

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ СБОРА И ОБРАБОТКИ ДАННЫХ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Е.В. Бочкарева, Л.И. Сучкова, А.И. Харламов, А.Г. Якунин

В данной работе рассмотрены принципы организации среды имитационного моделирования для исследования и оптимизации работы распределенной системы сбора и обработки данных. Предлагается организовать имитационную систему как набор взаимодействующих систем массового обслуживания (СМО) различных типов, в качестве требований в которых выступают события разных типов, а в качестве обслуживаемых устройств – диспетчеры устройств и диспетчер системы. Для моделирования систем получения данных по запросу (систем управления базами данных, хранилищ данных и т.д.) в параметрах модели учитываются размеры кешей диспетчеров, алгоритмы вытеснения данных, и размеры блоков данных в кеше.

Ключевые слова: распределенная система, имитационное моделирование, СМО, кеширование данных

Важным направлением в области разработки распределенных вычислительных систем является предварительное моделирование их работы, которое позволяет снизить риски и удешевить проектирование аппаратной части, структуры и размещения данных [1]. Результатом работы имитационной модели являются собранные в ходе вычислительных экспериментов статистические данные о наиболее важных характеристиках работы сети при заданных параметрах, что позволяет проектировать эффективную архитектуру распределенных систем, выбирать оборудование, протоколы взаимодействия, хранилища данных, программное обеспечение и т.д. Кроме того, предварительное моделирование работы распределенных систем позволяет выявить «узкие» места системы и предотвратить нештатные ситуации.

В аналогичных разработках авторами [2-3] учитывается загруженность системы: трафик по каналам связи в зависимости от параметров сетевого оборудования и каналов связи. Однако, не предусмотрена возможность анализа характеристик системы в зависимости от временной и емкостной сложности алгоритмов обработки данных, выполняющихся на устройствах сети, от протоколов передачи данных и структуры выполняющихся запросов. Кроме того, существующие программные продукты для проектирования распределенных систем нацелены на решение конкретных узкоспециализированных задач и определенное оборудование.

В аналогичных разработках применяется объектно-ориентированный подход к предметной области, заключающийся в том, что

вычислительную систему рассматривают как совокупность взаимодействующих объектов (микроконтроллеров, компьютеров, каналов связи, сетевого оборудования, средств хранения информации и т. д.), обладающих разными характеристиками [4]. Нами предлагается использовать функционально - ориентированный подход, суть которого - представить систему в виде набора функций, преобразующих поступающие данные. В рамках имитационной системы это реализуется путем выделения процессов сбора и обработки данных, выполняющихся на каждом устройстве, определения временной и емкостной сложности алгоритмов их реализующих, и выявления задержек в системе, вызываемых выполнением этих алгоритмов.

Для моделирования процессов в гетерогенной распределенной вычислительной среде разработана событийно-ориентированная имитационная система, в которой предусмотрены возможности построения сети заданной конфигурации из имеющихся базовых компонентов. Устройства на разных уровнях архитектуры различаются по функциональным возможностям и характеристикам, например, по быстродействию, объему памяти для работы алгоритмов обработки данных и для хранения результатов измерений и вычислений.

Поскольку события, происходящие в системе, носят случайный характер, то предлагается организовать работу каждого узла по принципу одноканальной системы массового обслуживания (СМО) с бесконечной или ограниченной очередью [5]. В качестве требования будем рассматривать событие, а в ка-

РАЗДЕЛ I. ОБЩИЕ ВОПРОСЫ СИНТЕЗА И АНАЛИЗА ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ СИСТЕМ

честве обслуживающего прибора – диспетчер (главная функция, обработчик событий). Тогда вся вычислительная система описывается набором взаимодействующих СМО различных типов. Это позволяет перейти от динамического моделирования системы, когда исследуем маршруты движения пакетов в сети, достижимость узлов, стремимся к минимизации трафика, к функциональному моделированию, при котором исследуется правильность выполнения запросов, оптимальность времени их выполнения.

Список событий определяется для каждого компонента имитационной системы: для сетевого оборудования, вычислительных узлов, дисковых хранилищ и т.д.

Предусмотрены очереди заявок для диспетчера каждого устройства и для диспетчера имитационной системы в целом. События, относящиеся к логике работы конкретного устройства (например, генерация некоторого значения, архивация данных, их фильтрация), обрабатываются диспетчером этого устройства. События, относящиеся к логике работы всей системы или необходимые для имитации взаимодействия между устройствами (например, остановка или запуск некоторого устройства, разрыв канала связи, нештатная ситуация, обмен данными между устройствами на аппаратном уровне, изменение масштаба модельного времени), обрабатываются главным диспетчером имитационной системы.

При обработке заявки в очереди этого или другого устройства (устройств) могут формироваться новые события.

Каждое событие характеризуется:

- типом (например, возникновение нештатной ситуации, отправка запроса на данные, получение данных, остановка работы процесса и т.д.);
- статусом (обработано или нет); планируемым временем обработки (некоторый момент модельного времени по расписанию);
- реальным временем обработки (момент модельного времени, когда диспетчер реально начал обработку события).
- Разница между планируемым и реальным временем наступления события представляет значительный интерес, так как характеризует задержку, возникающая в системе, которую надо стремиться минимизировать.

Поток заявок является неравномерным, так как события разных типов поступают на

обработку через неодинаковые промежутки времени и время обслуживания заявок тоже различается. Сохранение неоднородности реализуется путем выделения фиксированного количества видов обслуживаемых событий и определения алгоритма обработки для них. Законы распределения интервалов времени поступления заявок каждого типа в систему и распределения времени их обслуживания диспетчером выбираются исследователем в начале моделирования.

На время обработки события влияет тип события: генерация данных, фильтрация, архивация, обработка нештатной ситуации и т.д. Предусмотрена реализация событий в виде функций, обладающих разной временной и емкостной сложностью. Порядок сложности определяет класс функции времени обработки данных по определенному алгоритму. Функция может быть линейной, полиномиальной, экспоненциальной, логарифмической и т.п. Параметр, передаваемый функции, и порядок ее сложности определяют количество операций, необходимых для ее вычисления, и, как следствие, время (в условных временных единицах системы), необходимое для обработки соответствующего события. Емкостная сложность вычисляется аналогично и определяет объем оперативной памяти устройства, необходимой для вычисления функции его процессором.

На каждом устройстве в параллельных потоках одновременно могут функционировать несколько процессов. Процесс – это численное отражение изменения состояния некоторого объекта в системе. Процесс может запрашивать и/или получать данные от других процессов; отправлять данные другим процессам; обрабатывать и/или анализировать полученные данные и в зависимости от результатов обработки генерировать сообщения или управляющие сигналы для исполнительных механизмов. Это реализуется постановкой в очередь некоторого устройства (устройств) событий по обработке или получению данных. Пример того, какие события вызывает обмен данными между двумя процессами, приведен на рисунке 1.

Процессы, как и физические устройства, являются базовыми компонентами системы и могут настраиваться путем выбора функций и алгоритмов обработки данных. События, поступающие от процессов, выполняющихся на этом устройстве, помещаются в его очередь и обрабатываются диспетчером этого устройства.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ СБОРА И ОБРАБОТКИ ДАННЫХ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

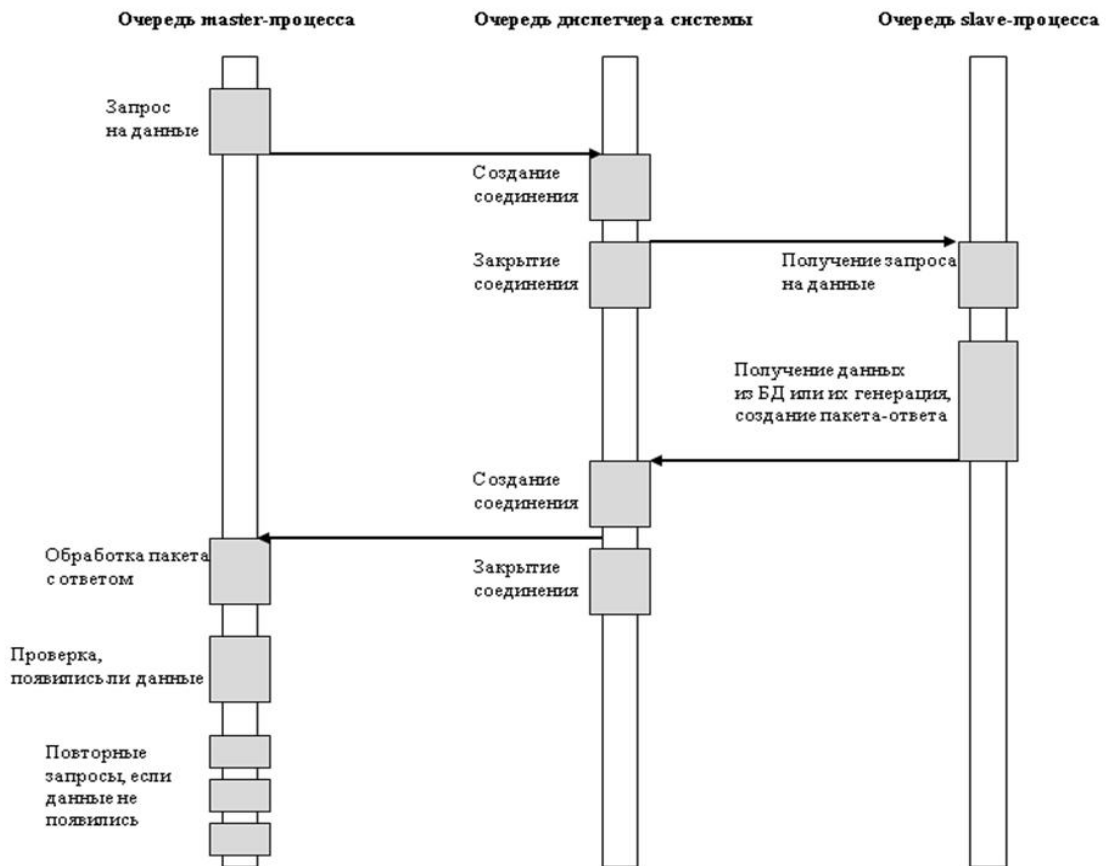


Рисунок 1 - Пример обработки запроса на получение данных одним процессом от другого.

Особый интерес представляют процессы, запрашивающие и обрабатывающие данные других процессов. Их работа организована по принципу ПОЛИЗа. Сначала в очередь устройства, на котором выполняются такие процессы, ставятся события на запросы данных от ведомых процессов и только тогда, когда собраны все необходимые для вычисления главного процесса данные (аргументы), происходит обработка события по обработке этих данных. Пусть процесс f_1 вычисляет сумму данных, полученных от процессов 2 и 3: $f_1 \rightarrow f_2 + f_3$. Тогда очередь событий, вызываемых процессом f_1 в очереди устройства, на котором этот процесс функционирует, будет выглядеть следующим образом:

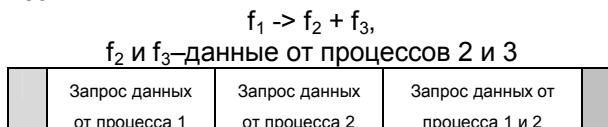


Рисунок 2. Организация очереди устройства при обслуживании процесса по принципу ПОЛИЗа.

Время в имитационной системе измеряется в условных временных единицах, не за-

висающих от реального времени или тактов работы процессоров устройств.

Коррекция таймера в имитационной системе осуществляется по событийному принципу: модельное время сдвигается не на фиксированную величину, а до первого запланированного события. В отличие от метода коррекции таймера с постоянным шагом, это гарантирует, что события будут обрабатываться одновременно тогда и только тогда, когда они действительно запланированы на одно время. За коррекцию таймера модельного времени отвечает главный диспетчер системы. Он определяет, в какой из очередей находится ближайшее запланированное событие, и передает управление диспетчеру соответствующего устройства. Если диспетчер устройства в этот момент не занят, то начинается обработка события. Иначе событие вновь ставится в очередь к соответствующему диспетчеру и к диспетчеру всей системы (если организована СМО с ожиданием) или покидает систему без обслуживания (в СМО с отказами).

Если работу систем сбора и обработки данных можно моделировать набором связанных очередей со временем обработки,

РАЗДЕЛ I. ОБЩИЕ ВОПРОСЫ СИНТЕЗА И АНАЛИЗА ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ СИСТЕМ

зависящим от архитектуры системы, характеристик ее устройств, параметров события, то системы получения данных по запросу (в частности системы управления базами данных, хранилища данных и т.д.) имеют более сложную структуру, обусловленную типом решаемых задач. Для таких систем характерно количество исходных данных, намного превышающее объем доступной оперативной памяти. Для дисковых хранилищ данных время доступа к информации на 3-4 порядка больше времени доступа в память [6]. Поэтому в память узла системы подгружается только ограниченный объем данных, над которыми и проводятся операции.

Производительность современных вычислительных систем ограничена не столько производительностью процессоров, сколько возможностью доставлять данные к ядрам процессоров и узлам кластера. Передача данных является узким местом, как при сетевом обмене, так и при обмене между ядрами процессоров и памяти, что проявляется в повсеместном использовании буферов (кешей) на всех уровнях архитектуры вычислительных систем (рисунок 3).

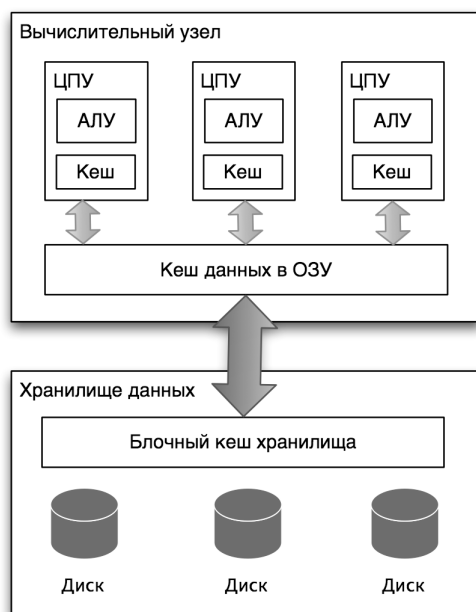


Рисунок 3 - Использование буферов (кешей) на разных уровнях архитектуры.

Как правило, вычислительная система представляет собой набор иерархий кешей с разным временем доступа [7]. Операции обработки данных в первую очередь обращаются в кеш, и если необходимые данные там отсутствуют ("промах кеширования" - cache miss), то происходит обращение к более низ-

ким и медленным уровням иерархии.

Ввиду ограниченности размера кешей, важно определить метод выбора набора закешированных данных, размера буферов и структуры данных для максимизации производительности системы.

Произвольный кеш может быть описан как словарь данных ограниченного размера, в котором блок данных связан с одним или более ключом. Диспетчер очереди в процессе обработки события обращается по ключу в кеш. Если необходимые данные в кеше отсутствуют, то они подгружаются с нижних уровней иерархии. В то же время из-за ограничений по размеру доступной памяти операция добавления данных в кеш выполняется совместно с удалением старых записей.

В большинстве случаев сложность систем и непредсказуемость реальных запросов не позволяют выбрать оптимальное множество закешированных ключей во время кеширования. Поэтому эффективность кеширующей иерархии зависит от алгоритма выбора вытесняемых данных, который определяет содержимое кеша и его способность быстро исполнять необходимые операции доступа. Таким образом, алгоритм вытеснения должен обеспечивать оптимальное или субоптимальное содержимое кешей, динамически подстраивая его под характеристики входящего потока запросов во время выполнения на основе исторических данных. В такой ситуации оптимальная конфигурация кеша для потока с одним профилем входящих событий может быть неоптимальна для потока другого профиля. Более того, переключение между разными профилями потока запросов породит переходный процесс, в течение которого производительность системы может быть значительно ниже средних значений. Это особенно важно, так как большинство промышленных систем одновременно обрабатывают несколько типов нагрузки, часто в режиме разделения времени, то есть переключаются между различными типами нагрузки.

Таким образом, к параметрам модели добавляются размеры кешей диспетчеров, алгоритмы вытеснения данных, и размеры блоков данных в кеше. Время обработки события есть также функция состояния кеша.

Выводы.

В результате исследования для моделирования работы гетерогенной распределенной системы сбора и обработки данных разработана событийно-ориентированная имитационная система. Предусмотрена возмож-

ПОЛЗУНОВСКИЙ ВЕСТНИК № 2 2010

БЫСТРЫЙ СУБОПТИМАЛЬНЫЙ СПОСОБ ВОССТАНОВЛЕНИЯ ТАКТОВОЙ И ЦИКЛОВОЙ
СИНХРОНИЗАЦИИ ПРИ НЕКОГЕРЕНТНОЙ ДЕМОДУЛЯЦИИ ЧАСТОТНО-МАНИПУЛИРОВАННОГО
СИГНАЛА С НЕПРЕРЫВНОЙ ФАЗОЙ

ность построения сети заданной конфигурации из имеющихся базовых компонентов (компьютеров, контроллеров, первичных преобразователей, каналов связи, сетевого оборудования, хранилищ данных и т.д.), обладающих заданными характеристиками. В построенной имитационной системе моделируется переключение между различными профилями потока запросов. Реализованы средства вывода и обработки статистической информации о результатах работы системы. На основе собранных статистических данных исследуются:

- общая производительность системы;
- потери на переходных процессах вариацией алгоритмов управления;
- размеры и структура кешей;
- задержки, возникающие в системе при ее работе в обычном (стационарном) режиме и при возникновении нештатных ситуаций;
- характеристики системы: длины очередей заявок; соотношение числа обслуженных требований и времени обслуживания требования; число событий, обслуженных без очереди и т.д. – в зависимости от числа узлов в сети, числа уровней в архитектуре и конфигурации сети; характеристик сети в зависимости от выбранного для реализации вида СМО диспетчеров системы и устройств.

СПИСОК ЛИТЕРАТУРЫ

1. Емельянов, В. Имитационное моделирование систем / В.В.Емельянов, С.И.Ясиновский. – М.: Издательство МГТУ им. Баумана, 2009 – 583 с.
2. Анашкин, А. Техническое и программное обеспечение распределенных систем управления./А.С. Анашкин, Э.Д. Кадыров, В.Г. Хазаров. под ред проф. В.Г. Хазарова – СПб.: П-2, 2004 – 368с.
3. Рыжиков, Ю. Имитационное моделирование: Теория и технологии / Ю.И. Рыжиков. - Киев: Альтекс, 2004. – 384с.
4. Бахарова, Н. Анализ производительности сетевых структур методами теории массового обслуживания. / Н.Ф.Бахарова // Научно-технические ведомости СПбГПУ. – 2009 – №3(80). – С. 25-34.
5. Гнеденко, Б. Введение в теорию массового обслуживания. / Б.В. Гнеденко, И.Н. Коваленко – М.: ЛКИ, 2007 - 400 с
6. Таненбаум, Э. Современные операционные системы. 2-е изд.. / Э. Таненбаум - СПб: Питер, 2002. - 1040 с.
7. Таненбаум, Э. Архитектура компьютера. / Э. Таненбаум - СПб: Питер, 2007. - 848 с.

Аспирант Е.В. Бочкарева – 22bev@mail.ru; к.т.н., проф. Л.И. Сучкова – lis@agtu.secna.ru; аспирант А.И. Харламов – aharlamov@gmail.ru; д.т.н., проф. А.Г. Якунин - (385-2) 29-07-86 - Алтайский Государственный технический университет, кафедра вычислительных систем и информационной безопасности.

УДК 004.318-181.4:681.51

БЫСТРЫЙ СУБОПТИМАЛЬНЫЙ СПОСОБ ВОССТАНОВЛЕНИЯ
ТАКТОВОЙ И ЦИКЛОВОЙ СИНХРОНИЗАЦИИ ПРИ
НЕКОГЕРЕНТНОЙ ДЕМОДУЛЯЦИИ ЧАСТОТНО-
МАНИПУЛИРОВАННОГО СИГНАЛА С НЕПРЕРЫВНОЙ ФАЗОЙ

В.В. Василевский

Проанализированы свойства способа восстановления тактовой синхронизации для частотно-манипулированных сигналов с непрерывной фазой путем поиска максимума раскрытия глазковой диаграммы по ограниченному набору отсчетов сигнала на выходе частотного дискриминатора [1]. Получены зависимости вероятности ошибок первого, второго рода и суммарной вероятности ошибки, графики функций распределения вероятности для различных отношений сигнал/шум (ОСШ). Произведено сравнение эффективности способа для синхропосылок разной длины и типа, произведено сравнение способа с оптимальным некогерентным обнаружителем, сделаны выводы и рекомендации по применению.

Ключевые слова: частотная манипуляция с непрерывной фазой, восстановление, синхронизации, тактовая синхронизация, цикловая синхронизация.

При построении многоканального цифрового радиоприемного устройства дискретных сообщений важной частью является реализация системы восстановления синхрони-

зации [2]. При некогерентном приеме необходимо восстановление следующих видов синхронизации: тактовой (символьной), цикловой (кадровой) и частотной [3].