

УДК 681.3

Аналізуються моделі організації взаємодії клієнтів і розподіленої бази даних. Обґрунтовується вибір тривірвнєвої моделі. Будується імітаційна модель обраної архітектури в AnyLogic. Досліджуються характеристики моделі
Ключові слова: розподілена база даних, сервер додатків

Анализируются модели организации взаимодействия клиентов и распределенной базы данных. Обосновывается выбор трехуровневой модели. Строится имитационная модель выбранной архитектуры в среде AnyLogic. Исследуются характеристики модели
Ключевые слова: распределенная база данных, сервер приложений

The models of interaction of clients and distributed database are analyzed. The choice of a three-level model is proved. We construct a simulation model in AnyLogic for study the characteristics of the model
Keywords: distributed database, application server

ИССЛЕДОВАНИЕ МОДЕЛИ ОБРАБОТКИ ЗАПРОСОВ К БАЗЕ ДААННЫХ В СРЕДЕ ANYLOGIC

Ю. Г. Шаталова

Кандидат технических наук, доцент
 Кафедра кибернетики и вычислительной техники
 Севастопольский национальный технический университет
 ул. Университетская, 33, г. Севастополь, Украина,
 99053
 Контактный тел.: (055) 210-69-59
 E-mail: volnajulia@mail.ru

1. Введение

Исследования, приведенные в статье, относятся к области организации распределенных баз данных (РБД). Различные объекты одной базы данных могут располагаться на удаленных компьютерах и управляться различными системами управления базами данных (СУБД). Запросы к базе поступают случайным образом от удаленных пользователей. Поэтому на распределенную систему накладываются дополнительные задачи: провести идентификацию поступающих запросов, проверить возможность обработки запроса, передать пользователю сообщение в случае неудачи обработки с объяснением причины.

Распределенная система должна обеспечить решение следующих задач [1]. 1). БД в каждый момент должна отражать текущее состояние предметной области, причем данные, которые хранятся в БД, должны быть непротиворечивыми и актуальными. 2). БД должна отражать некоторые правила предметной области, законы, по которым она функционирует (бизнес-логика). 3). Постоянный контроль за состоянием БД, отслеживание всех изменений и адекватная реакция на них. 4). Возникновение некоторой ситуации в БД четко и оперативно должно влиять на ход выполнения прикладной задачи. 5). Контроль типов данных. Кроме перечисленных задач, непосредственно связанных с БД на систему возлагаются дополнительные «организационные» задачи: надежность, быстротворчество, гарантированное

обслуживание пользовательских запросов, проверка запросов на право доступа и т.д. При повышении количества пользователей на базу данных возрастает нагрузка.

Поэтому актуальной задачей является выбор архитектуры распределенной системы, обеспечивающей выполнение перечисленных требований.

2. Анализ литературных данных и выбор модели организации взаимодействия клиентов с РБД

В основе работы распределенной системы лежит модель «клиент - сервер». В открытых системах клиенты и сервер располагаются на различных компьютерах. Из литературных источников известно существование следующих моделей взаимодействия клиентов и сервера РБД [2]: 1) модель удаленного управления данными; 2) модель файлового сервера; 3) модель сервера базы данных; 4) модель сервера приложений. На основе анализа достоинств и недостатков перечисленных моделей выбрана наиболее эффективная архитектура распределенной системы [3] - трехуровневая модель с использованием сервера приложений.



Рис. 1. Модель сервера приложений

Идея модели заключается в том, чтобы распределить функции РБД на три уровня, для оптимизации использования потенциала системы. Весь набор функций можно разделить на 5 групп [2]: 1) функции ввода и отображения данных; 2) прикладные функции, определяющие основные алгоритмы решения задач приложения (бизнес-логика); 3) функции обработки данных внутри приложения; 4) функции управления информационными ресурсами базы данных; 5) служебные функции, играющие роль связок между функциями первых четырех групп. На рис. 1 приведена схема архитектуры взаимодействия клиента с РБД с использованием сервера приложений.

Серверы приложений составляют новый промежуточный уровень архитектуры. Они могут исполнять общие функции для клиентов. Одна из наиболее важных функций, выносимых на этот уровень - проверка возможности обработки запросов и выполнения транзакций. За счет возможности использования нескольких промежуточных серверов, повышается надежность и быстродействие системы. Серверы баз данных в этой модели занимаются исключительно функциями СУБД: обеспечивают функции создания и ведения БД; поддерживают целостность реляционной БД; обеспечивают функции хранилищ данных. Кроме того, на них возлагаются функции создания резервных копий БД и восстановления БД после сбоев, управления выполнением транзакций и поддержки приложений.

Достоинства трехуровневой модели с сервером приложений: 1) снижение нагрузки на трафик; 2) уменьшение нагрузки на сервер базы данных; 3) отсутствие необоснованного дублирования кода приложений, за счет переноса общего кода на сервер приложений; 4) отсеивание «неуполномоченных» запросов до их поступления на сервер базы данных; 5) клиентские приложения не зависят от конкретной СУБД, что увеличивает переносимость системы и ее масштабируемость.

Таким образом, проведенный анализ позволяет сделать вывод о целесообразности применения трехуровневой архитектуры. Однако, для ее эффективного применения необходимо исследовать следующие характеристики модели: 1) оптимальное количество серверов приложений; 2) загрузку сервера приложений при изменении интенсивности поступления запросов от клиентов; 3) дополнительную задержку запроса на сервере приложений.

3. Цель и задачи исследования

Целью данной работы является исследование характеристик модели взаимодействия клиентов с распределенной базой данных на основе использования сервера приложений. Требуется оценить задержку обслуживания запросов пользователя на промежуточном сервере, загрузку промежуточного сервера, оценить необходимое количество серверов приложений для безотказной работы системы.

Для достижения цели исследования необходимо решить следующие задачи. 1) Построить имитационную модель взаимодействия. 2) Организовать проведение эксперимента при изменении интенсивности поступления запросов от клиентов. 3) На основании экспериментов собрать статистику поведения модели. 4) Провести анализ собранных данных. 5) Сформулировать предложения по повышению эффективности работы системы.

4. Построение имитационной модели взаимодействия в среде AnyLogic

В качестве среды моделирования был выбран язык моделирования AnyLogic 5.4.1. AnyLogic – инструмент имитационного моделирования нового поколения, основанный на результатах, полученных в теории моделирования и в информационных технологиях [4]. Основными строительными блоками модели AnyLogic являются активные объекты, которые позволяют моделировать любые объекты реального мира. Активный объект является экземпляром класса активного объекта, который реализуется путем создания классов активных объектов (или использованием объектов библиотек AnyLogic) и заданием их взаимосвязи. AnyLogic интерпретирует создаваемые графически классы активных объектов в классы Java. Поэтому модель позволяет пользоваться всеми преимуществами объектно-ориентированного моделирования. Схема построенной имитационной модели представлена на рис. 2.

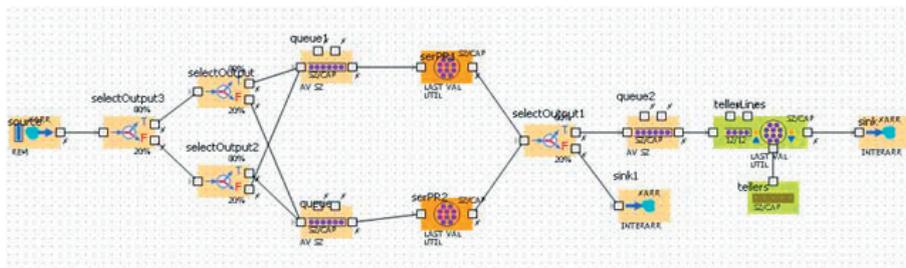


Рис. 2. Схема имитационной модели

5. Организация и проведение эксперимента

Для получения статистики, проведем эксперимент с имитационной моделью. Полагаем, что запросы поступают на вход модели случайным образом и распределены по экспоненциальному закону. Интенсивность поступления запросов меняется, выберем для проведения экспериментов интервал изменения интенсивности от 0,5 до 2. Полагаем также, что на промежуточном уровне используется два равноценных сервера приложений, со средним временем обработки запроса 1 мин. Организация процесса обработки запросов сводится к проверке готовности серверов приложений, в случае одобрения запрос поступает в очередь к одному из серверов приложений, затем на сервер, где проходит его предварительная обработка. Далее, если запрос проверен и достоверен, то поступает на дальнейшую обработку на сервер базы данных, иначе перемещается в очередь и ждет освобождение сервера базы данных. Среднее время обработки запроса на сервере базы

данных 6 мин. Был осуществлен «прогон» модели при изменении интенсивности в указанном диапазоне в течение 500 мин. каждый.

6. Анализ результатов

На основе построенной имитационной модели была собрана статистика и проведен анализ системы при изменении интенсивности поступления запросов. Рассчитаем характеристики системы, при этом не будем учитывать время обработки запросов на сервере базы данных, а также время ожидания в очередях к серверу базы данных. Это связано с целями исследования: нас интересуют характеристики именно промежуточного уровня.

Средняя загрузка системы $\rho = (\lambda T_{об}) / K$, где λ – интенсивность поступления запросов (0,8), $T_{об}$ – время обработки запроса на одном сервере приложений, K – количество серверов приложений, $\rho = 0,88$ запросов/мин.

Средняя длина очереди $L = 0,175$.

Среднее число запросов в системе $M = L + \rho K$, $M = 1,9$.

Время ожидания $T_{ож} = L / \lambda$, $T_{ож} = 0,2$ мин.

Среднее время пребывания запроса в системе $T_{пр} = T_{обс} + T_{ожид}$, где $T_{обс}$ – среднее время обслуживания запроса на серверах приложений, $T_{ожид}$ – среднее время ожидания в очередях системы. $T_{пр} = 2,43$ мин.

Таким образом, задержка запроса на промежуточном уровне составляет 2,43 мин, что вполне удовлетворительно, т.к. при такой интенсивности длина очереди к серверу базы данных составляет около трех запросов и время ожидания равно 3,75 мин.

На рис. 3 представлены зависимости длины очередей от интенсивности поступления запросов. На рис. 4 представлены зависимости загрузки серверов от интенсивности поступления запросов.

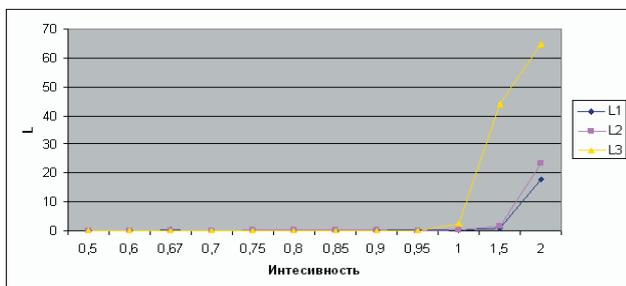


Рис. 3. Зависимости длины очереди L от интенсивности поступления запросов

Из графиков видно, что при интенсивности, не превышающей 0,95, система, включающая два сервера

приложений и один сервер базы данных, работает в нормальном, стационарном режиме, при увеличении интенсивности до 1 и выше, загрузка серверов приложений резко возрастает, длины очередей к этим серверам также увеличиваются. Стационарный режим работы становится невозможным. Сервер базы данных не меняет загрузки, но это связано только с тем, что очереди к нему увеличиваются значительно, и при увеличении интенсивности до 2 система перестает функционировать, поскольку не может обработать поступающие запросы.

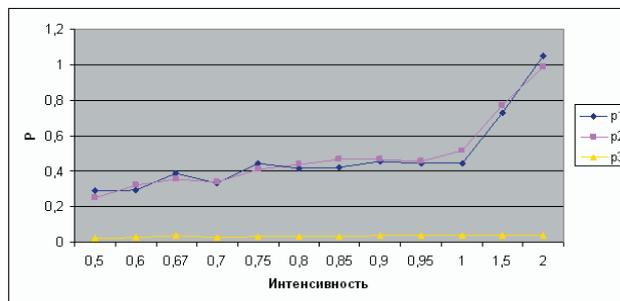


Рис. 4. Зависимости загрузки серверов rho от интенсивности поступления запросов

7. Выводы

В работе построена имитационная модель трехуровневой архитектуры организации взаимодействия клиентов и распределенной базы данных. Исследованы характеристики модели. На основании анализа характеристик можно сделать следующие выводы: при интенсивности поступления запросов не превышающей 0,95 для обеспечения стационарного режима работы и гарантированного обслуживания запросов достаточно двух серверов приложений, при этом среднее время ожидания не превышает 0,2 мин, общее время задержки запроса на промежуточном сервере составляет 2,43 мин.

При возрастании интенсивности поступления запросов до 2, количество серверов приложений должно быть увеличено для обеспечения стационарного режима работы системы, а также должен быть применен дополнительный сервер базы данных.

Следовательно, использование трехуровневой архитектуры ускоряет предварительную обработку запросов к РБД, разгружает сервер базы данных и не вызывает неоправданной задержки запросов на предварительном уровне. Использование двух и более серверов приложений позволяет повысить надежность работы системы и обеспечить гарантированную обработку запроса при выходе из строя одного из серверов.

Литература

1. Дейт, К. Дж. Введение в системы баз данных [Текст] / К. Дж. Дейт. - М.: Вильямс, 2001.-1072 с.
2. Кузнецов, С. Д. Основы баз данных [Текст] / С. Д. Кузнецов – 2-е изд., испр. – М.: Интернет – Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. – 484 с. : ил. – (Серия «Основы информационных технологий»).
3. Мандзо, Р. Экспоненциальная система массового обслуживания с отрицательными заявками и бункером для вытесненных заявок [Текст] / Р. Мандзо, Н. Касконе, Р.В. Разумчик // Автоматика и телемеханика. - 2008. - № 9. - С. 103-113.
4. Карпов, Ю. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. [Текст] / Ю. Карпов. – СПб.: БХВ-Петербург, 2005.–400 с.