

Исследование модели транзакционной системы с репликацией фрагментов базы данных, построенной по принципам облачной среды

С.П. Воробьев, В.В. Горобец

В настоящее время наблюдается тенденция, связанная с развертыванием OLTP-систем на базе облачной архитектуры и использованием технологии распределённой обработки данных. Направление облачных вычислений (Cloud Computing) является быстроразвивающимся и перспективным в современном мире информационных технологий. Идеология данного подхода заключается в переносе вычислений и обработки данных в существенной степени с персональных компьютеров на серверы сети Интернет. Поэтому актуальной является задача моделирования систем, использующих облачные технологии.

В связи с этим в рамках данной работы рассматривается модель архитектуры транзакционной системы, предполагающей размещение фрагментов распределенной базы данных (РБД) в вычислительной сети с произвольной топологией в рамках облачной структуры.

1. Архитектура сети

Архитектура сети (рис. 1) включает сеть с произвольной топологией, соединяющую узлы, которые состоят из ЭВМ и сетевого адаптера, и локальную вычислительную сеть (ЛВС) с регулярной структурой, все ЭВМ которой имеют доступ посредством канала связи в сеть с произвольной топологией. На все линии связи накладываются статические (фиксированные) ограничения по пропускной способности. Архитектура сети с произвольной топологией описывает облачную структуру.

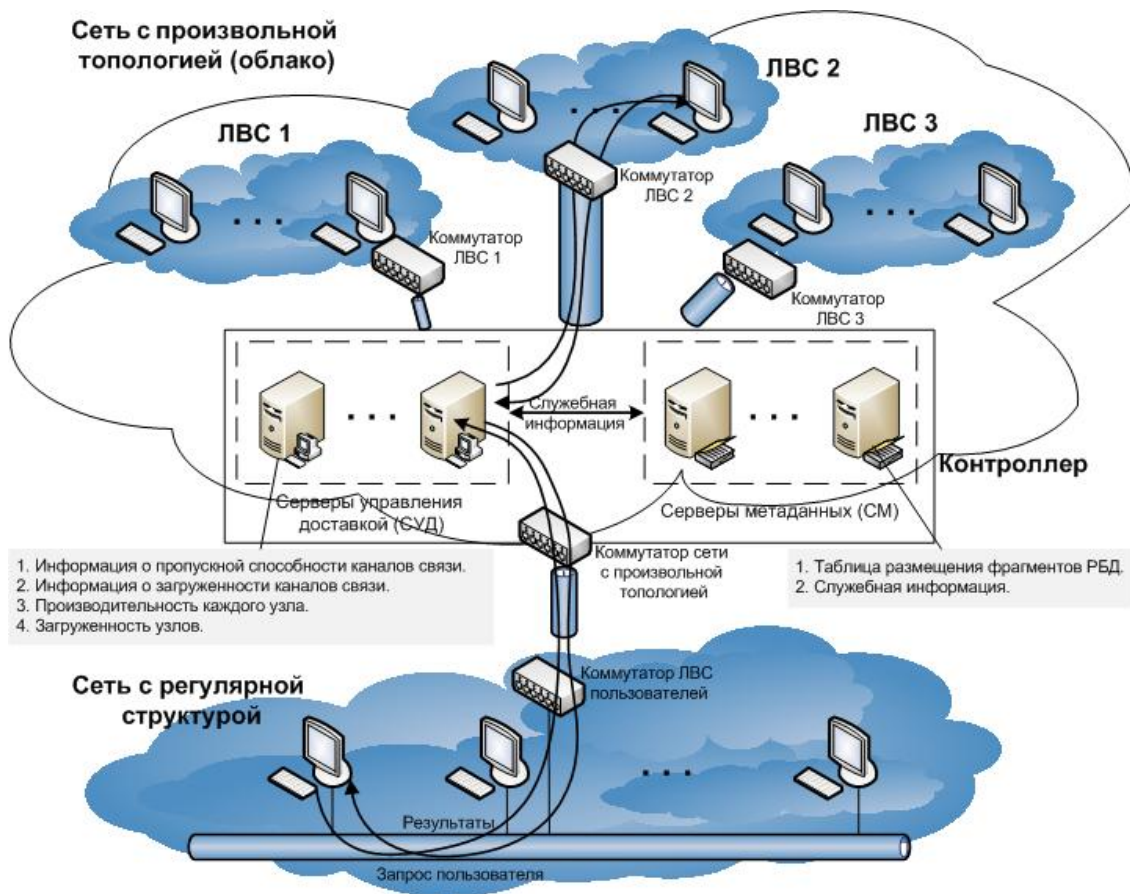


Рис. 1. Сетевая структура

В состав сети произвольной топологии входят узлы хранения (непосредственно рабочие станции, хранящие фрагменты баз данных), серверы управления доставкой контента, серверы метаданных, коммутаторы.

Серверы метаданных (СМ) содержат базы данных (БД) о местоположении и других параметрах фрагментов.

Серверы управления доставкой контента (СУД) выполняют работу по взаимодействию пользователя с распределенной системой и предназначены для последовательного выполнения следующих задач:

- прием запросов от пользователей;
- выбор узлов, которые будут обрабатывать запрос пользователя;
- перенаправление запроса на выбранные узлы;
- прием ответов от узлов на пользовательский запрос;
- составление результирующего набора данных;
- перенаправление ответа сервера на запрос пользователю.

Узлы хранения, содержащие сами фрагменты, могут быть совершенно разными гетерогенными системами, что является принципиальным отличием предлагаемой архитектуры от известных решений.

2. Порядок обработки запросов

Запрос, поступающий на любой узел сети с произвольной топологией, предусматривает доступ к определенному фрагменту РБД. Схема обработки запросов состоит в следующем. Запрос, инициированный на узле, входящем в состав сети регулярной структуры, проходя по линиям связи данной сети, поступает через канал связи во входную очередь контроллера сети с произвольной топологией (попадая на серверы управления доставкой контента). Затем в зависимости от загруженности (длины входной очереди) серверов контроллера запрос поступает во входную очередь конкретного сервера, который обрабатывает запросы в порядке их поступления. СУД производит предварительную обработку запроса.

Каждый сервер метаданных имеет информацию о размещении всех фрагментов РБД по узлам сети с произвольной топологией (облачной структурой). Когда освобождается один из потоков СУД, то из очереди выбирается следующее сообщение (запрос) и передается на обработку серверу метаданных, попадая в очередь. При этом сервер, выбирая сообщение из очереди, анализирует запрос, выявляя фрагменты РБД, необходимые для его выполнения. Затем по хранящейся у него таблице размещения фрагментов РБД сервер определяет узлы сети, на которых находятся требуемые фрагменты, и передает список конкурирующих узлов серверу управления доставкой. Он в свою очередь запрашивает требуемые фрагменты, учитывая при этом рейтинг узлов (чем меньше рейтинг, тем менее загруженным считается узел). Такой запрос фрагмента поступает по каналам связи облачной структуры во входную очередь наименее загруженного узла, при освобождении которого происходит выборка сообщения с запро-

сом из входной очереди и его обработка. После ее окончания результаты передаются инициировавшему данный запрос серверу управления доставкой, где происходит формирование окончательного набора данных, который пересылается по каналам связи узлу сети с регулярной структурой.

Общее значение объема данных, пересылаемых по каналам связи в рамках облачной структуры, зависит от распределения фрагментов по локальным БД сети с произвольной топологией. Чем меньше средний объем пересылаемых данных по каналам сети за единицу времени, тем выше скорость обработки запросов. Он будет минимален, если на каждом узле будет находиться полный набор фрагментов РБД. Но для больших баз данных это практически недостижимо, т.к. объемы памяти узлов ограничены.

Отличительная особенность представленной модели состоит в том, что для уменьшения временных затрат, связанных с обработкой данных в узлах сети и передачей информации по каналу связи, создаются копии (реплики) фрагментов БД, которые размещаются в различных узлах сети облачной структуры. Очевидно, что использование технологии репликации позволяет уменьшить нагрузку на канал связи и благодаря оптимальному размещению копий исключить существенные перегрузки отдельных узлов.

3. Построение модели

Рассмотрим модель системы массового обслуживания без отказов, без приоритетов, с дисциплиной диспетчеризации FIFO, с репликацией фрагментов БД. Пусть кроме того отсутствует сегментация сообщений, а столкновения в канале не учитываются, поскольку используемые для передачи данных протоколы сводят вероятность возникновения коллизий к минимуму. В качестве исходных данных используются:

- множество узлов – $U = \{U_1, U_2, \dots, U_n\}$;
- множество пользователей – $A = \{A_1, A_2, \dots, A_n\}$;
- множество фрагментов базы данных – $R = \{R_1, R_2, \dots, R_d\}$;

– множество запросов – $Q = \{Q_1, Q_2, \dots, Q_q\}$;

– матрица $X = \|x_{R_j U_z}\|$ распределения копий R_j -го фрагмента базы данных, $j = \overline{1, d}$, по узлам сети U_z , $z = \overline{1, n}$:

$$x_{R_j U_z} = \begin{cases} 1, & \text{если копия } R_j\text{-го фрагмента БД размещена в } U_z\text{-м узле;} \\ 0, & \text{в противном случае.} \end{cases}$$

При этом $K_j = \sum_{z=1}^n x_{R_j U_z}$ – количество копий R_j -го фрагмента БД, $j = \overline{1, d}$.

Концептуальная модель рассматриваемой информационной системы представляет разомкнутую сеть массового обслуживания, содержащую:

- приборы, моделирующие работу канала, коммутаторов и узлов;
- буферные памяти канала (коммутаторов), предназначенные для хранения транзакций пользователей;
- буферные памяти узлов.

Описанную модель распределенной системы можно представить в виде системы массового обслуживания (рис. 2).

Необходимо отметить, что рассматриваемая модель предполагает следующее. Узлы сети регулярной структуры, СУД, СМ, все узлы подсетей, входящие в состав сети облачной структуры, в совокупности образуют множество узлов U_i , $i = \overline{1, n}$, где n – суммарное количество всех узлов. Так как СУД самостоятельно не формируют пользовательские запросы, то вероятность формирования ими исходящих запросов равна нулю.

Аналогично положениям, описанным в [1], прикрепление пользователей к репликам фрагментов БД (связь A_s -го пользователя с копией R_j -го фрагмента) может быть реализовано на уровне всех запросов либо на уровне отдельных запросов пользователя. В первом варианте запросы, формируемые A_s -м пользователем, могут обращаться к только определенной копии фрагмента R_j , во втором варианте – к различным его копиям.

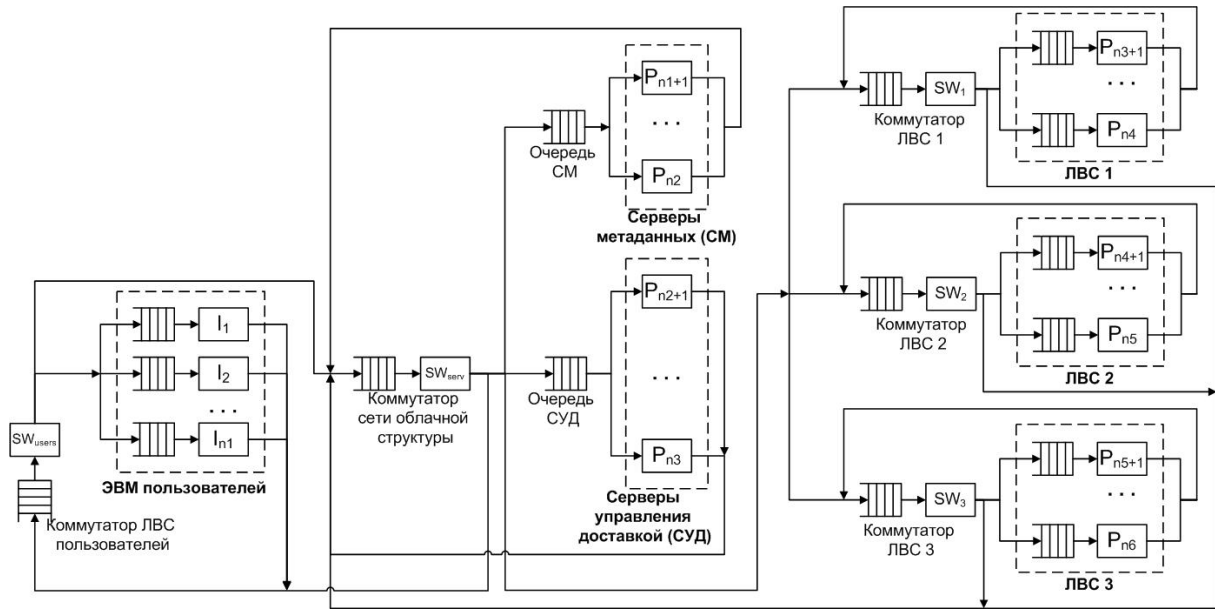


Рис. 2. Концептуальная модель системы массового обслуживания

Рассмотрим сначала вариант прикрепления пользователей к репликам фрагментов базы данных, при котором запросы, формируемые A_s -м пользователем, могут обращаться к только определенной копии фрагмента базы данных R_j . Тогда дополнительно вводится матрица $Y = \|y_{A_s R_j U_z}\|$ прикрепления запросов пользователей к конкретным копиям R_j -го фрагмента базы данных, $s = \overline{1, n}$, $j = \overline{1, d}$, $z = \overline{1, n}$, где

$$y_{A_s R_j U_z} = \begin{cases} 1, & \text{если } A_s\text{-й пользователь прикреплен к копии } R_j\text{-го фрагмента БД, размещенной в } U_z\text{-м узле;} \\ 0, & \text{в противном случае.} \end{cases}$$

Далее рассмотрим вариант прикрепления пользователей к конкретным копиям фрагментов БД на уровне отдельных запросов. Особенность этого способа состоит в том, что если A_s -й пользователь формирует запросы Q_i и Q_θ , которые обращаются к одному и тому же R_j -му фрагменту БД, то возможно прикрепление запросов к различным репликам этого фрагмента, размещенным в узлах U_s и U_z . Тогда вводится матрица $Z = \|z_{A_s Q_i R_j U_z}\|$ прикрепления запросов пользователей к конкретным копиям R_j -го фрагмента БД на уровне отдельных запросов, $s = \overline{1, n}$, $i = \overline{1, q}$, $j = \overline{1, d}$, $z = \overline{1, n}$, где

$$z_{A_s, Q_i, R_j, U_z} = \begin{cases} 1, & \text{если } Q_i\text{-й запрос к } R_j\text{-му фрагменту, формируемый } A_s\text{-м} \\ & \text{пользователем, прикреплен к копии фрагмента } U_z\text{-го уз-} \\ & \text{ла;} \\ 0, & \text{в противном случае.} \end{cases}$$

Для определения интегральных характеристик, таких как время реакции на запросы s -го пользователя ($s = \overline{1, n}$) и время реакции системы, можно использовать соотношения, приведенные в [2, 3].

4. Проведение экспериментов

Пусть необходимо определить, сколько требуется узлов сети произвольной структуры, чтобы в течение заданного промежутка времени не происходило задержек в обработке запросов, но в то же время не было значительного простоя узлов. В табл. 1 приведены значения параметров, заданные при построении данной модели:

Табл. 1. Исходные данные

Характеристика	Значение
Интервал поступления запросов	exp, 60
Интервал обработки запросов сервером управления доставкой	norm, 2, 1
Интервал обработки запросов сервером метаданных	norm, 2, 1
Интервал обработки запросов узлами сети облачной структуры	
– ЛВС 1	unif, 8, 4
– ЛВС 2	unif, 10, 4
– ЛВС 3	unif, 12, 5
Количество пользователей	50
Количество серверов управления доставкой	1
Количество серверов метаданных	2
Количество узлов каждой подсети облачной структуры	
– ЛВС 1	10
– ЛВС 2	5
– ЛВС 3	3
Длительность моделирования	86400

Моделирование проводилось в системе имитационного моделирования Pilgrim. При помощи конструктора моделей GEM 1.0 была построена визуальная модель. Затем был сгенерирован CPP-файл с исходным текстом программы на языке C++, в который внесены изменения, позволяющие динамически создавать очереди, обслуживающие приборы и управляемые

терминаторы транзактов (демультипликаторы). После построения проекта в среде Microsoft Visual Studio 2010 были произведены эксперименты с полученной моделью. Во время работы модели автоматически строился график задержек в узле «Оч. рез. к СУД» (рис. 3).

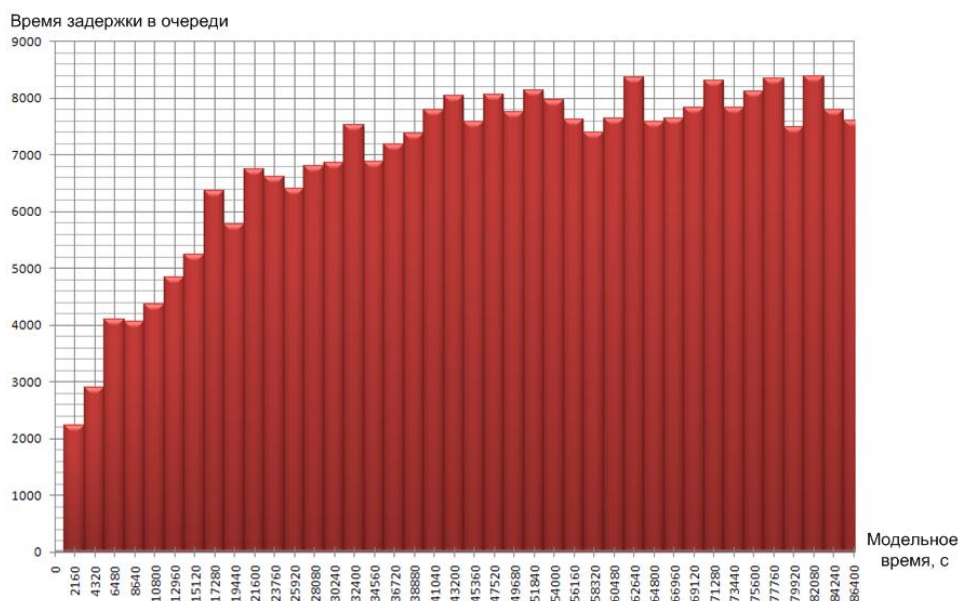


Рис. 3. График задержек в узле «Оч. рез. к СУД»

После окончания процесса моделирования был автоматически сформирован файл, содержащий результаты, представленные в табл. 2:

Табл. 2. Результаты моделирования

№ узла	Наименование узла	Тип узла	Загрузка (%)	M [t] среднее время	C ² [t] квадрат коэф. вар.	Счетчик входов и hold	Кол-во каналов	Оставшиеся транзакты
101	Оч. рез. к СУД	queue	-	6778.32	0.62	156070	1	17833
109	Обслуженные	term	-	16807.24	0.57	33018	0	0
110	Оч. запр. к СУД	queue	-	3481.63	0.64	141531	1	12475
112	Все запросы	creat	-	0.00	1.00	73150	1	0
114	СУД	serv	%=99.3	2.01	0.24	429257	1	1
116	ЛВС 1	serv	%=88.0	8.01	0.08	94900	10	10
120	Очередь СМ	queue	-	90.15	5.90	165490	1	6
121	СМ	serv	%=76.3	2.01	0.24	165484	2	2
122	Оч. служ. к СУД	queue	-	2899.87	0.82	165480	1	3516
131	ЛВС 2	serv	%=93.4	10.02	0.05	40280	5	5
133	ЛВС 3	serv	%=96.9	12.01	0.06	20908	3	3
201	Оч. польз. 0	queue	-	5.76	3.39	1532	1	0
...								
250	Оч. польз. 49	queue	-	5.64	3.52	1530	1	0

301	Польз. 0	serv	%=35.4	19.98	0.06	1532	1	0
...								
350	Польз. 49	serv	%=35.3	19.94	0.06	1530	1	0

Анализ приведенных результатов показывает, что за период моделирования, равный 1 суткам (86400 секунд), в систему поступило 73150 запросов. Каждый пользователь послал в среднем более 1500 запросов и ни один пользователь на момент окончания моделирования не обдумывает очередной запрос. В среднем пользователи потратили на обдумывание запросов около 35% времени работы с ЛВС. Среднее время ожидания запроса и ответа в очереди пользователя составило порядка 6%, а среднее время пребывания в узле приблизительно равно 20 секунд.

Из поступивших 73150 запросов были обработаны только 33018 запросов, остальные находятся в очередях к серверам и узлам. Такая ситуация обусловлена нехваткой вычислительных мощностей серверов управления доставкой, т.к. коэффициент нагрузки составляет 99,3%, причем в очереди находится 12475 запросов, 17833 фрагментов результирующих данных и 3516 необработанных сообщений от серверов метаданных.

Загрузка серверов метаданных составила 76,3%, что говорит об определенном запасе вычислительной мощности. Загрузка узлов подсетей облачной структуры колеблется от 88% до 96,9%, что свидетельствует об отсутствии большого запаса вычислительной мощности узлов.

При анализе состояния очередей особое внимание следует обратить на значение квадрата коэффициента вариации времени задержки. Если оно близко к нулю, время задержки не имело существенного разброса, то есть длина очереди была практически постоянной. Если же значение квадрата коэффициента вариации велико, то это значит, что транзакты приходили в очередь группами. Средний размер группы равен корню из квадрата коэффициента вариации. В данном случае квадрат коэффициента вариации не слишком большой, поэтому можно сделать вывод о том, что время задержек в очередях было приблизительно постоянным.

Таким образом, указанного количества узлов сети произвольной структуры достаточно, чтобы в течение заданного промежутка времени не происходило задержек в обработке запросов, но в то же время не было значительного простоя узлов. Но в ходе анализа результатов моделирования выяснилось, что существует нехватка вычислительных мощностей СУД, и это является узким местом рассматриваемой системы.

Заключение

В рамках данной работы были достигнуты следующие результаты:

1. Рассмотрена модель архитектуры транзакционной системы, предполагающей размещение реплик фрагментов РБД в вычислительной сети с произвольной топологией в рамках облачной структуры.

2. Приведен обобщенный порядок обработки запросов к РБД в сети, построенной по принципу облачной структуры.

3. Предложена модель, описывающая OLTP-систему как разомкнутую сеть массового обслуживания.

4. Проведены эксперименты по определению загруженности системы и сделан анализ их результатов.

Следует отметить, что необходимо создание других узкоспециализированных классов моделей для отдельных задач, которые позволят учитывать специфику современных программно-аппаратных средств.

Литература

1. Черноморов Г.А. Теория принятия решений: Учеб. пособие/ Юж.-Рос. гос. техн. ун-т – 2-е изд. перераб. и доп. – Новочеркасск: Ред. журн. «Изв. вузов. Электромеханика», 2005. – 448 с.
2. Клейнрок Л. Вычислительные системы с очередями. – М.: Мир, 1979. – 600 с.
3. Матвеев В.Ф., Ушаков В.Г. Системы массового обслуживания. – М.: Изд-во МГУ, 1984. – 240 с.