

УДК 004.414.23

Исследование работы центра обработки данных в объектно-ориентированной среде моделирования

*Труфанов Е.А., студент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»*

*Научный руководитель: Черненко М. В., к.т.н, доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»
chernen@bmstu.ru*

Известно, что имитационное моделирование является эффективным методом исследования сложных систем, поведение которой проявляется через взаимодействие ее элементов во времени и пространстве. Некоторая непредсказуемость этого взаимодействия создает неопределенность поведения системы. Соответственно, невозможность вовремя среагировать на определенное изменение в данном поведении может привести к неприятным последствиям, например, увеличения расходов на модернизацию данной системы, чтобы устранить эти недостатки. Поэтому имитационное моделирование оказывается универсальным подходом, который позволяет принять решение в условиях неопределенности, учесть в моделях факторы, которые трудно формализовать, выявить "узкие" места в системе, а также наглядно отобразить сложное взаимодействие между ее элементами. Кроме того, исследование, как правило, проходит без вмешательства в процессы уже работающей системы.

Технология SIMIO - это программное обеспечение, предназначенное для моделирования систем при условиях неопределенности ситуации. Это система моделирования, построенная на, так называемых, **интеллектуальных объектах**. Такие объекты создаются пользователем и могут быть заново использованы в комплексных проектах-моделях. Объекты могут находиться в библиотеках, и они легко доступны для использования. Объектом может быть любой предмет, например, механизм, робот, самолет, человек, или любой другой, встречающийся в жизни. Таким образом, есть возможность смоделировать любой элемент, взаимодействующий в системе. Модель строится посредством комбинации объектов, которые отражают физические компоненты системы. Модель в SIMIO выглядит, как реальная система. Логика и анимация модели

строятся как единый шаг, то есть, объект анимируется, тем самым, отражая изменение состояния объекта.

Каждый объект обязательно должен иметь процессы.

Процесс – последовательность действий (например, захват ресурса, задержка по времени), выполнение которой затрачивает определенное время и может изменить состояние модели. Процессы нужны или для того, чтобы определить поведение существующего объекта в системе или задать новые значения объекта.

Каждый процесс может состоять из одного или нескольких элементов, которые позволяют управлять транзактом, который поступил на вход данного процесса. Нужно отметить, что **транзакт** - минимальная единица в моделях SIMIO, обязательно имеет время создания и время уничтожения. Кроме того, транзакт - это тоже объект, с помощью него можно смоделировать элемент, взаимодействующий в системе.

Элементы, которые были использованы для моделирования:

Set Node – элемент, позволяющий установить узел назначения для транзакта.

Assign (рис. 1) – элемент, позволяющий задать новое значение для параметра.

Tally (рис. 1) - элемент, позволяющий собрать статистику для дальнейшего анализа.

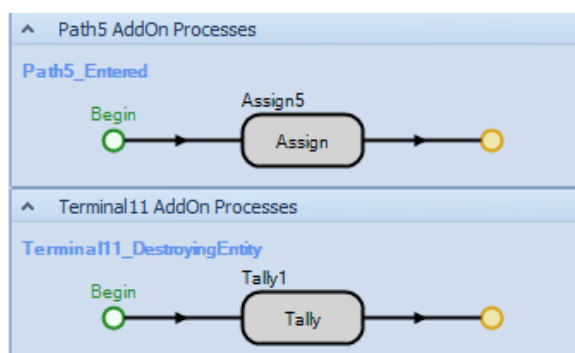


Рис. 1. Процессы задания новых значений транзакта и сбора статистики

Целью написания статьи является исследование возможностей технологии SIMIO при моделировании системы, связанной с предметной областью. Для достижения этой цели нужно решить следующие задачи:

- а) выбрать предметную область, которую необходимо моделировать. В данной работе – это работа центра обработки данных (ЦОД);
- б) выбрать состав ЦОД – физические устройства;
- в) определить логику взаимодействия этих устройств между собой;
- г) построить модель, используя интеллектуальные объекты SIMIO;

д) провести необходимые расчетные эксперименты, вывести результаты, в которых будут отражены параметры элементов, входящих в модель.

Предметная область для моделирования – работа центра обработки данных. В основе моделирования системы, использующей клиент-серверную технологию, положен диалоговый принцип: пользователь посылает запрос на сервер, где происходит его обработка, сервер формирует ответ, который пройдет тем же путем, как пришел запрос.

В модели используются интеллектуальные объекты SIMIO, чтобы включить в состав физические устройства центра обработки данных.

а) Объект терминал (рис. 2) в модели SIMIO представляет собой источник пользовательских запросов. Именно в терминале генерируются транзакты. В модели используются 3 терминала в качестве источника нагрузки для обработки на серверах.

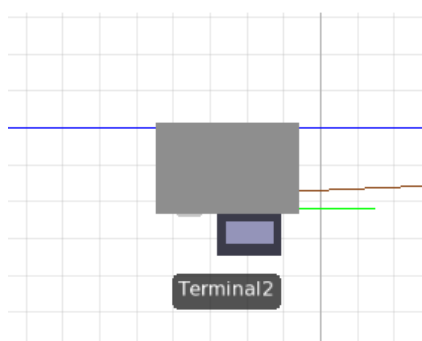


Рис. 2. Визуальное представление терминала в SIMIO

б) Коммутатор (рис. 3) – устройство для соединения нескольких узлов сети. В модели потребуется один коммутатор для соединения терминалов. Второй коммутатор – для соединения двух файл-серверов. Третий соединяет в сегмент три WEB-сервера, и последний коммутатор соединяет в сегмент три SQL-сервера. В модели сети используются четыре сегмента. Коммутатор 3 используется для передачи транзактов либо от всех терминалов, либо назад 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.

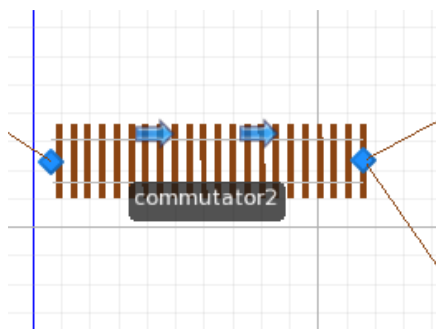


Рис. 3. Визуальное представление коммутатора в SIMIO

в) Роутер (рис. 4) – специализированное устройство, позволяющее передавать пакеты между различными сегментами сети. В модели задачей роутера является маршрутизация запросов между серверами.

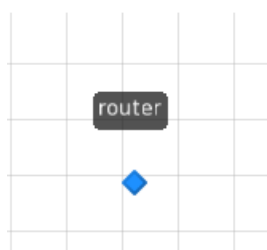


Рис. 4. Визуальное представление роутера в SIMIO

г) Сервер (рис. 5) – специальное компьютерное оборудование, специализированный для определенных сервисных функций. В модели существуют три типа серверов: три WEB-сервера, три SQL-сервера и два 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.

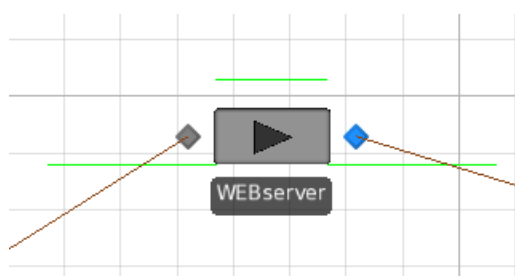


Рис. 5. Визуальное представление сервера в SIMIO

д) Линии связи (рис. 6) – совокупность технических устройств и физической среды, обеспечивающая распространение сигналов от передатчика к приемнику. В модели

служат для соединения остальных физических устройств, обеспечивая тем самым связность объектов в системе. Линии связи представлены путями, соединяющие два узла.

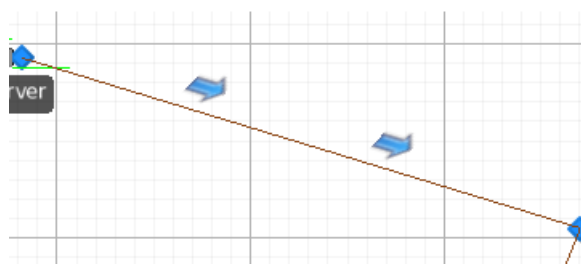


Рис. 6. Визуальное представление линии связи в SIMIO

После определения состава физических устройств в модели ЦОД необходимо задать логику взаимодействия данных устройств.

Создается три терминала, в каждом из них задается параметр, отвечающий за время генерации транзактов. 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. Тогда появляется возможность задать точное время генерации транзактов. Для этого создаются таблицы Time WEB (рис. 7), Time Sql1, Time Sql2. В свойствах этих таблиц задается тип единицы измерения и сами единицы измерения: время в секундах, соответственно. Задаются время появления транзактов и имена свойства. В итоге получены таблицы, которые можно заполнить вручную временами появления запросов. Таблиц три, так как имеется три терминала, которые генерируют транзакты.

Теперь необходимо определить сами транзакты для каждого терминала. Терминал 1 будет генерировать только WEB-запросы. На этой стадии необходимо задать переменную, которая будет отвечать за время обработки конкретного транзакта. Создается три вида по сложности запросов: ZaproWEB1, ZaproWEB2, ZaproWEB3.

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

Рис. 7. Заполненная таблица Time WEB

Затем в свойствах терминала 1 необходимо указать название переменной, или ссылки на таблицу, где указано, какому типу сущности какой закон распределения по времени обработки соответствует. Эта переменная задается в поле «Тип транзакта» свойств терминала 1. Терминалы 2 и 3 будут генерировать только SQL-запросы. В их свойствах указывается «тип транзакта», который генерируются этими терминалами. Во всех терминалах обязательно задается «arrival mode» (вид прибытия транзактов) в виде журнала запросов. На рис. 8 отражены заполненные поля свойств 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

Рис. 8. Значения свойств для транзактов терминала 1

Далее необходимо реализовать маршрутизацию запросов, чтобы WEB-запросы шли к WEB-серверу, а FS-запросы к файл-серверам. Для этого создается аналог таблиц маршрутизации, где ее компонентами являются непосредственно имена узлов сети. Так как типов запросов три, то и таблиц маршрутизации будет три. Для создания таблиц используется команда “Add sequence table”, и заполняется таблица именами тех узлов, через которые должен пройти запрос указанного типа.

Каждому транзакту (рис. 9) в свойство “Routing logic” в поле “Initial Sequence” добавляется значение – имя таблицы маршрутизации, которая будет использоваться для определения его пути в виде ответа от сервера до терминала (рис. 10).

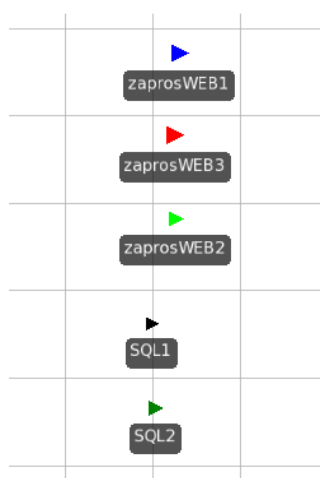


Рис. 9. Сущности (транзакты) в модели

Для завершения построения пути прохождения транзактов объединяются терминалы коммутатором. В качестве коммутатора будет выступать объект «Конвейер» (рис. 3). Он определяется, как объект позволяющий соединить два узла, но отличается от стандартной линии связи возможностью задать свойство буфера с накоплением. При отсутствии этого свойства появляется возможность останавливать работу конвейера, если транзакты прошли до конца конвейера, но им некуда выйти. При наличии этого свойства появляется возможность не останавливать работу конвейера и двигаться транзактам дальше. Если по каким-то причинам транзакт не может продвигаться дальше, он встает в очередь и будет пытаться пройти еще раз.

Properties: zaprosWEB (ModelEntity)	
Travel Logic	
Initial Desired ...	2.0
Initial Network	Global
Network Turnaro...	Exit & Re-enter
Routing Logic	
Initial Priority	1.0
Initial Sequence	Routing1

Рис. 10. Задание маршрута для транзакта «zaprosWEB»

Роутер реализован **узлом передачи**, и с него начинают работать таблицы маршрутизации.

Далее настраивается работа серверов. Основная суть заключается в том, чтобы в свойстве сервера (указывается для WEB-серверов и SQL-серверов) 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. Для реализации выходов из строя серверов воспользуемся свойством «Логика надежности» (рис. 11). В поле «Тип выхода из строя» укажем значение «зависимое от календарного времени», в поле «Интервал времени между сбоями в работе» укажем закон распределения данного интервала. Единицы измерения – секунды. Ниже есть поле «Время, затраченное на ремонт», в котором также укажем закон распределения времени, необходимое для восстановления работы сервера.

Reliability Logic	
Failure Type	Calendar Time Based
Uptime Between Failures	Random.Exponential(150)
Units	Seconds
Time To Repair	Random.Triangular(0.5, 1.0, 1.5)
Units	Seconds

Рис. 11. Логика надежности сервера

Все указанные выше объекты соединяются линиями связи в единую топологию, учитывая составленные ранее таблицы маршрутизации. Все объекты удваиваются, чтобы обеспечить обратный путь. Это же правило касается и линий связи. Но добавляются также линии связи на узлы конвейеров (коммутаторов), чтобы обеспечить обратный путь транзактам с некорректными данными. Затем будет произведена попытка повторной передачи данных. Для того чтобы различать корректные данные, некорректные данные обозначены, как транзакты красного цвета. Но эта условность существует только для человека, при моделировании, чтобы транзакт расценивался, как корректный или некорректный, необходимо создать процесс привязки определенного состояния транзакта (без ошибки/с ошибкой) к определенному цвету. К красному – если транзакт ошибочный, к другому цвету – правильный. Процесс называется ASSIGN, он позволяет переопределить стандартное значение для объекта. Задаем значение 0 для красных транзактов, 1 – для остальных (рис. 12).

Basic Logic	
State Variable Name	MyModelEntity.Picture
New Value	0.0
Assignments (More)	0 Rows

Рис. 12. Определение значения для состояния транзакта

Далее необходимо у выходных узлов конвейеров задать правило выхода транзакта – по весу линии связи. Задаются веса выходных линий связи - основной, что включена в основной маршрут, дополнительной, по которой пройдут транзакты с ошибкой. Графическое представление работающей модели показано на рис. 13, 14, 15.

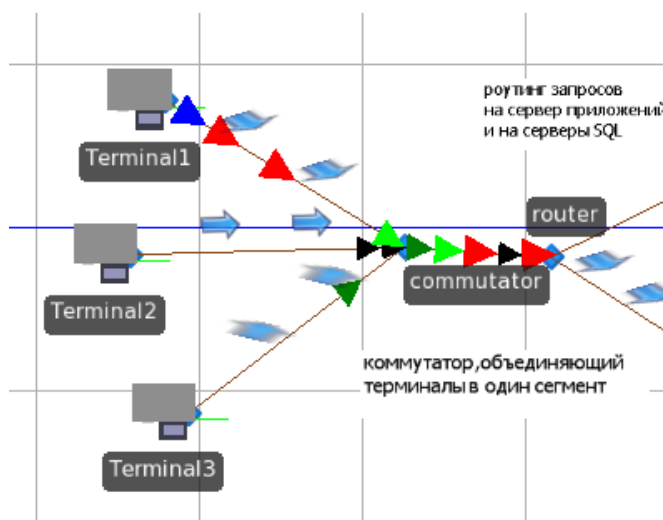


Рис. 13. Генерация транзактов, передача через коммутатор на роутер

После генерации транзактов и передачи их через коммутатор роутер распределяет эти транзакты на сервера для обработки.

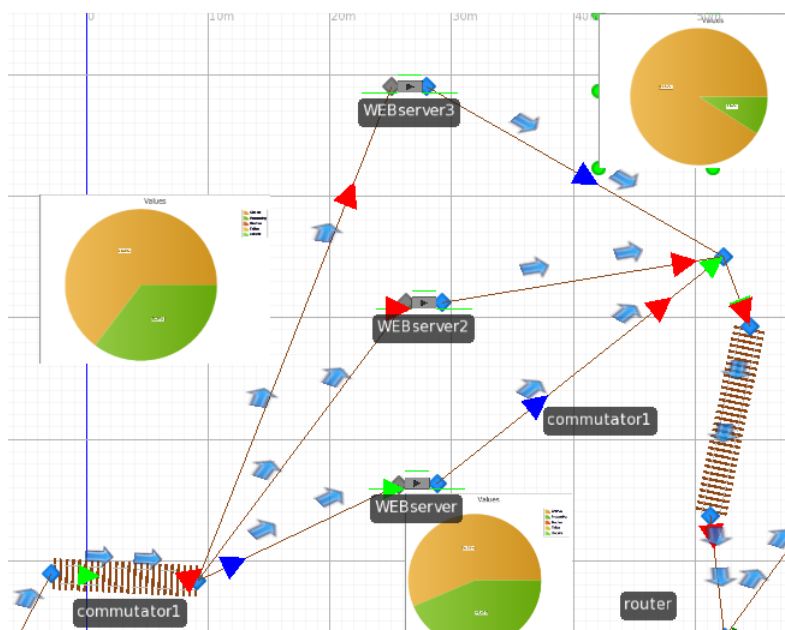


Рис. 14. Работа WEB-серверов

После успешного процесса обработки сервер посылает ответ в виде того же самого транзакта терминалу, который сгенерировал данный транзакт.

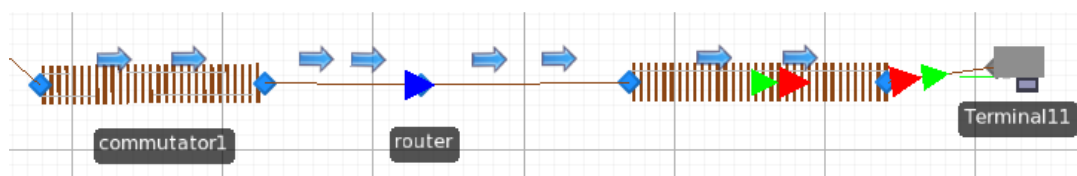


Рис. 15. Передача ответов с WEB-серверов к терминалу 1

Для получения результатов моделирования проводятся расчетные **эксперименты** над моделью. Для этого необходимо заполнить два поля - Controls (Контроль), Responses (Ответы). В поле «Контроль» заданы законы распределения времени обработки данных на серверах. В поле «Ответы» будут выведены результаты эффективной работы серверов, высчитанные по формуле: «ProcessingTime*число вышедших транзактов после обработки сервером». В "ProcessingTime" указывается закон распределения, согласно которому будет обрабатываться каждый транзакт, появившийся на сервере. Результат эксперимента показан на рис. 16.

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

а

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

б

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

в

Work of SQL server1 (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

г

Рис. 16. а - поле управления данными с заполненным свойством "ProcessingTime"

б - время работы файл-серверов

в - время работы WEB-серверов

г - время работы SQL-серверов

а) Результат работы терминалов. На рис. 17 представлена часть таблицы выходных параметров расчетного эксперимента. Проанализировав результаты, 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
	terminal2	выходной буфер	содержимое	число транзактов на	максимум	1
				источнике	минимум	0
			пропускная способность	число входных 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

Рис. 17. Таблица с параметрами терминалов

б) результат работы коммутаторов. На рис. 18 представлена часть таблицы, выходных параметров работы коммутаторов.

commutator3	элементы конвейера	содержимое	число транзактов на конвейере	максимум	11			
				минимум	0			
		время добегания стока		содержимое	время передачи транзактов через конвейер (с)	максимум	10.8	
						среднее	10.6	
						минимум	10.4	
		пропускная способность		содержимое	число входных транзактов	общее	109	
						число выходных транзактов	общее	109
						число ошибочных транзактов	общее	20
		commutator4	элементы конвейера	содержимое	число транзактов на конвейере	максимум	7	
						минимум	0	
время добегания стока				содержимое	время передачи транзактов через конвейер (с)	максимум	9	
						среднее	8.9	
						минимум	8.8	
пропускная способность				содержимое	число входных транзактов	общее	43	
						число выходных транзактов	общее	43
						число ошибочных транзактов	общее	28

Рис. 18. Таблица с параметрами коммутаторов

Анализ показал, что больше всего ошибок приходится на коммутатор 4.

в) результат работы серверов. На рис. 19 представлена часть таблицы выходных параметров работы серверов. Показан сервер, на котором задержка обработки транзактов по времени превышает задержку по времени остальных серверов.

WEB-server1	ресурсы	структура ресурсов	время обработки транзактов (с)	процент	40			
					минимум	152		
					среднее	187		
			время простоя (с)				максимум	238
							процент	60
							минимум	228
			время ремонта (с)				среднее	280.7
							максимум	357
							общее	374
			входной буфер		содержимое	число транзактов на сервере	минимум	0
	максимум	3						
пропускная способность		содержимое			число входных транзактов	общее	18	
						число выходных транзактов	общее	18

Рис. 19. Таблица с параметрами сервера

г) Информация о транзактах, генерируемых в модели. На рис. 20 представлена часть таблицы существующих транзактов 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
	zaposWEB1	множество	время добегания	время нахождения	минимум	252
		транзактов	стока	транзакта в	среднее	362
				системе (с)	максимум	571
	zaposWEB2	множество	время добегания	время нахождения	минимум	218
		транзактов	стока	транзакта в	среднее	331
				системе (с)	максимум	601
	zaposWEB3	множество	время добегания	время нахождения	минимум	233
		транзактов	стока	транзакта в	среднее	374
				системе (с)	максимум	663

Рис. 20. Таблица с параметрами транзактов

Вполне закономерно, что увеличилось время существования транзактов (WEB-запросов) в системе. Оно обусловлено задержками при ремонте серверов и повторных передачах на коммутаторах.

Анализ выходов из строя серверов представлен на рис. 21. Нужно отметить, что единицы измерения - минуты.



Рис. 21. Таблица с параметрами транзактов

Проявляется закономерность - чем больше выходов из строя, тем больше транзактов будет находиться в очереди на обработку.

Таким образом, система SIMIO обладает широкими возможностями по моделированию систем и анализа их.

В заключение статьи представлены общие выводы по моделированию в системе SIMIO.

Можно было бы сравнить технологию SIMIO с другой технологией моделирования, например с GPSS. Но использовать метод взвешенной суммы локальных критериев не получится, так как технологии слишком разные. В GPSS используется набор команд для того, чтобы описать процесс, в SIMIO мы «рисует» процесс. Можно сравнить

с технической точки зрения эти две технологии, но здесь тоже не все однозначно. В SIMIO используется каждое ядро процессора для просчета каждого эксперимента, причем 10 репликаций с двумя ядрами прогоняются приблизительно за 7 секунд. В GPSS подсчеты результатов протекают довольно медленно. В целом, напрямую эти технологии сравнить нельзя, они не являются однотипными объектами, как, например, серверы или другие физические устройства.

Основное преимущество SIMIO именно в возможности визуального моделирования. Это намного упрощает решение задач моделирования, так как позволяет видеть результаты моделирования на любой стадии (стадия генерации транзактов, обработка или же стадия уничтожения транзактов).

Есть возможность на любой стадии моделирования проводить расчетные, статистические эксперименты.

Транзакты в SIMIO – не что-то абстрактное, а конкретный объект, имеющий размерность (высота, ширина, длина, объем и прочие).

Количество используемых объектов в модели зависит от ресурсов ЭВМ.

В SIMIO существуют разные формы представления результатов (сводная таблица со всеми параметрами модели, графики, диаграммы, отчеты, результаты экспериментов).

Графики могут строиться в реальном времени моделирования.

Из недостатков можно выделить не до конца продуманную логику маршрутизации. Например, если данный узел есть в созданной таблице маршрутизации, то вы не сможете реализовать на нем правило выборки выходной линии (по весам линий связи).

Список литературы

1. Процесс, интеллектуальные объекты. «Simio Reference guide». Режим доступа: http://lyle.smu.edu/emis/docs/Simio/Simio_Reference_Guide.pdf (дата обращения 20.02.2015).
2. Принципы построения моделей. Видеоуроки от разработчиков системы. Режим доступа: <http://www.simio.com/resources/videos/introduction-to-simio-training-series/> (дата обращения 03.03.2015).
3. Термины имитационного моделирования. Черненький В. М. Формирование имитационного процесса на основе алгоритмического описания функционирования информационной системы // Наука и образование. МГТУ им. Н. Э. Баумана. Электрон. журн. 2011, № 11. Режим доступа: <http://technomag.bmstu.ru/doc/291975.html> (дата обращения 25.02.2015).

4. Принципы построения моделей сети. Галкин В. А., Миляев Н. А. Подходы к построению имитационных моделей сети хранения данных // Инженерный вестник. МГТУ им. Н. Э. Баумана. Электрон. журн. 2013. № 8. Режим доступа: <http://engbul.bmstu.ru/doc/638235.html> (дата обращения 10.03.2015).
5. Логика построения ЦОД, расчетные эксперименты. W. David Kelton, Jeffrey S. Smith, David T. Sturrock. Simio And Simulation: Modeling, Analysis, Applications. Piitsburg, PA,,: Createspace, 2013. 416 p.
6. Определения сервера, роутера, коммутатора, линий связи. Википедия. Режим доступа: <https://ru.wikipedia.org/wiki/> (дата обращения 10.03.2015).