

Введение

Имитационное моделирование становится эффективным методом исследования сложных систем со случайным взаимодействием элементов, таких как транспортные потоки, многоступенчатое промышленное производство, распределенные объекты управления. Принцип имитационного моделирования заключается в том, что поведение системы отображают компьютерной моделью взаимодействия ее элементов во времени и пространстве.

Главная ценность имитационного моделирования состоит в том, что в его основу положена методология системного анализа. Она дает возможность исследовать проектируемую или анализируемую систему по технологии операционного исследования, включая такие взаимосвязанные этапы, как содержательная постановка задачи; разработка концептуальной модели; разработка и программная реализация имитационной модели; оценка адекватности модели и точности результатов моделирования; планирование экспериментов; принятие решений. Благодаря этому имитационное моделирование можно применять как универсальный подход для принятия решений в условиях неопределенности и для оценки влияния трудно формализуемых факторов.

Изучение системы с помощью модели позволяет проверить новые решения без вмешательства в работу реальной системы, растянуть или сжать время функционирования системы, понять сложное взаимодействие элементов внутри системы, оценить степень влияния факторов и выявить узкие места.

Применение имитационного моделирования целесообразно, если:

- проведение экспериментов с реальной системой невозможно или дорого;
- требуется изучить поведение системы при ускоренном или замедленном времени;
- аналитическое описание поведения сложной системы невозможно;
- поведение системы зависит от случайных воздействий внешней среды;
- требуется выявить реакцию системы на непредвиденные ситуации;
- нужно проверить идеи по созданию или модернизации системы;
- требуется подготовить специалистов по управлению реальной системой.

Цели и задачи исследования

Целью выполнения курсовой работы является исследование организации обработки запросов в центре обработки данных. Для этого необходимо решить следующие задачи:

- 1) Изучить предметную область, которую необходимо моделировать. В данной работе – это работа центра обработки данных (ЦОД).
- 2) Выбрать состав модели ЦОД – физические устройства.
- 3) Определить логику взаимодействия этих устройств между собой.
- 4) Построить базовую модель. Эта модель структурно соответствует существующей сети, является базисом для построения модели следующего уровня.
- 5) Выявить недостатки базовой системы с целью их устранения при построении следующей модели.
- 6) Построить модель для экспериментов. Она отличается от базовой модели наличием возможности обрабатывать запросы разной сложности с определённым временем обработки.
- 7) Выявить недостатки второй модели для устранения недостатка – загруженность серверов.
- 8) Построить модель оптимизированной системы. В ней используется сложная маршрутизация, запросы с web-серверов перенаправляются на SQL-серверы и возвращается ответ к web-серверам. Предусмотрено распределение нагрузки на серверы, добавлены для этого дополнительные серверы с аналогичными функциями для работы.
- 9) Выявить недостатки третьей модели. Основной недостаток – оборудование работает непрерывно и никогда не выходит из строя, что не соответствует условиям эксплуатации.
- 10) Построить расширенную модель с добавлением дополнительных линий связи для повторной передачи данных в случае возникновения ошибки. Каждый сервер иногда может выйти из строя, но через некоторое время восстанавливается его рабочее состояние.
- 11) Провести эксперимент структурной оптимизации (вычисление времени обработки всех транзактов на сервере), вывести результаты в сводную таблицу, в которой отражены параметры объектов, входящих в модель.

Дополнительные задачи для достижения цели:

- ознакомление с возможностями системы моделирования SIMIO;
- проведение анализа возможных методов решения поставленной задачи;
- организация экспериментов с моделью;
- обработка результатов моделирования.

Вариант модели 1

Для того чтобы построить первую модель, необходимо выполнить задачи 1, 2, 3. Предметная область для моделирования – работа центра обработки данных. В ЦОД используется клиент-серверная технология обработки запросов. Принцип: пользователь посылает запрос на сервер, где происходит его обработка, сервер формирует ответ, который пройдет тем же путём, как пришёл запрос.

Основная цель построения данной модели: создать базовую модель ЦОД, в которой будет реализована простая передача данных. В дальнейшем эту модель будем использовать, как основу для других более сложных и развитых моделей.

Физические устройства:

а) Терминал – компьютер пользователя, в модели SIMIO – это генератор транзактов (запросов). Подразумевается использование 3-х терминалов для нагрузки системы и реализации обработки на серверах разных типов запросов. Визуальное представление терминала показано на рисунке 1.

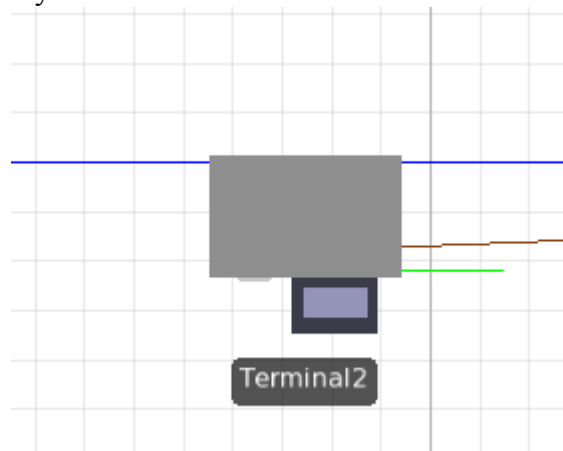


Рисунок 1. Визуальное представление терминала в SIMIO.

б) Коммутатор – устройство для соединения нескольких узлов сети. В модели потребуется один коммутатор для соединения терминалов, генерирующих запросы, второй коммутатор – для соединения двух серверов SQL. Таким образом, в модели сети образуются минимум два сегмента. Коммутатор 1 используется для передачи транзактов от всех терминалов, либо к терминалам. Коммутатор 2 используется для передачи запросов на SQL серверы, либо ответов с них к терминалам. Визуальное представление коммутатора в SIMIO показано на рисунке 2.

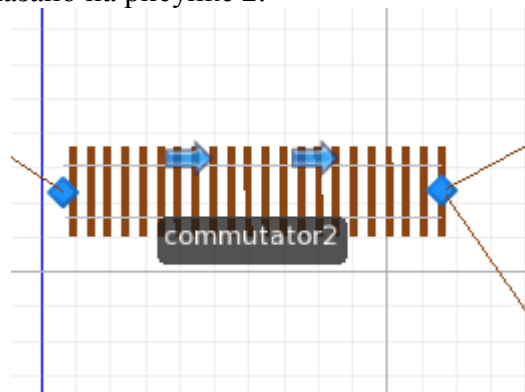


Рисунок 2. Визуальное представление коммутатора в SIMIO.

в) Роутер (маршрутизатор) – специализированное устройство, позволяющее передавать пакеты между различными сегментами сети. В первой модели необходимо, чтобы запросы пришли на обработку на нужный сервер (если веб-запрос, то на веб-сервер, если SQL-запрос, то на SQL-сервер). Это и будет задача роутера. Визуальное представление роутера в SIMIO показано на рисунке 3.

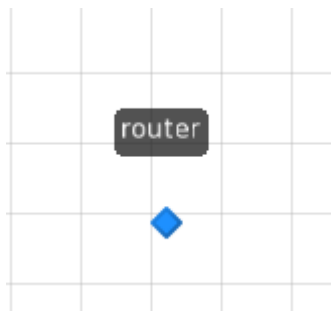


Рисунок 3. Визуальное представление роутера в SIMIO.

г) Сервер – компьютер или специальное компьютерное оборудование, выполняющий определённые сервисные функции. В модели существуют два типа серверов: WEB-сервер и 2 SQL-сервера. Каждый из них обрабатывает запросы с определённых терминалов. Время обработки запроса подчиняется некоторому закону распределения. Визуально представление WEB-сервера показано на рисунке 4а, SQL-сервера на рисунке 4б.

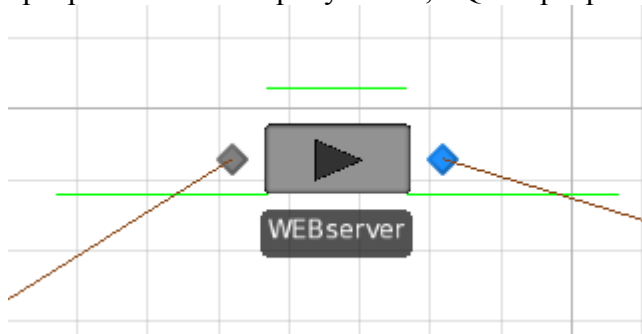


Рис 4а. Визуальное представление WEB-сервера в SIMIO.

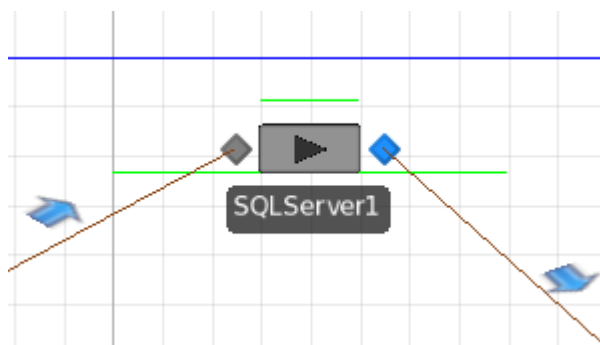


Рисунок 4б. Визуальное представление SQL-сервера в SIMIO.

д) Линии связи – совокупность технических устройств и физической среды, обеспечивающая распространение сигналов от передатчика к приёмнику. В модели служат для соединения остальных физических устройств, обеспечивая тем самым некоторую логику работы системы в целом. Линии связи представлены путями, соединяющие два узла. Визуальное представление линии связи показано на рисунке 5.

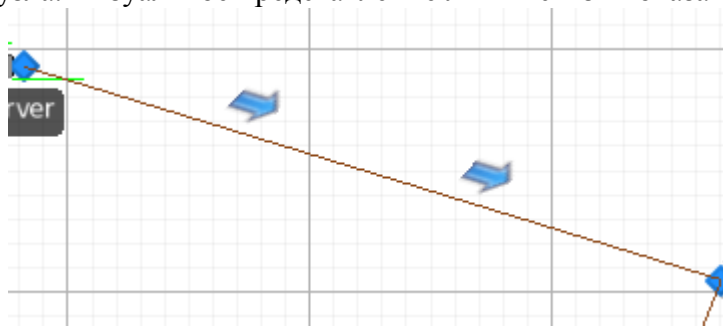


Рисунок 5. Визуальное представление линии связи в SIMIO.

Логика взаимодействия физических устройств.

а) Создаём три терминала, в каждом из них нужно задать параметр, отвечающий за время генерации транзактов. Было решено, что транзакты генерируются не по некоторому закону распределения по времени, а в любой заданный момент времени. Это приближает модель к реалистичной, так как можно открыть журнал запросов, например, и увидеть, в какое время был сделан запрос. То есть появляется возможность задать точное время генерации транзактов. Для этого создаётся таблицы Table 1, Table 2, Table 3. В свойствах этих таблиц задаём тип единицы измерения и сами единицы измерения: время в секундах, соответственно. Задаём имя свойства. Теперь мы получили таблицы, которые можно заполнить вручную временами появления запросов. Таблиц три, так как имеется три терминала, которые генерируют транзакты. Пример такой таблицы отражён на рисунке 6, а пример заполнения свойства таблицы представлен на рисунке 7.

Table1	Table2	Table3	Routing1	R
Время появления (Seconds)				
				1,4
				10,3
				15,7
				25,3
				76,8
				90,7
				100
				106
				108
				130
				164
				187
				200

Рисунок 6. Пример таблицы (лог запросов).

Properties: RealProperty1 (Numeric Property)	
Logic	
Default Value	0.0
Data Format	Real
Unit Type	Time
Default Units	Seconds
Appearance	
Display Name	Время появления
General	
Name	RealProperty1
Description	
Required Value	True

Рисунок 7. Пример заполнения свойства таблицы (лога запросов).

Теперь необходимо определить сами транзакты для каждого терминала. В SIMIO транзакты называют «сущностями». Терминал 1 будет генерировать только WEB-запросы, Терминалы 2 и 3 будут генерировать только SQL-запросы. В свойствах каждого типа транзакта задаётся имя, например «zaproсWEB». Затем в свойствах каждого терминала мы должны указать «тип сущности», которые генерируются данным терминалом, обязательно задать «arrival mode» или вид прибытия транзактов, как лог запросов, то есть в моменты времени, указанные в ранее созданных таблицах. Задаём свойство времени прибытия транзакта с помощью свойства тех же самых таблиц как показано на рисунке 8.

Properties: Terminal1 (Source)	
Entity Arrival Logic	
Entity Type	zaproсWEB
Arrival Mode	Arrival Table
Arrival Time Prop...	Table1.RealProperty
Arrival Events Pe...	1
Arrival Time D...	0.0
Arrival No-Show ...	0
Entities Per Arrival	1
Repeat Arrival P...	False
Stopping Conditions	
Maximum Arrivals	Infinity
Maximum Time	5
Units	Minutes
Stop Event Name	
Table Reference Assignments	
State Assignments	

Рисунок 8. Процедура присваивания сущности «zaproсWEB» к терминалу 1.

На рисунке 8 видно также, что указано условие остановки генерации, то есть через 5 минут после начала работы модели.

б) Далее необходимо реализовать роутинг запросов, чтобы WEB-запросы шли к WEB-серверу, а SQL-запросы к SQL-серверу. Для этого создаём аналог таблиц маршрутизации, только в модели ее компонентами являются непосредственно имена узлов сети. Так как типов запросов три, то и таблиц маршрутизации будет три. Для создания таблиц используем кнопку “Add sequence table”, и заполняем ее именами тех узлов, через которые должен пройти определенный запрос. Далее ко всем выходным узлам сети после роутера добавляем в свойство “Routing logic” в поле “Entity destination” значение «By sequence». То есть, таким образом, дальнейшее продвижение транзактов будет зависеть от тех имён,

которыми заполнены таблицы маршрутизации. Теперь каждой сущности (их визуальное представление показано на рисунке 9) в свойство “Routing logic” в поле “Initial Sequence” добавляем значение – имя таблицы маршрутизации, которая будет использоваться для определения пути сущности в виде ответа до своего терминала (рисунок10). Пример таблицы маршрутизации представлен на рисунке 11.

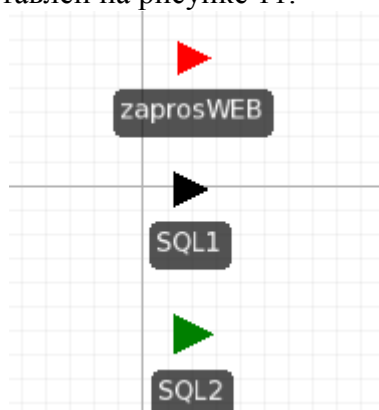


Рисунок 9. Сущности (транзакты) в модели 1.

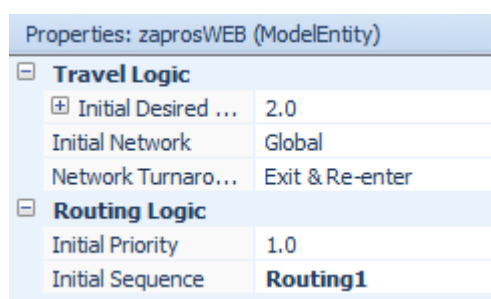


Рисунок 10. Задание маршрута для транзакта “zaprosWEB”.

Table1	Table2	Table3	Routing1	Routing2
Sequence				
TransferNode3				
TransferNode4				
Input@SQLServer1				
TransferNode5				
TransferNode6				
TransferNode7				
TransferNode8				
Input@Terminal22				
*				

Рисунок 11. Пример таблицы маршрутизации в SIMIO.

Примечание: любая модель в SIMIO строится по принципу: вход-обработка-выход. То есть простейшая модель может быть представлена источником (пример: терминал), сервером и стоком (для нашей модели – тоже терминал, играющий роль входного терминала).

В силу этой особенности, практически невозможно организовать перемещение транзакта по тому же пути назад в терминал. Поэтому каждое устройство, узел сети имеет копию, которая выполняет те же функции, что и оригинал. Тогда в модели будут отражены последовательно фазы передачи запроса на сервер и передача ответа к терминалу.

в) Теперь необходимо построить путь прохождения для транзактов. Объединим терминалы коммутатором. В качестве коммутатора будет выступать объект «Конвейер». Он определяется, как объект позволяющий соединить два узла, но отличается от стандартной линии связи (в SIMIO – path) возможностью задать свойство «аккумулятивности». Аккумулятивность дает возможность не останавливать работу конвейера и двигаться транзактам дальше. Если по каким-то причинам сущность не может продвинуться дальше, она встает в очередь и будет пытаться пройти еще раз. Без свойства аккумулятивности работа конвейера останавливается, если сущности прошли до конца конвейера, но им некуда выйти.

г) Роутер реализован обычным «Transfer node». Именно в нем начинают работать таблицы маршрутизации. Можно было реализовать его работу либо разделителем Separator, но он делит транзакт на несколько частей, что не имеет смысла для нас. Можно было собирать запросы со всех терминалов в один «комбайнером», а затем распределять их на свой сервер сепаратором, но тогда модель искусственно усложняется, а, во-вторых, комбайнер будет ждать запросы со всех трёх терминалов и только потом передавать, как единый транзакт.

д) Теперь организовываем работу серверов. Суть настройки заключается в том, чтобы в свойствах серверов в поле «Processing time» задать закон распределения для времени обработки запроса и единицу измерения времени в секундах. Эта процедура отображена на рисунках 12, 13.

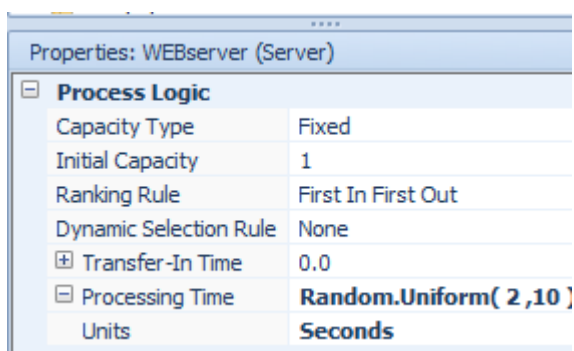


Рисунок 12 . Время обработки WEB-сервера распределено по равномерному закону.

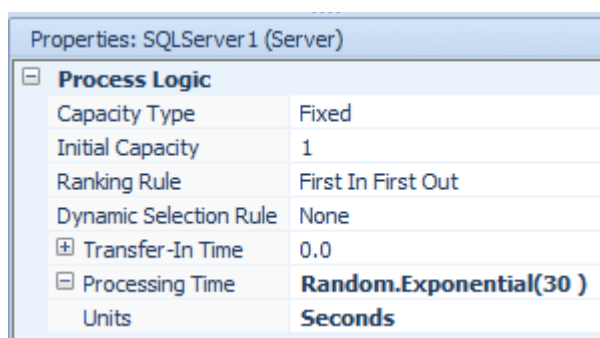


Рисунок 13. Время обработки SQL-сервера 1 распределено по экспоненциальному закону.

е) соединяем линиями связи все указанные выше объекты в единую топологию, учитывая составленные ранее таблицы маршрутизации. Как уже говорилось выше, все объекты удваиваются, чтобы обеспечить обратный путь. Это же правило касается и линий связи.

Результаты работы модели 1

На рисунках 14, 15, 16 представлены результаты работы модели.

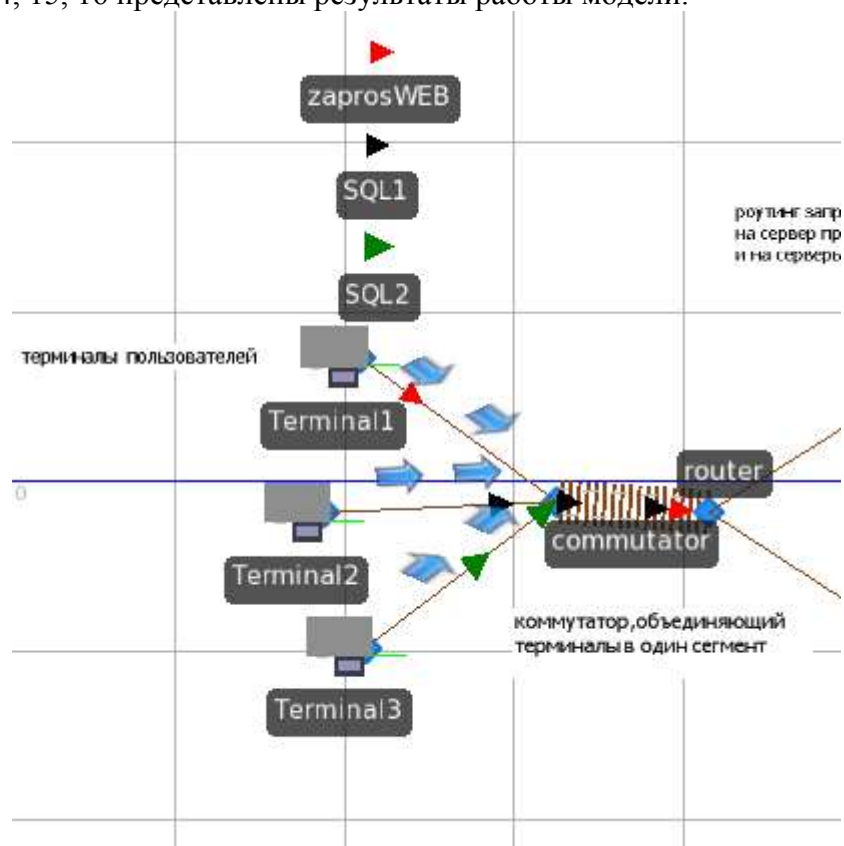


Рисунок14. Генерация транзактов и передача их через коммутатор на роутер.

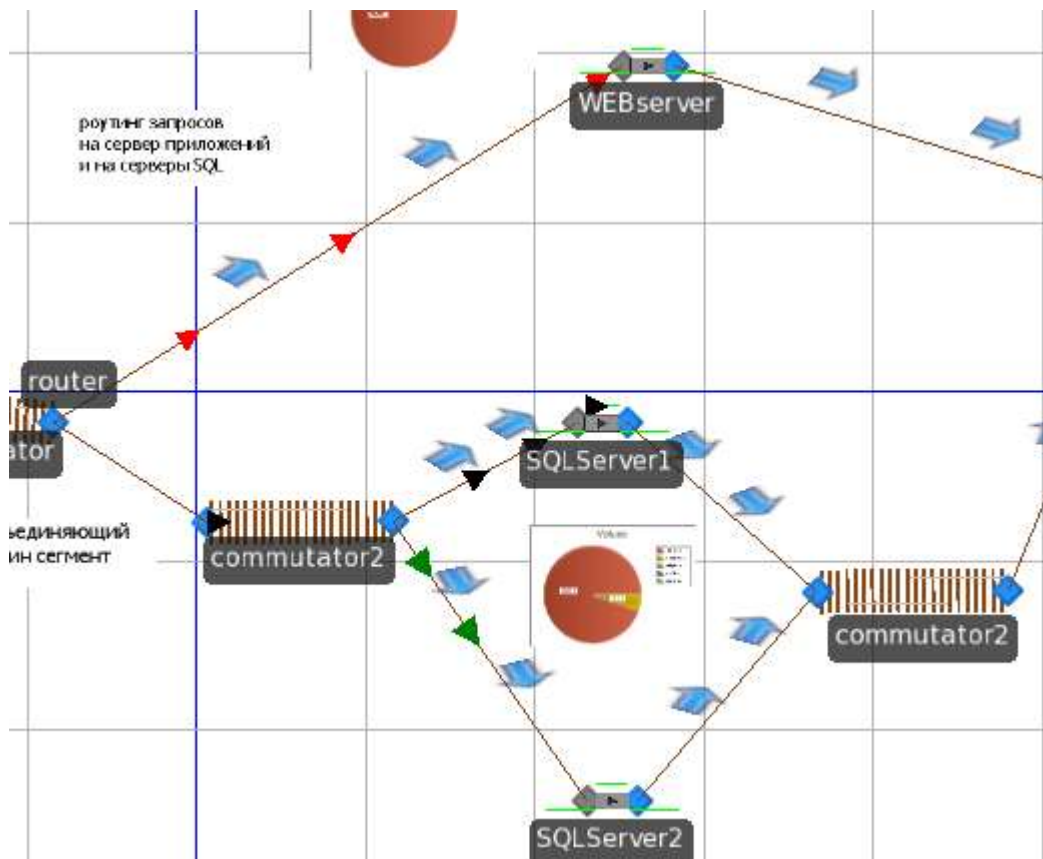


Рисунок 15. Маршрутизация запросов и обработка на серверах.

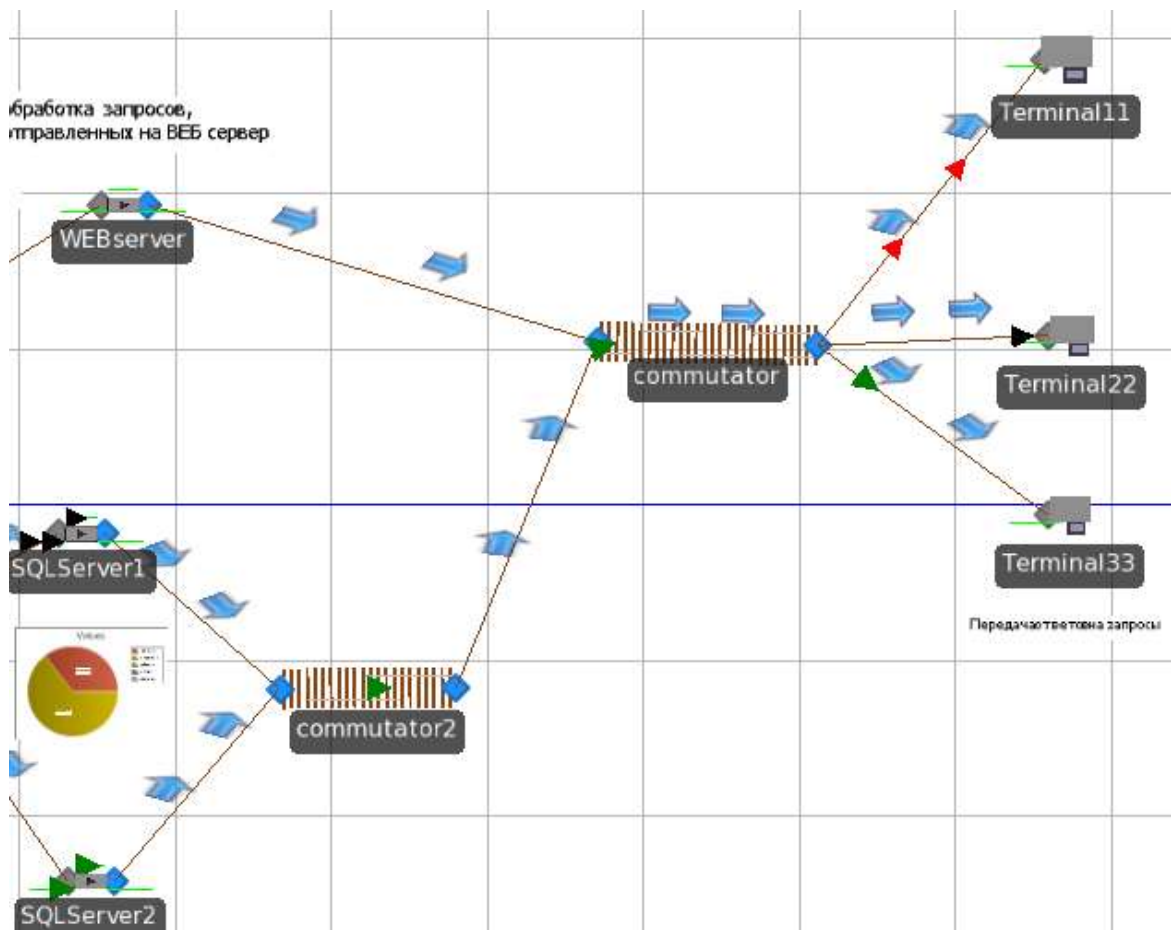


Рисунок 16. Передача ответов по обратному пути соответственным терминалам.

Анализ результатов модели 1

а) Первое, что можно проанализировать по модели – это работа серверов. Для этого можно применить диаграмму статуса. В SIMIO для серверов определена стандартная диаграмма под названием “Resource state”. В этой диаграмме отражено 5 аспектов работы сервера: процент, отражающий время обработки информации, процент ожидания/простоя, процент неуспешных передач информации, процесс блокируемых сущностей (они будут передаваться медленнее остальных), процент неуправляемых сущностей (в моделях не встретилось). Результаты моделирования здесь показали довольно немалую загруженность серверов (процент обработки информации достигал до 70-75 процентов). Диаграмма выглядит следующим образом (рисунок 17).

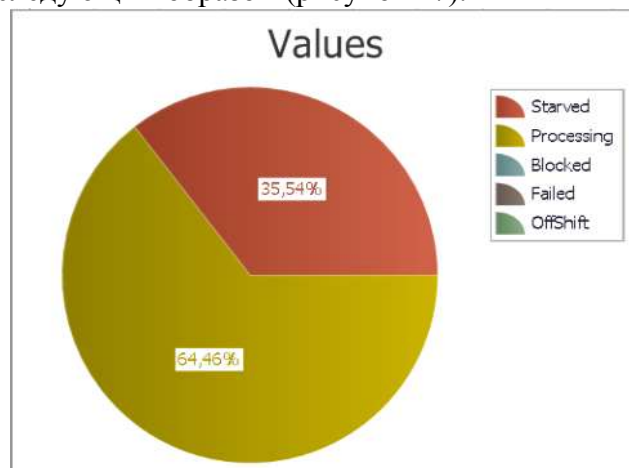


Рисунок 17. Диаграмма, отражающая работу сервера.

б) Второе, что мы можем проанализировать, это работу системы в целом. Это возможно с помощью составления сводной таблицы с параметрами объектов. Для этого необходимо создать эксперимент в SIMIO. Для первой модели управляемые параметры нам не понадобятся. Поэтому сделаем 10 реплик в эксперименте на модели и откроем таблицу результатов. Здесь мы увидим достаточно полную сводную таблицу со средними, максимальными и минимальными значениями параметров. Например, информация по одному коммутатору выглядит так (рисунок 18).

Average						Scenario 1						
Minimum						Average	Minimum	Maximum	Half Width			
Maximum						Scenario 1						
Half Width						Average	Minimum	Maximum	Half Width			
Object Type	Object Name	Data Source	Category	Data Item	Statistic	Average	Minimum	Maximum	Half Width			
Conveyor	Conveyor 1	[Flow]	Throughput	VolumeFlowIn	Total (Cubic ...	0,0000	0,0000	0,0000	0,0000			
				VolumeFlowOut	Total (Cubic ...	0,0000	0,0000	0,0000	0,0000			
				WeightFlowIn	Total (Klogra...	0,0000	0,0000	0,0000	0,0000			
				[Object]	Costs	CapitalCost	Total (USD)	0,0000	0,0000	0,0000	0,0000	
						TotalCost	Total (USD)	0,0000	0,0000	0,0000	0,0000	
				[Travelers]	Content	NumberAccumulated	Average	0,0002	0,0002	0,0002	0,0000	
							Maximum	2,0000	2,0000	2,0000	0,0000	
							NumberOnLink	Average	0,0026	0,0026	0,0026	0,0000
								Maximum	6,0000	6,0000	6,0000	0,0000
						EntryQueue	NumberWaiting	Average	0,0000	0,0000	0,0000	0,0000
									Maximum	1,0000	1,0000	1,0000
					TimeWaiting			Average (Se...	0,0198	0,0198	0,0198	0,0000
							Maximum (Se...	0,5487	0,5487	0,5487	0,0000	
							Minimum (Sec...	0,0000	0,0000	0,0000	0,0000	
				FlowTime	TimeOnLink	Average (Se...	5,2584	5,2584	5,2584	0,0000		
							Maximum (Se...	5,3649	5,3649	5,3649	0,0000	
							Minimum (Sec...	5,2137	5,2137	5,2137	0,0000	
				Throughput		NumberEntered	Total	42,0000	42,0000	42,0000	0,0000	

Рисунок 18. Сводная таблица для коммутатора, объединяющего терминалы в сегмент.

Такую таблицу сложно анализировать, поэтому имеет смысл, рассмотреть только самые важные параметры и для каждого устройства сети составить свою таблицу. В качестве базы данных будет выступать Excel, где можно поддерживать структуру данных, подобную данной таблице, но она содержит только значимые параметры в данной модели. В целом структуру данной БД можно описать так:

- Тип анализируемого объекта.
- Имя объекта.
- Источник данных – то есть откуда идет информация.
- Категория, к которой принадлежит данная информация.
- Рассматриваемый элемент данных.
- Статистика.
- Значение элемента данных.

в) Анализ работы терминалов. На рисунке 19 представлена часть БД, посвященная анализу работы терминалов. Проанализировав данные, можно увидеть, что сгенерировано в целом 42 запроса. Самое большое количество запросов сгенерировал терминал 1, то есть поступило больше всего WEB-запросов от пользователей.

тип объекта	имя объекта	источник данных	категория	элемент данных	статистика	значения
источник	terminal1	выходной буфер	содержимое	число транзактов на	максимум	1
				источнике	минимум	0
			пропускная способность	число входных транзактов	общее	21
	число выходных транзактов	общее		21		
	terminal2	выходной буфер	содержимое	число транзактов на	максимум	1
				источнике	минимум	0
			пропускная способность	число входных транзактов	общее	12
	число выходных транзактов	общее		12		
	terminal3	выходной буфер	содержимое	число транзактов на	максимум	1
источнике				минимум	0	
пропускная способность			число входных транзактов	общее	9	
	число выходных транзактов	общее	9			

Рисунок 19. Таблица с параметрами терминалов.

г) анализ работы коммутаторов. На рисунке 20 представлена часть БД, посвященная анализу работы коммутаторов. В целом, с точки зрения технологии, можно сделать вывод, что чем больше длину конвейера мы выставим, тем, соответственно, больше время передачи. Таким образом, можно регулировать задержку при передаче данных с коммутатора. Зная время передачи транзакта, и число прошедших транзактов через коммутатор, можно вычислить общее эффективное время работы коммутатора.

конвейер	имя коммутатора	элементы конвейера	категория	элемент данных	статистика	значения
конвейер	commutator1	элементы конвейера	содержимое	число транзактов на	максимум	6
				конвейере	минимум	0
			время добега-ния стока	время передачи транзактов через конвейер (с)	максимум	5,3649
				среднее	5,2584	
			пропускная способность	число входных транзактов	общее	42
				число выходных транзактов	общее	42
	commutator2	элементы конвейера	содержимое	число транзактов на	максимум	5
				конвейере	минимум	0
			время добега-ния стока	время передачи транзактов через конвейер (с)	максимум	6,3299
				среднее	6,2682	
			пропускная способность	число входных транзактов	общее	21
				число выходных транзактов	общее	21

Рисунок 20. Таблица с параметрами коммутаторов.

д) анализ работы серверов. На рисунке 21 представлена часть БД, посвященная анализу работы серверов. Проанализировав данные, можно увидеть, что обработано 42 запроса. Ни один из них не был потерян, так как терминалы сгенерировали 42 запроса. Самый загруженный сервер SQL1 сервер (80 процентов обработки). WEB-сервер больше всех стоит в простое, хотя он получил больше всех запросов на обработку. Но это значит, что у него стоит в параметрах самое минимальное время обработки. Это и будет недостатком модели, сервер обрабатывает все запросы подряд, не анализируя их сложность и, соответственно, выделяя нужное время на их обработку.

сервер	SQL1server	ресурсы	структура ресурсов	время обработки транзактов (с)	процент	80			
					минимум	218			
					среднее	364			
					максимум	529			
					время простоя (с)	процент	20		
						минимум	54,5		
						среднее	91		
						максимум	133		
					входной буфер	содержимое	число транзактов на сервере	минимум	5
				максимум			11		
				пропускная способность		число входных транзактов	общее	12	
						число выходных транзактов	общее	12	
				SQL2server	ресурсы	структура ресурсов	время обработки транзактов (с)	процент	60
минимум	100								
среднее	108								
максимум	118								
время простоя (с)	процент	40							
	минимум	67							
	среднее	72							
	максимум	79							
входной буфер	содержимое	число транзактов на сервере	минимум					0	
		максимум	1						
	пропускная способность	число входных транзактов	общее				9		
		число выходных транзактов	общее				9		
WEB-server	ресурсы	структура ресурсов	время обработки транзактов (с)				процент	35	
				минимум	106				
				среднее	121				
				максимум	137				
				время простоя (с)	процент	65			
					минимум	197			
					среднее	225			
					максимум	254			
				входной буфер	содержимое	число транзактов на сервере	минимум	1	
			максимум			2			
			пропускная способность		число входных транзактов	общее	21		
					число выходных транзактов	общее	21		

Рисунок 21. Таблица с параметрами серверов.

е) анализ транзактов, генерируемых в модели. На рисунке 22 представлена часть БД, посвященная анализу существующих сущностей в модели.

сущность (транзакт)	SQL1	множество транзактов	время добегания стока	время нахождения транзакта в системе	минимум	81				
					среднее	192				
					максимум	301				
					SQL2	множество транзактов	время добегания стока	время нахождения транзакта в системе	минимум	47
									среднее	53
									максимум	60
	zaproSWEB	множество транзактов	время добегания стока	время нахождения транзакта в системе	минимум	42				
					среднее	46				
					максимум	53				

Рисунок 22. Таблица с параметрами транзактов.

Самым значимым параметром был выбран параметр «Время нахождения транзакта в системе». Долго ждать ответа на запрос придется пользователю терминала 2. Транзакт может находиться в системе 301 единицу модельного времени. Это обуславливается задержкой обработки на сервере, и возможно, задержкой при передаче через коммутатор.

Выводы по модели 1

- 1) Недостаточное количество запросов (42 транзакта) для модели ЦОД.
- 2) Неравномерная загрузка серверов.
- 3) Все транзакты обрабатываются серверами случайным образом без учета типа запроса.
- 4) Низкая точность моделирования.
- 5) Бесконечные буферы у серверов.
- 6) Построена базовая модель, которую можно расширить полезными дополнениями.
- 7) Отражена работа основных устройств сети.
- 8) Представлена некоторая статистика работы серверов.

Вариант модели 2

Для того чтобы построить расширенную модель на основе базовой модели, необходимо, чтобы были выполнены задачи 1-5.

Основная цель построения данной модели: создать модель ЦОД, в которой будет реализована обработка запросов серверами, исходя из их сложности. То есть будет выделен определённый закон распределения на время обработки запроса данной сложности. Также планируется увеличить количество запросов, поэтому будут более подробно заполнены таблицы запросов.

Физические устройства:

а) Терминал – компьютер пользователя, в модели SIMIO – это генератор транзактов (запросов). Подразумевается использование 3-х терминалов для нагрузки системы и реализации обработки на серверах разных типов запросов. Визуальное представление терминала показано на рисунке 1.

б) Коммутатор – устройство для соединения нескольких узлов сети. В модели потребуется один коммутатор для соединения терминалов, генерирующих запросы, второй коммутатор – для соединения двух серверов SQL. Таким образом, в модели сети образуются минимум два сегмента. Коммутатор 1 используется для передачи транзактов от всех терминалов, либо к терминалам. Коммутатор 2 используется для передачи запросов на SQL серверы, либо ответов с них к терминалам. Визуальное представление коммутатора в SIMIO показано на рисунке 2.

в) Роутер – специализированное устройство, позволяющее передавать пакеты между различными сегментами сети. В модели необходимо, чтобы запросы пришли на обработку на нужный сервер (если веб-запрос, то на веб-сервер, если SQL-запрос, то на SQL-сервер). Это и будет задача роутера. Визуальное представление роутера в SIMIO показано на рисунке 3.

г) Сервер – в модели существуют два типа серверов: WEB-сервер и 2 SQL-сервера. Каждый из них обрабатывает запросы с определенных терминалов. Время обработки запроса подчиняется некоторому закону распределения. WEB-сервер будет обрабатывать три типа запросов (по сложности). Для каждого типа запроса определено свое время обработки. Визуально представление WEB-сервера показано на рисунке 4а, SQL-сервера на рисунке 4б.

д) Линии связи – совокупность технических устройств и физической среды, обеспечивающая распространение сигналов от передатчика к приемнику. В модели служат для соединения остальных физических устройств, обеспечивая тем самым некоторую логику работы системы в целом. Линии связи представлены путями, соединяющие два узла. Визуальное представление линии связи показано на рисунке 5.

Логика взаимодействия физических устройств

а) Создаем три терминала, в каждом из них нужно задать параметр, отвечающий за время генерации транзактов. Было решено, что транзакты генерируются не по некоторому закону распределения по времени, а в любой заданный момент времени. Это приближает модель к реальной жизни, так как можно открыть журнал запросов, например, и увидеть, в какое время был сделан запрос. То есть появляется возможность задать точное время генерации транзактов. Для этого создаётся таблицы Time WEB, Time Sql1, Time Sql2. В свойствах этих таблиц задаём тип единицы измерения и сами единицы измерения: время в

секундах, соответственно. Задаём времена появления транзактов через интервал времени, который будет меньше, чем интервал времени между появлениями запросов в базовой модели. Теперь мы получили таблицы, которые можно заполнить вручную временами появления запросов. Таблиц три, так как имеется три терминала, которые генерируют транзакты. Пример новой изменённой таблицы показан на рисунке 23.

Time WEB	Time Sql1	Time Sql2
Время появления (Seconds)		
		1,4
		2,7
		3,4
		5,8
		10
		12
		13,2
		17,6
		20
		23
		26
		28,9
		34
		37
		43,2
		46
		50
		60
		69
		75

Рисунок 23. Пример заполненной таблицы Time WEB.

Теперь необходимо определить сами транзакты для каждого терминала. Терминал 1 будет генерировать только WEB-запросы. Именно на этой стадии необходимо задать переменную, которая будет отвечать за время обработки для конкретной сущности. Создаём три вида по сложности запросов: ZaproWEB1, ZaproWEB2, ZaproWEB3. Затем в свойствах терминала 1 необходимо указать название переменной, или конкретней сказать, ссылки на таблицу, где указано, какому типу сущности какой закон распределения по времени обработки соответствует. Это значение задаётся в поле «Тип сущности» свойств терминала 1. Терминалы 2 и 3 будут генерировать только SQL-запросы. В их свойствах мы тоже должны указать «тип сущности», которые генерируются данным и терминалами. Во всех терминалах обязательно задаём «arrival mode» или вид прибытия транзактов, как лог запросов, то есть в моменты времени, указанные в ранее созданных таблицах. Задаём свойство времени прибытия транзакта с помощью свойства тех же самых таблиц. На рисунке 24 отражена процедура заполнения полей свойства терминала 1, который будет генерировать WEB-запросы различной сложности.

Properties: Terminal1 (Source)	
Entity Arrival Logic	
Entity Type	ObrabotkaWeb.EntityInstanceProperty1
Arrival Mode	Arrival Table
Arrival Time Property	TimeWEB.RealProperty1
Arrival Events Per Time Slot	1
⊕ Arrival Time Deviation	0.0
Arrival No-Show Probability	0
Entities Per Arrival	1
Repeat Arrival Pattern	False

Рисунок 24. Значения свойств для транзактов терминала 1.

б) Далее необходимо реализовать роутинг запросов, чтобы WEB-запросы шли к WEB-серверу, а SQL-запросы к SQL-серверу. Для этого создаём некоторый аналог таблиц маршрутизации, только ее компонентами являются непосредственно имена узлов сети. Так как типов запросов три, то и таблиц маршрутизации будет три. Для создания таблиц используем кнопку “Add sequence table”, и заполняем ее именами тех узлов, через которые должен пройти определённый запрос. Далее ко всем выходным узлам сети после роутера добавляем в свойство “Routing logic” в поле “Entity destination” значение «By sequence». То есть, таким образом, дальнейшее продвижение транзактов будет зависеть от тех имён, которыми заполнены таблицы маршрутизации. Но это не все, каждой сущности (их визуальное представление показано на рисунке 25) в свойство “Routing logic” в поле “Initial Sequence” добавляем значение – имя таблицы маршрутизации, которая будет использоваться для определения пути сущности в виде ответа до своего терминала. (рисунок10). Пример таблицы маршрутизации представлен на рисунке 11.

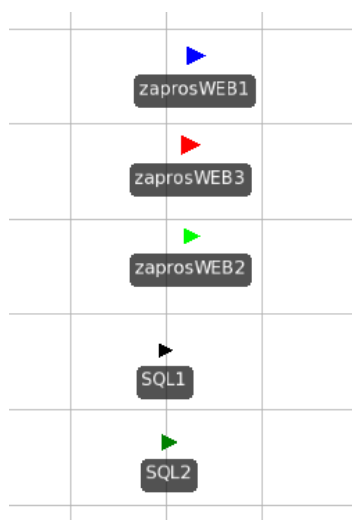


Рисунок 25. Сущности (транзакты) в модели 2.

в) Теперь необходимо построить путь прохождения для транзактов. Объединим терминалы коммутатором. В качестве коммутатора будет выступать объект «Конвейер». Определяется, как объект позволяющий соединить два узла, но отличается от стандартной линии связи возможностью задать свойство «аккумулятивности».

г) Роутер реализован обычным «Transfer node». Именно с него начинают работать таблицы маршрутизации.

д) Теперь организовываем работу серверов. Суть заключается в том, чтобы в свойстве сервера в поле «Processing time» определить ссылку на столбец таблицы, в котором заданы

законы распределения для времени обработки запроса и единицу измерения времени в секундах. В соседних столбцах таблицы отражены имя сервера и имена транзактов всех сложностей. Каждому запросу поставлен в соответствие свой закон распределения для времени его обработки. Эта таблица показана на рисунке 26. Заполненные поля свойств WEB-сервера показаны на рисунке 27.

Time WEB	Time Sql1	Time Sql2	Routing1	Routing2	Routing3	Obrabotka Web
	Processing time (Seconds)	Server name	Entity Instance Property1			Integer Property1
▶	Random.Uniform(2 , 10)	WEBserver	zaproSWEB1			1
	Random.Triangular (1, 3, 8)	WEBserver	zaproSWEB2			2
	Random.Exponential (10)	WEBserver	zaproSWEB3			3
*						

Рисунок 26 . Таблица соответствия типа запроса по сложности времени обработки на WEB-сервере.

Properties: WEBserver (Server)	
Process Logic	
Capacity Type	Fixed
Initial Capacity	1
Ranking Rule	First In First Out
Dynamic Selection Rule	None
⊕ Transfer-In Time	0.0
Processing Time	➡ ObrabotkaWeb.ExpressionProperty1
Buffer Capacity	
Input Buffer	1
Output Buffer	Infinity

Рисунок 27. Свойства WEB-сервера.

Примечание: всем серверам было поставлено ограничение в буфере на возможное количество находящихся там транзактов.

е) соединяем линиями связи все указанные выше объекты в единую топологию, учитывая составленные ранее таблицы маршрутизации. Как уже говорилось выше, все объекты удваиваются, чтобы обеспечить обратный путь. Это же правило касается и линий связи.

Результаты работы модели 2

На рисунках 28, 29, 30 представлены результаты работы модели.

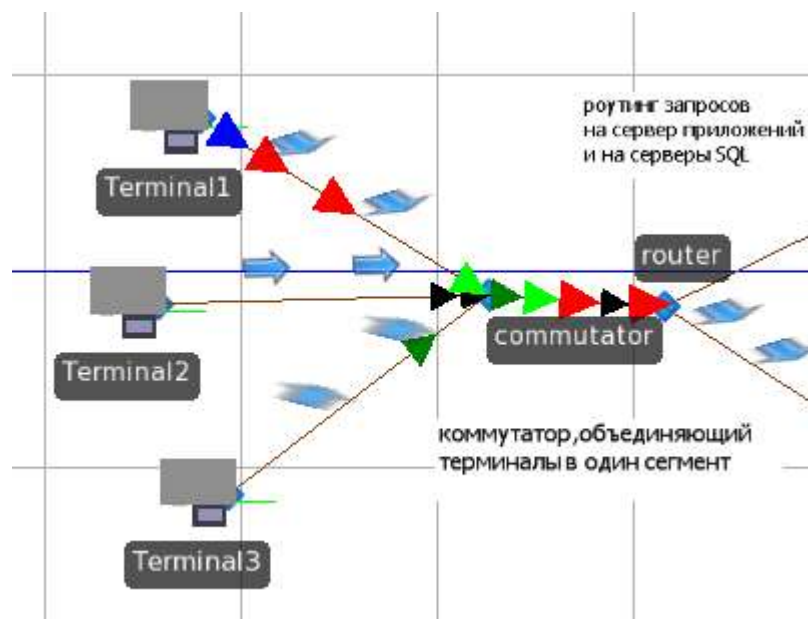


Рисунок 28. Генерация большего количества транзактов и передача их через коммутатор на роутер.

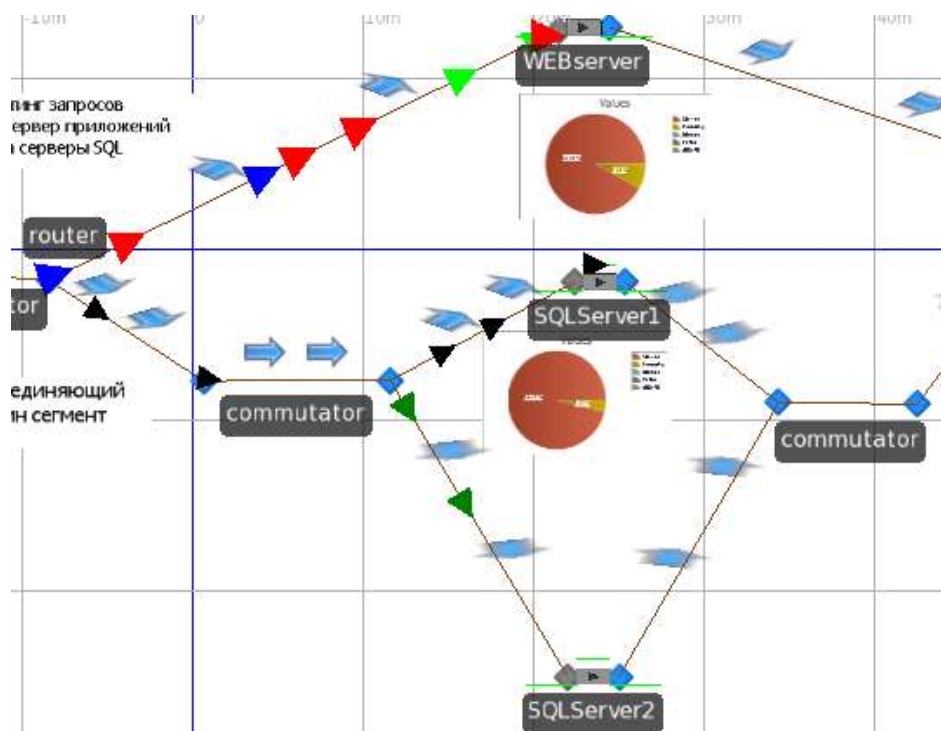


Рисунок 29. Роутинг запросов и обработка их на соответствующих серверах.

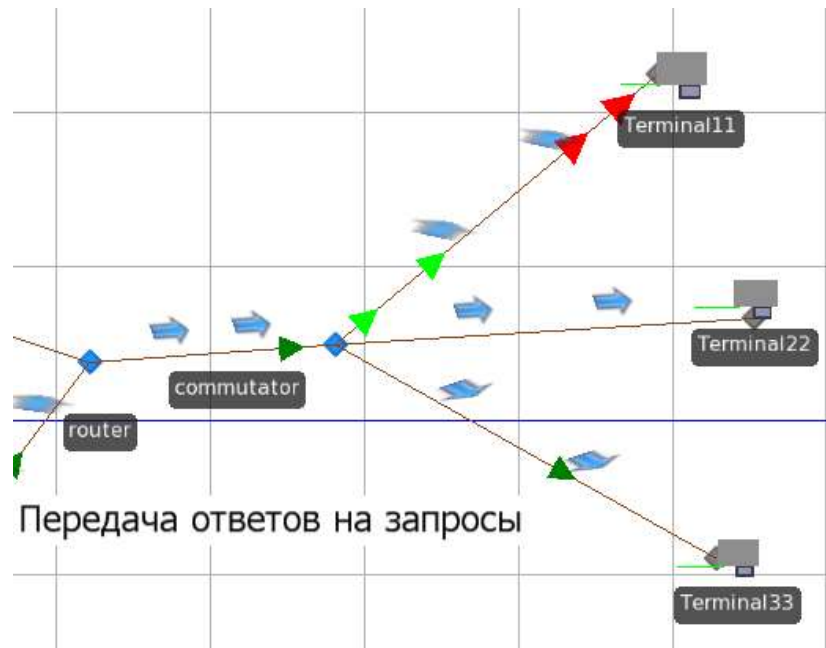


Рисунок 30. Передача ответов по обратному пути соответственным терминалам.

Анализ результатов модели 2

а) Анализ работы терминалов. На рисунке 31 представлена часть БД, посвященная анализу работы терминалов. Проанализировав данные, можно увидеть, что сгенерировано в целом 109 запросов. Самое большое количество запросов сгенерировали терминал 1 и 2, то есть поступило больше всего WEB-запросов от пользователей, как и SQL-запросов от пользователей терминала 2.

тип объекта	имя объекта	источник данных	категория	элемент данных	статистика	значения
источник	terminal1	выходной буфер	содержимое	число транзактов на источнике	максимум	1
				минимум	0	
			пропускная способность	число входных транзактов	общее	43
	число выходных транзактов	общее		43		
	terminal2	выходной буфер	содержимое	число транзактов на источнике	максимум	1
				минимум	0	
			пропускная способность	число входных транзактов	общее	43
	число выходных транзактов	общее		43		
	terminal3	выходной буфер	содержимое	число транзактов на источнике	максимум	1
минимум				0		
пропускная способность			число входных транзактов	общее	23	
	число выходных транзактов	общее	23			

Рисунок 31. Таблица с параметрами терминалов.

б) анализ работы коммутаторов. На рисунке 32 представлена часть БД, посвященная анализу работы коммутаторов. Коммутаторы стали более нагруженными, через первый коммутатор проходит 109 запросов. Если умножить это значение на время передачи транзакта через коммутатор, то получим приблизительно значение 688 секунд, что для данной модели достаточно большое время эффективной работы.

конвейер	commutator1	элементы конвейера	содержимое	число транзактов на конвейере	максимум	5
					минимум	0
			время добегания стока	время передачи транзактов через конвейер (с)	максимум	6,3299
					среднее	6,2682
			пропускная способность	число входных транзактов	общее	109
	число выходных транзактов	общее		109		
	commutator2	элементы конвейера	содержимое	число транзактов на конвейере	максимум	5
					минимум	0
			время добегания стока	время передачи транзактов через конвейер (с)	максимум	6,3299
					среднее	6,2682
пропускная способность			число входных транзактов	общее	66	
	число выходных транзактов	общее	66			

Рисунок 32. Таблица с параметрами коммутаторов.

в) анализ работы серверов. На рисунке 33 представлена часть БД, посвященная анализу работы серверов. Проанализировав данные, можно увидеть, что обработано 109 запросов. Ни один из них не был потерян, так как терминалы сгенерировали 109 запросов. Нагрузка на серверы распределена равномерно, за некоторым исключением: на SQL-сервере 1 максимальное время обработки транзактов достигает 601 секунды. Процент обработки у серверов выше, чем процент простоя. Однако с увеличением количества запросов и типов их сложности WEB-сервер стал более загруженным сервером в данной модели, чем в первой.

сервер	SQL1server	ресурсы	структура ресурсов	время обработки транзактов (с)	процент	60		
					минимум	143		
					среднее	336		
					максимум	602		
				время простоя (с)	процент	40		
					минимум	95		
					среднее	224		
					максимум	401		
				входной буфер	содержимое	число транзактов на сервере	минимум	4
							максимум	11
	пропускная способность	число входных транзактов	общее		43			
		число выходных транзактов	общее		43			
	SQL2server	ресурсы	структура ресурсов		время обработки транзактов (с)	процент	60	
						минимум	100	
				среднее		109		
				максимум		116		
				время простоя (с)	процент	40		
					минимум	67		
					среднее	73		
					максимум	77		
входной буфер	содержимое	число транзактов на сервере	минимум	0				
			максимум	1				
	пропускная способность	число входных транзактов	общее	23				
		число выходных транзактов	общее	23				

	WEB-server	ресурсы	структура ресурсов	время обработки транзактов (с)	процент		60
					минимум		100
					среднее		160
					максимум		230
				время простоя (с)	процент		40
					минимум		67
					среднее		107
					максимум		153
		входной буфер	содержимое	число транзактов на сервере	минимум		0
					максимум		0
			пропускная способность	число входных транзактов	общее		43
				число выходных транзактов	общее		43

Рисунок 33. Таблица с параметрами серверов.

г) анализ транзактов, генерируемых в модели. На рисунке 34 представлена часть БД, посвященная анализу существующих сущностей в модели.

сущность (транзакт)	SQL1	множество транзактов	время добегания стока	время нахождения транзакта в системе (с)	минимум		108
					среднее		220
					максимум		313
	SQL2	множество транзактов	время добегания стока	время нахождения транзакта в системе (с)	минимум		86
					среднее		92
					максимум		97
	zaproSWEB1	множество транзактов	время добегания стока	время нахождения транзакта в системе (с)	минимум		71
					среднее		76
					максимум		83
	zaproSWEB2	множество транзактов	время добегания стока	время нахождения транзакта в системе (с)	минимум		68
					среднее		75
					максимум		90
	zaproSWEB3	множество транзактов	время добегания стока	время нахождения транзакта в системе (с)	минимум		68
					среднее		81
					максимум		98

Рисунок 34. Таблица с параметрами транзактов.

Самым значимым параметром был выбран параметр «Время нахождения транзакта в системе». В целом, практически все ответы приходят за одинаковое время. Долго ждать ответа на запрос придется пользователю терминала 1. Транзакт может максимально находиться в системе 313 единиц модельного времени. Это обуславливается задержкой обработки на сервере, возможно, задержкой при передаче через коммутатор.

Выводы по модели 2

- 1) Смоделирована обработка запросов различной сложности с соответственным временем обработки.
- 2) Увеличено количество запросов в сети.
- 3) Стабилизирован процент обработки информации на всех серверах.
- 4) Загруженность WEB-сервера.
- 5) Пользователи напрямую практически никогда не обращаются на SQL-сервер. Требуется доработка: перенаправление WEB-запросов на SQL-сервер и получение ответа от него.
- 6) Простая топология сети, требуется больше физических устройств.

Вариант модели 3

Для того чтобы построить модель с развитой архитектурой на основе расширенной базовой модели, необходимо, чтобы были выполнены задачи 1-7.

Основная цель построения данной модели: проверить архитектуру ЦОД, добавив дополнительные возможности: наличие новых задач (файл-серверы), реализация логики «WEB-сервер → SQL-сервер → WEB-сервер → ответ терминалу 1», появление дополнительных WEB-серверов и SQL-серверов для распределения загрузки. Провести сравнительный эксперимент с различными законами распределения времени обработки серверов.

Физические устройства:

а) Терминал – компьютер пользователя, в модели SIMIO – это генератор транзактов (запросов). Подразумевается использование 3-х терминалов для нагрузки системы и реализации обработки на серверах разных типов запросов. Визуальное представление терминала показано на рисунке 1.

б) Коммутатор – устройство для соединения нескольких узлов сети. В модели потребуется один коммутатор для соединения терминалов, генерирующих запросы, второй коммутатор – для соединения двух файл-серверов, еще один соединяет в сегмент три WEB-сервера и последний коммутатор соединяет в сегмент три SQL-сервера. Таким образом, в модели сети образуются минимум четыре сегмента. Коммутатор 3 используется для передачи транзактов от всех терминалов, либо к терминалам. Коммутатор 2 используется для передачи запросов на файл-серверы, либо ответов с них к терминалам. Коммутатор 1 требуется для передачи данных к WEB-серверам от терминалов и для передачи обработанных запросов с WEB-серверов к SQL-серверам. Коммутатор 4 используется для передачи данных к SQL-серверам от WEB-серверов и для передачи обработанных запросов с SQL-серверов к WEB-серверам. Визуальное представление коммутатора в SIMIO показано на рисунке 2.

в) Роутер – специализированное устройство, позволяющее передавать пакеты между различными сегментами сети. В модели необходимо, чтобы запросы пришли на обработку на нужный сервер (если веб-запрос, то на веб-сервер, если запрос на файл-сервер, то на File-сервер). Это и будет задача роутера. Визуальное представление роутера в SIMIO показано на рисунке 3.

г) Сервер – в этой модели существуют три типа серверов: 3 WEB-сервера, 3 SQL-сервера и 2 файл-сервера. Каждый из них обрабатывает запросы с определенных терминалов. Время обработки запроса подчиняется некоторому закону распределения. WEB-сервер будет обрабатывать три типа запросов (по сложности). Для каждого типа запроса определено свое время обработки. Визуально представление WEB-сервера показано на рисунке 4а, SQL-сервера на рисунке 4б.

д) Линии связи – совокупность технических устройств и физической среды, обеспечивающая распространение сигналов от передатчика к приемнику. В модели служат для соединения остальных физических устройств, обеспечивая тем самым некоторую логику работы системы в целом. Линии связи представлены путями, соединяющие два узла. Визуальное представление линии связи показано на рисунке 5.

Логика взаимодействия физических устройств

а) Создаём три терминала, в каждом из них нужно задать параметр, отвечающий за время генерации транзактов. Было решено, что транзакты генерируются не по некоторому закону распределения по времени, а в любой заданный момент времени. Это приближает модель к реальной жизни, так как можно открыть журнал запросов, например, и увидеть, в какое время был сделан запрос. То есть появляется возможность задать точное время генерации транзактов. Для этого создаём таблицы Time WEB, Time Sql1, Time Sql2. В свойствах этих таблиц задаём тип единицы измерения и сами единицы измерения: время в секундах, соответственно. Задаём времена появления транзактов через интервал времени, который будет меньше, чем интервал времени между появлениями запросов в базовой модели. Задаём имя свойства. Теперь мы получили таблицы, которые можно заполнить вручную временами появления запросов. Таблиц три, так как имеется три терминала, которые генерируют транзакты. Пример изменённой таблицы показан на рисунке 23.

Теперь необходимо определить сами транзакты для каждого терминала. Терминал 1 будет генерировать только WEB-запросы. Именно на этой стадии необходимо задать переменную, которая будет отвечать за время обработки для конкретной сущности. Создаём три вида по сложности запросов: ZaproWEB1, ZaproWEB2, ZaproWEB3. Затем в свойствах терминала 1 необходимо указать название переменной, или конкретней сказать, ссылки на таблицу, где указано, какому типу сущности какой закон распределения по времени обработки соответствует. Это задаётся в поле «Тип сущности» свойств терминала 1. Терминалы 2 и 3 будут генерировать только SQL-запросы. В их свойствах мы тоже должны указать «тип сущности», которые генерируются данным и терминалами. Во всех терминалах обязательно задаём «arrival mode» или вид прибытия транзактов, как лог запросов, то есть в моменты времени, указанные в ранее созданных таблицах. И, наконец, задаём свойство времени прибытия транзакта с помощью свойства тех же самых таблиц. На рисунке 24 отражена процедура заполнения полей свойства терминала 1, который будет генерировать WEB-запросы различной сложности.

б) Далее необходимо реализовать роутинг запросов, чтобы WEB-запросы шли к WEB-серверу, а FS-запросы к файл-серверам. Для этого создаём некоторый аналог таблиц маршрутизации, только ее компонентами являются непосредственно имена узлов сети. Так как типов запросов три, то и таблиц маршрутизации будет три. Для создания таблиц используем кнопку “Add sequence table”, и заполняем ее именами тех узлов, через которые должен пройти определённый запрос.

Примечание: Таблицы в третьей модели будут реструктурированы, для путей с файл-серверами добавляются в столбцы несколько новых узлов, так как архитектура сети усложнилась. Для WEB-серверов и SQL-серверов из таблицы маршрутизации выбираются узлы, через которые нужно пройти, но там не прописан полный путь, так как на полном пути нам понадобятся другие типы маршрутизации, используемые в SIMIO.

Каждой сущности (их визуальное представление показано на рисунке 25) в свойство “Routing logic” в поле “Initial Sequence” добавляем значение – имя таблицы маршрутизации, которая будет использоваться для определения пути сущности в виде ответа до своего терминала (рисунок 10). Пример новой таблицы маршрутизации представлен на рисунке 35.

Sequence
▶ TransferNode3
TransferNode4
Input@FileServer2
TransferNode7
TransferNode8
TransferNode5
TransferNode24
TransferNode6
Input@Terminal33
*

Рисунок 35. Пример таблицы маршрутизации в модели 3.

в) Теперь необходимо построить путь прохождения для транзактов. Объединим терминалы коммутатором. В качестве коммутатора будет выступать объект «Конвейер».

г) Роутер реализован обычным «Transfer node», но с таблицами маршрутизации.

д) Теперь организовываем работу серверов. Вся суть заключается в том, чтобы свойство «Processing time» можно было задавать полем контроля в эксперименте. Для этого с помощью вкладки “Definitions” создаем свойство «Processing time» тип объекта – выражение, задаем единицы измерения – секунды, в стандартном значении – задаем любой закон распределения, а условие выбора закона – equal. Таким образом, теперь в поле контроля эксперимента (смотри анализ данных) можно задавать любые существующие законы распределения и анализировать, как это влияет на обработку информации в целом. Процедура определения нового свойства Processing time показана на рисунках 36, 37.

Name	Object Type	Display Name
> Properties (Inherited)		
> WorkDayExceptions.Properties (Inherited)		
> WorkPeriodExceptions.Properties (Inherited)		
▼ Properties		
WEB	Expression Property	WEB
ProcessingTime	Expression Property	ProcessingTime

Рисунок 36 . Создание нового свойства.

Properties: ProcessingTime (Expression Property)	
Logic	
Default Value	Random.Triangular(.1,.2,.3)
Switch Property Name	
Switch Condition	Equal
Switch Value	
Candidate References	False
Unit Type	Time
Default Units	Seconds
Appearance	
Display Name	ProcessingTime
Category Name	General
Category Expanded	False
Parent Property Name	
General	
Name	ProcessingTime
Description	

Рисунок 37. Определение значений созданного свойства.

е) соединяем линиями связи все указанные выше объекты в единую топологию, учитывая составленные ранее таблицы маршрутизации. В этой модели все объекты удваиваются, чтобы обеспечить независимый обратный путь. Это же правило касается и линий связи.

Результаты работы модели 3

На рисунках 38-43 представлены результаты работы модели.

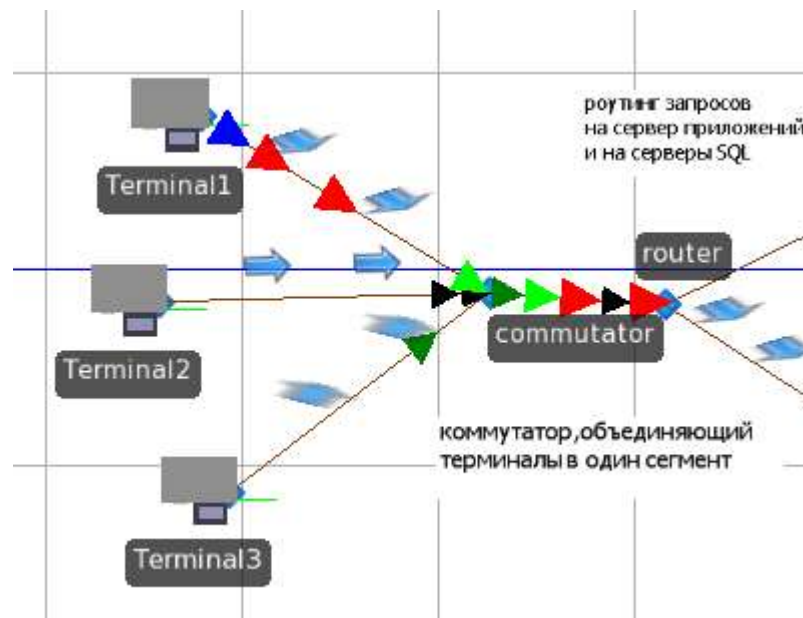


Рисунок 38. Генерация транзактов и передача их через коммутатор на роутер.

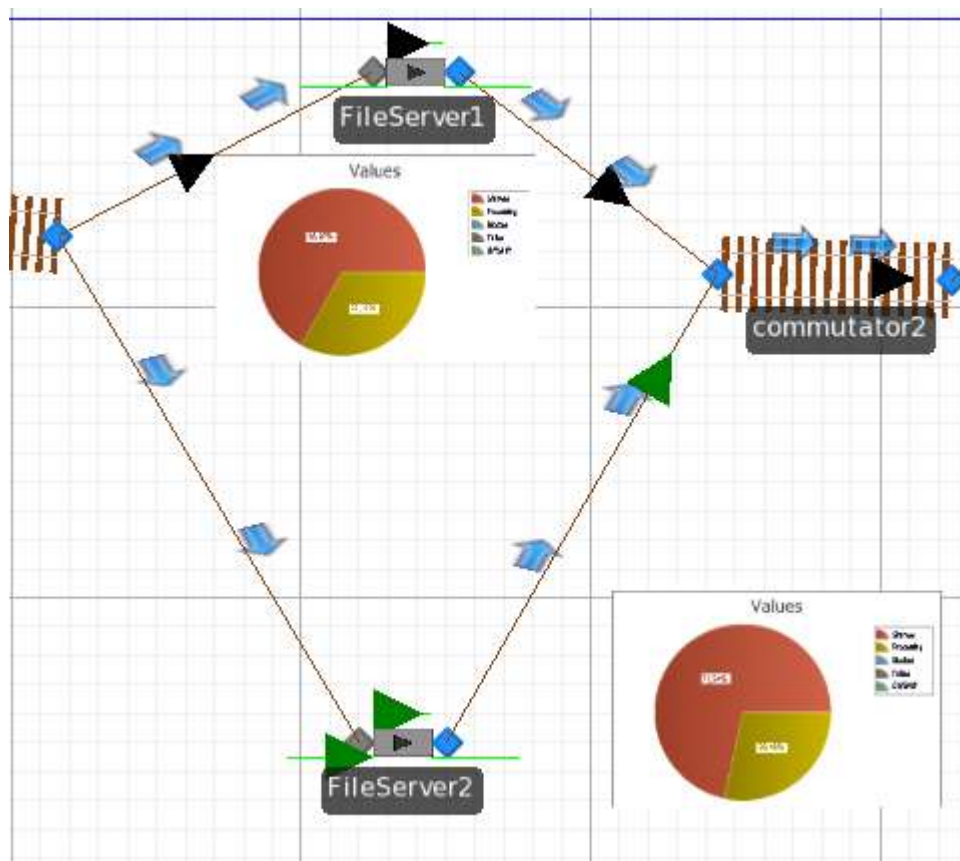


Рисунок 39. Работа файл-серверов.

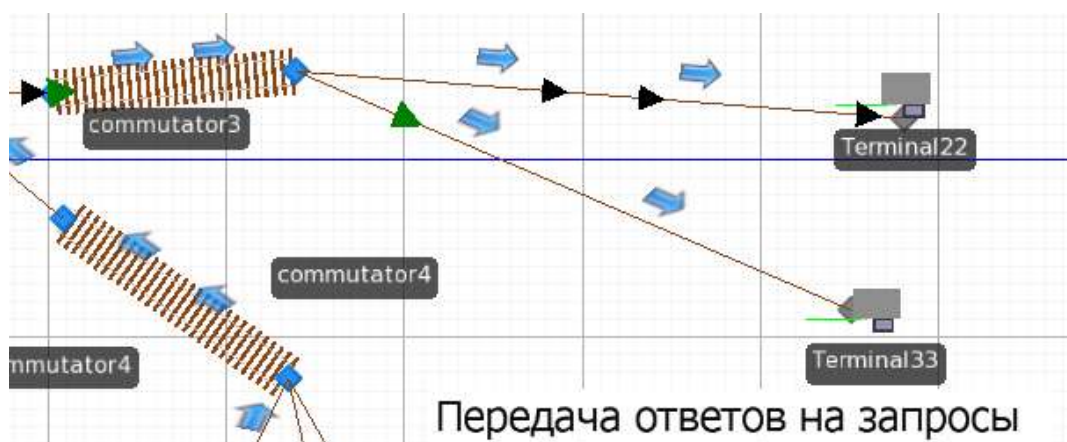


Рисунок 40. Передача ответов с файл-серверов терминалам 2 и 3.

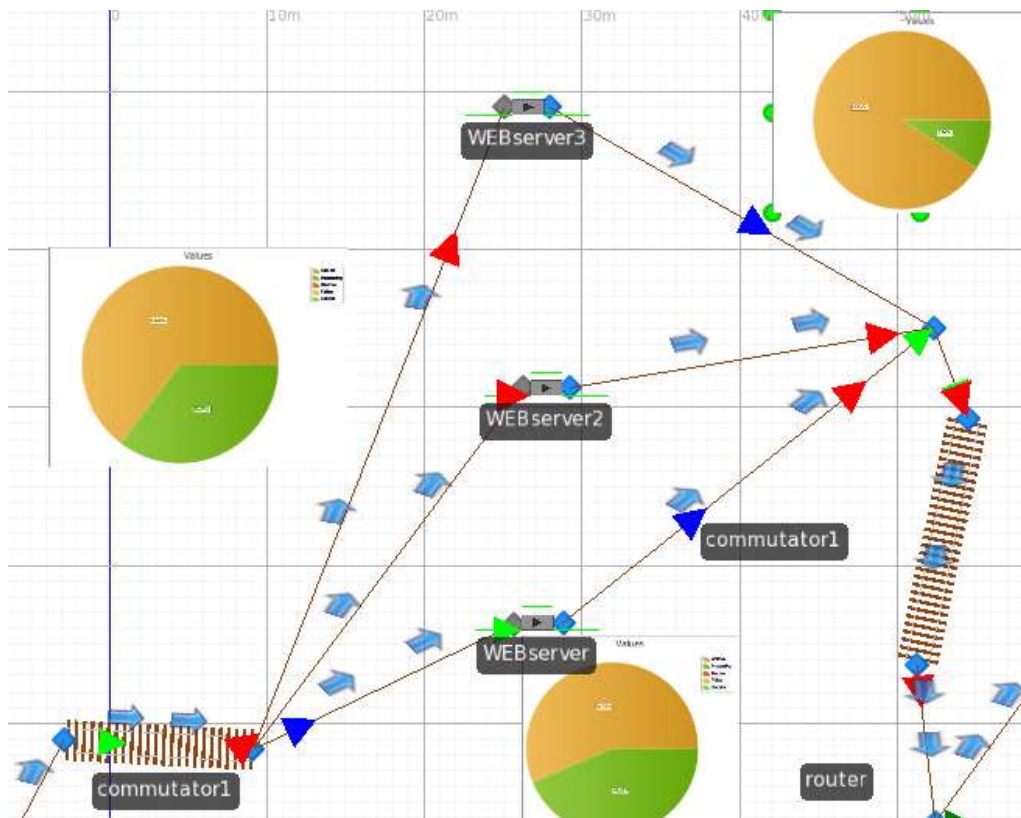


Рисунок 41. Работа WEB-серверов.

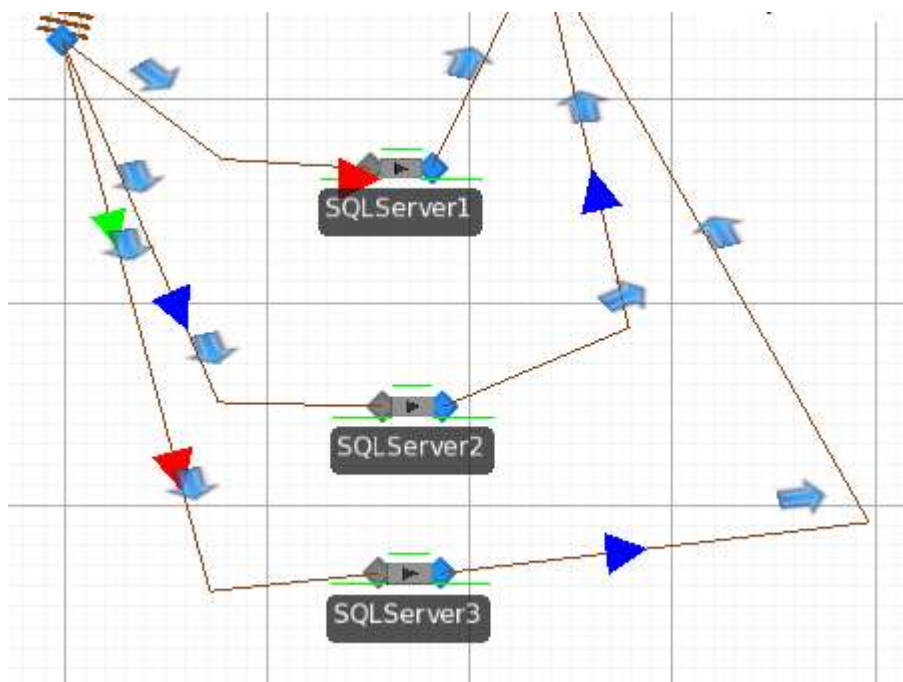


Рисунок 42. Работа SQL-серверов.

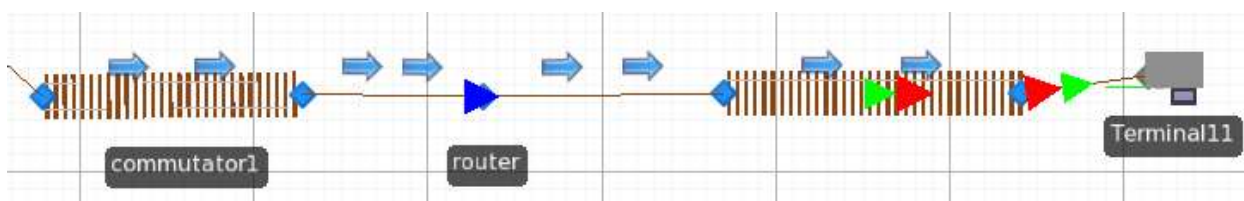


Рисунок 43. Передача ответов от WEB-серверов терминалу 1.

Анализ результатов модели 3

а) Эксперимент с моделью 3. В поле «Контроль» заданы законы распределения времени обработки данных на серверах при помощи ранее созданного нами свойства «Processing time». В поле «Ответ» будут выведены результаты эффективной работы серверов, высчитанные по формуле: «processing time*число вышедших транзактов после обработки сервером». Результат эксперимента показан на рисунках 44-47.

Controls	
WEB	ProcessingTime
5	Random.Triangular (1, 3, 8)
8	Random.Exponential (10)
10	Random.Uniform(2 , 10)

Рисунок 44. Поле управления данными.

Responses	
Work of FS1 (Seconds)	Work of FS2 (Seconds)
143,137	102,954
804,498	102,295
219,728	158,789

Рисунок 45. Работа файл-серверов.

Work of WEB server1 (Seconds)	Work of WEB server2 (Seconds)	Work of WEB server3 (Seconds)
79,0731	50,9388	25,3506
374,212	70,6492	80,6905
117,491	77,4031	39,7778

Рисунок 46. Работа WEB-серверов.

Work of SQL server 1 (Seconds)	Work of SQL server2 (Seconds)	Work of SQL server3 (Seconds)
71,9818	47,8513	58,0384
61,562	149,557	88,9454
101,995	80,3673	89,5321

Рисунок 47. Работа SQL-серверов.

б) Анализ работы терминалов. На рисунке 48 представлена часть БД, посвященная анализу работы терминалов. Проанализировав данные, можно увидеть, что сгенерировано в целом 109 запросов. Самое большое количество запросов сгенерировали терминал 1 и 2, то есть поступило больше всего WEB-запросов от пользователей, как и запросов на файл-сервер от пользователей терминала 2.

тип объекта	имя объекта	источник данных	категория	элемент данных	статистика	значения
источник	terminal1	выходной буфер	содержимое	число транзактов на	максимум	1
				источнике	минимум	0
			пропускная способность	число входных транзактов	общее	43
				число выходных транзактов	общее	43
	terminal2	выходной буфер	содержимое	число транзактов на	максимум	1
				источнике	минимум	0
			пропускная способность	число входных транзактов	общее	43
				число выходных транзактов	общее	43
	terminal3	выходной буфер	содержимое	число транзактов на	максимум	1
				источнике	минимум	0
			пропускная способность	число входных транзактов	общее	23
				число выходных транзактов	общее	23

Рисунок 48. Таблица с параметрами терминалов.

в) анализ работы коммутаторов. На рисунке 49 представлена часть БД, посвященная анализу работы коммутаторов. Коммутаторы стали более нагруженными, через третий коммутатор проходит 109 запросов. Если умножить это значение на время передачи транзакта через коммутатор, то получим приблизительно значение 1100 секунд, что для данной модели достаточно большое время эффективной работы.

конвейер	commutator1	элементы конвейера	содержимое	число транзактов на конвейере	максимум	4		
					минимум	0		
			время добега-ния стока	время передачи транзактов через конвейер (с)	максимум	7,34		
					среднее	7,01		
			пропускная способность	число входных транзактов	общее	43		
					число выходных транзактов	общее	43	
			commutator2	элементы конвейера	содержимое	число транзактов на конвейере	максимум	5
							минимум	0
	время добега-ния стока	время передачи транзактов через конвейер (с)			максимум	6,34		
					среднее	6,27		
	пропускная способность	число входных транзактов			общее	66		
					число выходных транзактов	общее	66	
	commutator3	элементы конвейера			содержимое	число транзактов на конвейере	максимум	11
							минимум	0
			время добега-ния стока	время передачи транзактов через конвейер (с)	максимум	10,8		
					среднее	10,6		
			пропускная способность	число входных транзактов	общее	109		
					число выходных транзактов	общее	109	
			commutator4	элементы конвейера	содержимое	число транзактов на конвейере	максимум	7
							минимум	0
время добега-ния стока	время передачи транзактов через конвейер (с)	максимум			9			
		среднее			8,9			
пропускная способность	число входных транзактов	общее			43			
		число выходных транзактов			общее	43		

Рисунок 49. Таблица с параметрами коммутаторов.

г) анализ работы серверов. На рисунке 50 представлена часть БД, посвященная анализу работы серверов. Проанализировав данные, можно увидеть, что обработано 109 запросов. Ни один из них не был потерян, так как терминалы сгенерировали 109 запросов. Нагрузка на серверы распределена равномерно (50 процентов обработка, 50 процентов простой). В целом, стабилизируя некоторым образом обработку, можно в следующей модели добавить возможность выхода из строя серверов и их ремонт. Соответственно, в таблице следующей модели будут отражены «поломки» серверов и время, затраченное на ремонт.

сервер	FileServer1	ресурсы	структура ресурсов	время обработки транзактов (с)	процент	50				
					минимум	153				
					среднее	166				
				время простоя (с)	максимум	186				
					процент	50				
					минимум	153				
				входной буфер	содержимое	число транзактов на сервере	среднее	166		
							максимум	186		
						пропускная способность	число входных транзактов	общее	43	
		число выходных транзактов	общее	43						
		FileServer2	ресурсы	структура ресурсов	время обработки транзактов (с)	процент	50			
						минимум	250			
						среднее	266			
						время простоя (с)	максимум	280		
							процент	50		
минимум	250									
входной буфер	содержимое					число транзактов на сервере	среднее	266		
							максимум	280		
						пропускная способность	число входных транзактов	общее	0	
число выходных транзактов	общее			4						
WEB-server1	ресурсы			структура ресурсов	время обработки транзактов (с)	общее	23			
						число входных транзактов	общее	23		
						время простоя (с)	содержимое	число транзактов на сервере	минимум	0
									максимум	3
								пропускная способность	число входных транзактов	общее
		число выходных транзактов	общее			18				
		входной буфер	содержимое			число транзактов на сервере	процент	40		
							минимум	152		
							среднее	187		
				время простоя (с)	содержимое	число транзактов на сервере	максимум	238		
							процент	60		
						пропускная способность	число входных транзактов	общее	228	
		число выходных транзактов	общее	280,7						
		WEB-server2	ресурсы	структура ресурсов	время обработки транзактов (с)	максимум	357			
						время простоя (с)	содержимое	число транзактов на сервере	минимум	0
максимум	3									
пропускная способность	число входных транзактов							общее	14	
	число выходных транзактов					общее	14			
входной буфер	содержимое					число транзактов на сервере	процент	50		
							минимум	75		
							среднее	118		
						время простоя (с)	содержимое	число транзактов на сервере	максимум	171
				процент	50					
				пропускная способность	число входных транзактов			общее	75	
число выходных транзактов	общее				118					

Рисунок 50а. Таблица с параметрами серверов.

	WEB-server3	ресурсы	структура ресурсов	время обработки транзактов (с)	процент	50
					минимум	51
					среднее	88
				максимум	174	
				время простоя (с)	процент	50
					минимум	51
		среднее	88			
		входной буфер	содержимое	число транзактов на сервере	минимум	0
					максимум	3
			пропускная способность	число входных транзактов	общее	11
					число выходных транзактов	общее
			SQLserver1	ресурсы		структура ресурсов
минимум	61					
среднее	109					
максимум	148					
время простоя (с)	процент	60				
	минимум	91.5				
	среднее	163.5				
входной буфер	содержимое	число транзактов на сервере		минимум	0	
				максимум	3	
	пропускная способность	число входных транзактов		общее	18	
				число выходных транзактов	общее	18
	SQLserver2	ресурсы			структура ресурсов	время обработки транзактов (с)
			минимум	52		
среднее			112			
максимум			217			
время простоя (с)			процент	50		
			минимум	52		
		среднее	112			
входной буфер		содержимое	число транзактов на сервере	минимум	0	
				максимум	3	
		пропускная способность	число входных транзактов	общее	14	
				число выходных транзактов	общее	14
		SQLserver3	ресурсы		структура ресурсов	время обработки транзактов (с)
	минимум			63		
среднее	92					
максимум	134					
время простоя (с)	процент			50		
	минимум			63		
	среднее		92			
входной буфер	содержимое		число транзактов на сервере	минимум	0	
				максимум	2	
	пропускная способность		число входных транзактов	общее	11	
				число выходных транзактов	общее	11

Рисунок 506. Таблица с параметрами серверов.

д) анализ транзактов, генерируемых в модели. На рисунке 51 представлена часть БД, посвященная анализу существующих сущностей в модели.

сущность (транзакт)	FS1	множество	время добегания	время нахождения	минимум	98
		транзактов	стока	транзакта в	среднее	101
				системе (с)	максимум	108
	FS2	множество	время добегания	время нахождения	минимум	113
		транзактов	стока	транзакта в	среднее	138
				системе (с)	максимум	191
	zaproSWEB1	множество	время добегания	время нахождения	минимум	213
		транзактов	стока	транзакта в	среднее	232
				системе (с)	максимум	258
	zaproSWEB2	множество	время добегания	время нахождения	минимум	194
		транзактов	стока	транзакта в	среднее	226
				системе (с)	максимум	279
	zaproSWEB3	множество	время добегания	время нахождения	минимум	199
		транзактов	стока	транзакта в	среднее	240
				системе (с)	максимум	301

Рисунок 51. Таблица с параметрами транзактов.

Самым значимым параметром был выбран параметр «Время нахождения транзакта в системе». Долго ждать ответа на запрос придётся пользователю терминала 1. Транзакт может максимально находиться в системе 301 единицу модельного времени. Это обуславливается задержкой обработки на сервере, и возможной задержкой при передаче через коммутатор.

Выводы по модели 3

- 1) В модели реализована сложная архитектура сети ЦОД.
- 2) Проведен эксперимент структурной оптимизации.
- 3) Оптимизирована загрузка серверов.
- 4) Нет реализации повторной передачи данных при возникновении ошибки.