пользователя, агент AgUst генерирует стандартный отчет с результатами моделирования.

Агент репликации модели AgRep, который описан в [11], копирует GPSS-модель во внутреннем формате (модель, сущности) для данного задания с i -го сервера на z -й:

$$A_i^{AgRep} : P_z(S_z, F_z, T_z, R_z) \rightarrow P_z(S_z \cup (S_i \setminus S_z), F_z \cup (F_i \setminus F_z), T_z \cup \{t_{ij} \mid \forall j, s_{ij} \in (S_i \setminus S_z)\}, R_z \cup (R_i \setminus R_z)), \forall i, z : \mathbf{B}_{iz} = 1.$$

На каждом сервере хранится своя копия задания, и далее при балансировке нагрузки передаются только маркеры начала и конца обработки. Задача этого агента состоит в том, чтобы реплицировать модель на как можно большее количество серверов в кластере.

Агент приёмо-передачи сообщений AgTrf обеспечивает обмен сообщениями между серверами и поддерживает целостность системы относительно модельного времени:

$$A_{ij}^{AgTrf} : P_z(Q_z) \rightarrow P_z(Q_z \cup \{d_{ijz} \mid \forall z, \mathbf{B}_{iz} = 1, z \in M_{ij}\}).$$

При обработке транзактов агентом имитационного моделирования AgSim возможны два режима передачи сообщений: локальный и глобальный, которые зависят от маркера начала обработки и маркера конца обработки транзактов. В первом случае происходит попадание события в локальный кэш событий: транзакт не нужно передавать на другой сервер, и он передается в СБС (1). Во втором, если вышли за маркер конца обработки, сообщение пересылается агенту приёмо-передачи сообщений (2), который рассылает этот транзакт следующему серверу из списка M_{ij} .

$$A_{ij}^{AgSim} : P_i(\{t_{i1}, t_{i2}, \dots, t_{ij}, \dots, t_{ik}\}, D_i) \rightarrow P_i(\{t_{i1}, t_{i2}, \dots, t_{ij}, \dots, t_{ik}\}, D_i \cup \{d_{ij} \mid m_{ijz} \in M_{ij}, \mathbf{B}_{iz} = 1\}), \quad (2)$$

где $t_{ij} \leq t_{ij}$.

Если при обработке происходит изменение состояний сущностей, то всегда генерируется сообщение об изменении состояния сущности и далее передаётся агенту AgTrf, который опять, используя M_{ij} передаёт сообщения нужным серверам.

Агент разделения нагрузки AgSpl распределяет нагрузку среди серверов в кластере в зависимости от структуры GPSS-модели (статический анализ) и нагрузки серверов, полученной от агента AgPwt (динамический анализ). Работа агента описывается следующим оператором:

$$A_{ij}^{AgSpl} : P_i(M_i) \rightarrow P_i(M_i \cup M_{ij}).$$

Результатом работы агента является передача серверам (а именно, обработчикам событий каждого сервера) маркеров начала обработки и маркеров конца обработки. Агентом формируется список M_{ij} , в котором хранится информация, какие сообщения переслать каким серверам. Этот список агент AgSpl передает агенту приёмо-передачи данных для дальнейшего хранения и использования для передачи сообщений. На агент разделения нагрузки также возложено выполнение переконфигурации в случае отказа одного из серверов.

Основными задачами агента синхронизации AgSnc являются глобальное управление выполнением имитаций и поддержание непротиворечивости моделей в точках сохранения. В системе OpenGPSS используется оптимистический подход синхронизации, и агент AgSnc используется как механизм глобального контроля через вычисление глобального виртуального модельное время t_{ij}^{GVT} для каждого задания сервера.

Непротиворечивость моделей в точках сохранения необходима для обеспечения отказоустойчивости модели (в случае отказа одного из серверов информация оказывается продублированной на остальных.) Работа агента защищена от сбоев – используется распределенная транзакция с двухфазным подтверждением 2PC. Первая фаза начинается перед определением агентом глобального виртуального модельного времени t_{ij}^{GVT} . В контрольных точках все сущности и все списки на серверах одинаковы, чем и достигается непротиворечивость имитации. Обновление данных о сущностях происходят путем послыки сообщений с модельным временем t_{ij}^{GVT} о состояниях сущностей агентам AgTgf нужных серверов, на которых, в случае опережения времени t_{ij}^{GVT} , происходят откаты. При успешно проведенной фазе обновления выполняется фаза подтверждения транзакции, а на каждом сервере – фиксация транзакции (commit). СУБД Oracle поддерживает систему контрольных точек [15], которые используются для восстановления состояния заданий серверов. В случае неудачного обновления производится откат до предыдущей, успешной, контрольной точки.

На основе полученной от агента AgPwg информации вычисляется время следующего запуска, и, наконец, агент AgSnc переходит в режим ожидания.

Агент производительности сервера AgPwg периодически оценивает загрузженность процессора и размеры СБС, СТС, входящей и выходящей очереди агента приема-передачи сообщений. При размерах списков больше пороговых происходит неотложный запуск агента AgSpl.

Агент сборки мусора AgGbr предназначен для удаления неиспользуемых моделей. Когда агент AgPwg определяет небольшую загрузку сервера, запускается агент AgGbr, который удаляет завершившиеся программы и их статистические отчеты на каждом сервере после заданного времени ожидания. Работа агента описывается оператором $A_i^{AgGbr} : P_i(S_i, F_i, T_i) \rightarrow P_i(S_i \setminus F_i, F_i, T_i \setminus \{t_{ij} \mid \forall j, s_{ij} \in F_i\})$.

Заключение. Предложенная агентная архитектура распределенной дискретно-событийной системы имитационного моделирования построена на основе СУБД Oracle. Использование агентов в таких системах повышает масштабируемость, отказоустойчивость и надежность, при незначительном снижении производительности из-за избыточности данных и поддержания целостности имитаций. Перспективным является оптимизация режимов работы кластера для увеличения скорости моделирования, а также построение и исследование кластера с большим количеством узлов (10–20).

Литература

1. SPEEDES. <http://www.speedes.com>
2. Mascarenhas E., Knop F., Vernon R. ParaSol: A multithreaded system for parallel simulation based on mobile threads. Winter Simulation Conference 1995.
3. Шрайбер Томас Дж. Моделирование с использованием GPSS. М.: Машиностроение, 1980. 593 с.
4. Томашевский В. Н., Жданова Е. Г. Имитационное моделирование в среде GPSS. М.: Бестселлер, 2003. 416 с.
5. Киевский Центр Имитационного Моделирования. <http://www.simulation.kiev.ua>

6. WebGPSS. <http://www.webgpss.com>
7. **Томашевский В. Н., Диденко Д. Г.** Агентная архитектура распределенной дискретно-событийной системы имитационного моделирования OpenGPSS//Системні дослідження та інформаційні технології. 2006. № 4. К.: ВПК «Політехніка», 2006. С.123–133.
8. **Воеводин В. В., Воеводин В. В.** Параллельные вычисления. Научное издание. СПб.: BHV.
9. **Fujimoto Richard M.** Parallel And Distributed Simulation Systems. Wiley, 2000.
10. **Ахо Альфред, Сети Рави, Ульман Джеффри.** Компиляторы: принципы, технологии, инструменты. Пер. с англ. М.: Вильямс, 2001. 768 с.
11. **Диденко Д. Г.** Агент реплікації в розподіленій дискретно-подійній системі імітаційного моделювання OPENGPSS//Матеріали міжнародної наукової конференції «Інтелектуальні системи прийняття рішень та прикладні аспекти інформаційних технологій», 2006. С. 264–266.
12. **Кайт Том.** Oracle для профессионалов. Книга 2. Расширение возможностей и защита: Пер. с англ. М.: ДиасофтЮП, 2003. 848 с.