

УДК 004.658.2

## АВТОМАТИЗИРОВАННОЕ ПОСТРОЕНИЕ ИМИТАЦИОННОЙ МОДЕЛИ ИНФОРМАЦИОННОЙ СИСТЕМЫ В ДВУХЗВЕННОЙ АРХИТЕКТУРЕ

**Вербов Денис Валерьевич**

Начальник подразделения, e-mail: [vedenis@yandex.ru](mailto:vedenis@yandex.ru)

**Лавринчук Роман Владимирович**

Ведущий специалист по администрированию БД, e-mail: [romanzi@mail333.com](mailto:romanzi@mail333.com)  
ФГУП Всероссийский научно-исследовательский институт автоматики, Россия, Москва,  
101000, Моспочтамт, а/я 918

**Аннотация.** Рассматривается анализ интеграции информационных систем на основе имитационного моделирования, основные подходы к интеграции справочников. Дается обзор моделирования этих процессов в среде имитационного моделирования «JaamSim» и предлагается подход к автоматизированному построению модели.

**Ключевые слова:** базы данных, анализ, интеграция, информационные системы, моделирование.

**Введение.** Задачи интеграции информационных систем остаются актуальными по мере развития информационных технологий и смены различных поколений программного обеспечения и подходов к автоматизации предприятий. Можно заметить, что широко применяемые на практике эксплуатационные решения и перспективные подходы к разработке программного обеспечения отстают друг от друга на одно-два поколения. Так, когда развивались двухзвенные и на подходе были трехзвенные архитектуры, огромной популярностью пользовались файл-серверные решения, и постепенно выходили из эксплуатации системы, основанные на мэйнфреймах. Сейчас в пору зрелости входят средства виртуализации и сервисно-ориентированные системы, а им на смену спешат контейнерные технологии, обусловленные популярностью облачных решений, и микросервисные подходы к реализации распределенных систем. При этом, как и ранее, остаются крайне актуальными вопросы интеграции систем разных поколений. Во многом это обусловлено экономией на средствах интеграции в составе разрабатываемых комплексов в условиях долгосрочного планирования и дефицита средств. Динамизм развития существующих систем также приводит к расхождению с источниками формального описания бизнес-процессов. Таким образом, уже для клиент-серверных систем необходимы инструменты анализа и внешнего аудита.

**1. Применение имитационного моделирования при анализе сложных автоматизированных систем.** Как правило, при разработке информационных систем (ИС) используют либо аналитические методы, либо проводят тестирование прототипа, однако для анализа существующих систем, отличающихся, в процессе эволюции, внутренней сложностью и взаимосвязями, предпочтительно использовать специальные автоматизированные средства, например, основанные на методах моделирования. Для существующих систем они позволяют, с одной стороны, выполнить исследование их работы

без приостановки рабочих процессов, а с другой – прогноз результатов и возможных проблем интеграции.

В данной статье предлагается использовать средства имитационного моделирования для анализа существующих клиент-серверных архитектур с реляционной системой управления базой данных (СУБД), как центральной частью построения комплекса ИС.

Разумеется, имитационное моделирование не может охватить все аспекты работы системы. Так, например, производительность работы базы данных (БД) может зависеть не только от выбранного алгоритма обработки данных, но и от особенностей реализации системы хранения, кода драйверов, работы менеджера процессов и памяти операционной системы. Как следствие, для полноценного анализа работы системы предпочтительна реализация тестового стенда, фактически в основных чертах повторяющего рабочую систему.

Тем не менее, для комплексных систем, на работу которых влияние отдельных технических элементов не столь значительно, а построение полноценных тестовых стендов затруднено, моделирование оказывается предпочтительным средством анализа. Также интересен анализ на основе моделирования для проверки работы алгоритмов в идеальном случае без привязки к специфике реализации.

Так, в качестве примера применения моделирования для анализа отдельных алгоритмов работы системы управления базой данных (СУБД) можно привести статьи [2] и [6], где рассматриваются отдельные аспекты управления транзакциями в ядре СУБД.

Пример моделирования работы СУБД для приложения, реализованного в архитектуре «клиент-сервер», приводится в работе [4]. В этой модели не учитываются задержки передачи данных по сети и потребление иных ресурсов сервера, кроме дисковой подсистемы и процессора. Пример моделирования взаимодействия нескольких БД в рамках распределенных транзакций приводится в работе [1].

Однако вопросы моделирования, учитывающие потоки данных и взаимодействие систем в рамках одной или нескольких БД, недостаточно подробно освещаются в известной автору литературе.

В разрезе анализа унаследованных ИС с двухзвенной архитектурой в первую очередь интерес представляют влияние изменений на результирующую нагрузку, с одной стороны, а с другой, выявление наиболее значимых источников данных (далее справочников).

При этом с практической точки зрения крайне полезным является сопоставление имеющейся нагрузки с конкретным справочником. Это позволяет получить хотя бы приблизительную оценку влияния процессов интеграции данных на действующие приложения и подсистемы.

**2. Подходы к интеграции автоматизированных систем и моделированию процессов.** Рассмотрим основные подходы к интеграции БД на примере СУБД Oracle. Можно выделить следующие подходы, описываемые ниже. Первый подход – это прямая связь по технологии «DBLINK» между базами с удаленным выполнением запросов. При этом справочники распределены между базами, соответственно задачам, в которых они используются. Второй подход оптимизирует взаимодействие за счет кэширования данных и использует механизм материализованных представлений («MVIEW»). Третий подход предполагает реорганизацию справочников, ведение истории по ним и иерархическое обновление. Наконец четвертый подход в общем случае предполагает переход от

двухзвенной архитектуры к многоуровневой и применение так называемой шины данных. Поскольку последний подход выходит за рамки рассматриваемой архитектуры, в данной статье не рассматривается.

Кратко опишем вышеперечисленные подходы.

По технологии «DBLINK» каждая из прикладных систем непосредственно обращается к справочным данным в общем случае через случайные промежутки времени, так как момент обращения определяется работой оператора и приложения. При этом создается нагрузка как на каналы передачи, так и на подсистему ввода-вывода сервера источника и память приемника. Нагрузку на центральное процессорное устройство (ЦПУ) можно считать минимальной в случае простых запросов.

Технология «MVIEW» предполагает кэширование данных с периодическим обновлением кэша из данных источника и сохранением их на приемнике. Как правило, промежутки времени заданы априори. Это позволяет детерминировать нагрузку для отдельно взятого набора данных. При несогласованном обновлении множества наборов данных могут возникать локальные пики нагрузки. Кроме того, полное обновление локального справочника оказывает влияние на обращение прикладных систем к кэшируемым данным. Наконец, иерархическое хранение, если таковое еще не было реализовано, позволяет избежать ряда отмеченных проблем за счет усложнения логики, как ведения справочника, так и обращения к его данным. При этом предполагается изменение кода существующих приложений и базы данных. В данном подходе для обновления справочников реализуется мастер-копия с централизованным хранением и историей изменений.

Для планирования, анализа необходимости и возможных результатов таких изменений предлагается использовать методы дискретно-событийного моделирования.

В настоящее время доступны различные реализации систем моделирования, как коммерческие, так и с открытым исходным кодом. Достаточно подробный обзор и сравнение последних приводятся в статье [3]. В качестве инструмента моделирования был выбран программный пакет «JaamSim» [5]. Критериями выбора были, во-первых, доступность решения (бесплатность, открытый исходный код), во-вторых, наличие наглядных средств визуализации, в-третьих, возможность расширяемости и автоматизации построения модели.

Рассмотрим различные варианты моделирования ранее описанных процессов.

Так, частная модель, представляющая из себя имитацию обновления материализованного представления, состоит из следующих элементов. В качестве транзактов выбраны операции обновления справочников и запросы обращения к данным справочников. Источниками транзактов являются генератор событий заданий обновления и генератор обращений пользователей. При этом события генератора обновления носят детерминированный характер с пренебрежимо малым значением отклонения. В то же время обращения пользователей не детерминированы и происходят в случайные моменты времени.

Обработка транзактов выполняется на двух серверах для случая операции обновления и только на одном из них, когда выполняется запрос данных от пользователя справочника. В модели учитываются времена обработки запросов, время задержки передачи данных по сети и время записи данных на диск. По логике работы материализованного представления во время операции обновления данные не доступны для запросов. Эта ситуация моделируется с помощью очереди. Кроме того, обработка запросов на сервере также ограничена количеством свободных процессорных ядер, что также моделируется с помощью очередей.

Для случая множества баз данных и приложений необходимо реализовать автоматизированное построение модели. Для этого выделим следующие элементы будущей модели. В качестве активных генераторов нагрузки можно рассматривать как приложения, так и внутренние задания системы. При этом в рамках модели можно агрегировать ряд источников со схожей нагрузкой, рассчитав средние параметры нагрузки (процессор, память, сеть, ввод-вывод) и частотные характеристики генерации транзактов операций. Под транзактами операций понимаются запросы к БД. В качестве обработчиков нагрузки будут выступать сервера БД, которым будут сопоставлены ресурсы, моделирующиеся счетчиками и очередями.

Так как в рамках модели необходимо получить не просто оценку нагрузки, но и учитывать конечные точки потоков данных, необходимо каждому транзакту в модели сопоставить данные реляционных отношений, к которым он обращается, и тип операции.

Это важное уточнение позволяет в рамках модели учитывать влияние различных приложений друг на друга, например, когда осуществляется перенос подсистемы на другой обработчик нагрузки, при том условии, что к данным системы обращаются не только непосредственные пользователи данного приложения, но и, косвенно, пользователи сторонних подсистем.

Оценка влияния такой косвенности не только важна для понимания внутренних зависимостей между приложениями, но и, в ряде случаев, не может быть получена напрямую из существующего формального описания информационных систем. Это объясняется тем, что связи между приложениями могут возникать по мере появления новых подсистем, а также по мере развития и совершенствования имеющихся.

Для моделирования архитектуры с выделением справочников на основе схемы посредника (федерализация данных) необходимо учесть, с одной стороны, усложнение ведения справочника, а с другой, упрощение запросов к нему от внешних систем. Так как данные оценки не могут быть напрямую получены из действующих систем, предлагается проводить корректировку модели на основе данных прототипа.

Автоматизацию построения модели предлагается осуществлять за счет агрегации данных систем мониторинга и внутренних данных каталога СУБД.

**Заключение.** Рассмотренный подход к моделированию существующей инфраструктуры клиент-серверных информационных систем позволяет как дать оценку нагрузки от отдельных подсистем, так и спрогнозировать последствия внесения изменений или замены отдельных подсистем. В условиях неполной информации об объекте и его действующих ИС, такое моделирование позволяет избежать как технических ошибок, так и потенциальных финансовых рисков при развитии, поддержке и интеграции унаследованных ИС.

В качестве дальнейших шагов планируется рассмотреть вопрос о моделировании совместной работы нескольких архитектур, а также проработать задачу оценки масштабирования нагрузки при построении модели.

#### СПИСОК ЛИТЕРАТУРЫ

1. Василева С., Носков Ю.М. Моделирование распределенной двухверсионной двухфазной блокировки // Имитационное моделирование. Теория и практика: Сборник докладов

- четвертой всероссийской научно-практической конференции ИММОД-2009. Том 2. СПб.: ОАО «ЦТСС». 2009. С. 304 – 308.
2. Agrawal R., Carey M. J., Livny M. Concurrency control performance modeling: Alternatives and implications // ACM Transactions on Database Systems (TODS). 1987. Т. 12. №. 4. Pp. 609 – 654. DOI=<http://dx.doi.org/10.1145/32204.32220>
  3. Dagkakis G., Heavey C. A review of open source discrete event simulation software for operations research // Journal of Simulation. 2015. Т. 1. Pp. 14.
  4. Garcia D. F. Performance modeling and simulation of database servers //The Online Journal on Electronics and Electrical Engineering (OJEEE). 2010. Т. 2. №. 1. Pp. 183., <http://infomesr.org/attachments/W09-0034.pdf>
  5. King D. H., Harrison H. S. Open-source simulation software JaamSim // Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World. IEEE Press. 2013. Pp. 2163 – 2171.
  6. Miller J. A. Simulation of database transaction management protocols: Hybrids and variants of Time Warp // Proceedings of the 24th conference on Winter simulation. ACM. 1992. Pp. 1232 – 1241. DOI=<http://dx.doi.org/10.1145/167293.167905>
- 

**УДК 004.658.2**

**AUTOMATED CONSTRUCTION OF THE SIMULATION MODEL OF THE  
INFORMATION SYSTEM IN A TWO-TIER ARCHITECTURE**

**Denis V. Verbov**

Department Manager, e-mail: [vedenis@yandex.ru](mailto:vedenis@yandex.ru)

**Roman V. Lavrinchuk**

Senior database administrator, e-mail: [romanzi@mail333.com](mailto:romanzi@mail333.com)

All-Russia Research Institute of Automatics,  
Moscow Post Office, P.O. Box 918, Moscow 101000, Russia

**Abstract.** The analysis of integration of information systems on the basis of simulation modeling, basic approaches to the integration of directories is considered. An overview of the modeling of these processes in the simulation environment of "JaamSim" is given and an approach to the automated construction of the model is proposed.

**Keywords:** database, analyses, integration, information systems, modeling

**References**

1. Vassileva S., Noskov Yu.M. Modelirovaniye raspredelennoy dvukhversionnoy dvukhfaznoy blokirovki [Simulation of a distributed two-phase locking dwuhvalentnoe] // Simulation. Theory and practice: Collection of reports of the fourth all-Russian scientific-practical conference IMMOD-2009. Volume 2. SPb.: JSC "CTS". 2009. Pp. 304 – 308. (in Russian).
2. Agrawal R., Carey M. J., Livny M. Concurrency control performance modeling: Alternatives and implications // ACM Transactions on Database Systems (TODS). 1987. Т. 12. №. 4. Pp. 609 – 654. DOI=<http://dx.doi.org/10.1145/32204.32220>

3. Dagkakis G., Heavey C. A review of open source discrete event simulation software for operations research // Journal of Simulation. 2015. Т. 1. Pp. 14.
4. Garcia D.F. Performance modeling and simulation of database servers //The Online Journal on Electronics and Electrical Engineering (OJEEE). 2010. Т. 2. №. 1. Pp. 183. <http://infomesr.org/attachments/W09-0034.pdf>
5. King D.H., Harrison H.S. Open-source simulation software JaamSim // Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World. IEEE Press. 2013. Pp. 2163 – 2171.
6. Miller J.A. Simulation of database transaction management protocols: Hybrids and variants of Time Warp // Proceedings of the 24th conference on Winter simulation. ACM. 1992. Pp. 1232 – 1241. DOI=<http://dx.doi.org/10.1145/167293.167905>