

А.А. ГОРДЕЕВ, С.А. МАРЬИН, А.В. ТКАЧУК, А.В. ЦЫМБАЛ

ЭВРИСТИЧЕСКИЕ МЕТОДЫ ОПТИМИЗАЦИИ РЕЗЕРВИРОВАНИЯ В СИСТЕМАХ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ

Важной задачей распределения ограниченного числа ресурсов между поступающими заявками является выбор такого механизма распределения, при котором используются данные ресурсы с максимальной отдачей. Задача ставится таким образом: пусть имеется набор ресурсов (объектов), каждый из которых обладает набором свойств. В случайные моменты времени поступают заявки на резервирование (использование) набора ресурсов с заданными свойствами в определенный промежуток времени. Временной промежуток заявки на ресурс может начинаться либо с текущего момента времени, либо с определенного момента в будущем. В процессе жизненного цикла системы заявки на резервирование могут создаваться, переноситься во времени, удаляться, изменять требования к ресурсам. Возможны такие ситуации, когда в определенные моменты ресурсы заняты (зарезервированы) таким образом, что невозможно удовлетворить очередную заявку, хотя общая загруженность ресурсов невысока.

Итак, в процессе резервирования система должна выполнить следующие действия:

- определить наличие подмножества свободных в данный промежуток времени ресурсов с заданными свойствами;
- выбрать из этого подмножества нужное число ресурсов;
- попытаться перераспределить уже имеющиеся заявки таким образом, чтобы освободить необходимое число ресурсов, если на данный промежуток времени в системе нет свободных ресурсов.

Типичным примером подобной модели может служить резервирование номеров в гостинице. Обычно резервирование происходит следующим образом: клиент звонит по телефону в гостиницу и спрашивает: “Есть ли у Вас двухместный номер “люкс” с 21 октября на 7 дней, и сколько это будет стоить”? Работник гостиницы должен быстро определить наличие свободных мест и выдать информацию клиенту. Гостиница будет терпеть убытки в случае неоперативного реагирования на заявки клиентов. Таким образом, в этом примере ресурсами являются жилые номера, а свойствами ресурсов — количество комнат в номере, наличие телевизора, мини-бара, факса, уровень престижности.



Рис. 1

На рис. 1 слева изображен список номеров; а сверху — календарь. Таким образом бронь *A* занимает номер 101 с 21.10.97 по 23.10.97. Допустим, поступает заявка на бронирование номера люкс с 21.10.97 по 26.10.97. Задача сводится к поиску номера класса люкс, свободного в эти 7 дней. Как видно на рисунке, в данный момент нет свободного номера в требуемый календарный промежуток. Однако удовлетворить заявку возможно, если переместить бронь *B* из номера 102 в номер 101. В этом случае номер 102 освободится на необходимое время, и заявка может быть выполнена.

Рассмотрим процесс обслуживания заявок в системе. Рассматриваемая система является разновидностью системы массового обслуживания (СМО) с потерями [1,2]. Данный тип СМО характеризуется отсутствием очереди (сразу же после поступления заявку надо либо принять и обработать, либо отказать). Целью работы таких СМО обычно является минимизация потерь. Примером такого типа СМО являются АТС, гостиницы.

СМО с потерями бывают двух видов:

- с немедленной обработкой заявки (городские АТС);
- с заявками, которые необходимо выполнять в определенное время, т.е. заявка здесь — резервирование ресурса на определенное время (АТС с заказными разговорами, гостиницы).

В обоих случаях сразу после поступления заявки необходимо либо принять ее, либо отказать.

В системах второго типа в процессе поступления заявок происходит перераспределение ресурсов между заявками так, чтобы уже принятые заявки были обработаны на одном из подходящих ресурсов, и чтобы потери (необработанные заявки) были минимальными. Рассматриваемая нами система является примером СМО с потерями второго типа.

Мы предлагаем использовать двухуровневую схему обработки заявок:

- обработка заявок при наличии свободных (незарезервированных) ресурсов;
- обработка заявок, конфликтующих с уже зарезервированными ресурсами (оптимизация размещения заявок на ресурсах).

С самого начала работы системы (число поступивших заявок равно нулю) используется первый уровень обработки заявок. Он используется до тех пор, пока очередная поступившая заявка не будет конфликтовать с текущим распределением ресурсов. Для обработки таких заявок применим алгоритм второго уровня, использующий перераспределение ресурсов между заявками с целью удовлетворения поступившей заявки.

На первом уровне обработка происходит по следующему алгоритму:

определение множества свободных в заданный промежуток времени ресурсов с указанными в заявке свойствами;

вычисление определенного функционала "качества" этого ресурса для каждого ресурса из этого множества;

резервирование ресурса с оптимальным показателем качества.

Основной целью этого этапа является обработка максимального числа заявок без привлечения процедуры второго уровня (без оптимизации размещения заявок). Процедура второго уровня является более трудоемкой, чем процедура первого уровня, и может выполняться неограниченное время (время ее выполнения зависит от многих факторов — от числа принятых и перемещаемых заявок, от продолжительности времени выполнения каждой заявки и др.).

Кроме того, на данном этапе могут быть достигнуты и другие цели (например, равномерное использование ресурсов, использование наиболее дешевых ресурсов). Критерии использования ресурсов на этом уровне, а также приоритет каждого из критериев должны отражаться в используемом функционале "качества" ресурсов.

К процедура второго уровня прибегают в случае отсутствия в системе в данный промежуток времени свободных ресурсов и заключается она в попытке перераспределения ресурсов уже принятых заявок с целью освобождения необходимого числа ресурсов. Самый простой путь решения задачи — полный перебор вариантов размещения принятых заявок на подходящих ресурсах. Однако число просматриваемых вариантов в таком случае будет равно

$$N_z^{(r-1)},$$

где N_z — число перераспределяемых заявок, r — число подходящих ресурсов для одной заявки (в случае равного количества подходящих ресурсов).

Мы предлагаем алгоритм, который позволяет получить некоторые отсечения в переборе. Рассмотрим рекурсивный вариант этого алгоритма.

РАЗМЕСТИТЬ (Заявка i)

Найти подмножество ресурсов

$$R_i \subset R,$$

где R — множество всех ресурсов системы, удовлетворяющее требованиям Заявки i . Пусть J — мощность множества

$$R_j: |R_j| = J.$$

Для каждого ресурса

$$r_j \in R_i, j = \overline{1, J},$$

определить множество заявок Z_{ij} , препятствующих обслуживанию Заявки i с использованием ресурса r_j .

Если существует

$$Z_{ij} = \emptyset,$$

то разместить Заявку i на ресурсе r_j и завершить процедуру.

Иначе для каждого из множеств

$$Z_{ij}, j = \overline{1, J}$$

попытаться разместить все входящие в него заявки на других ресурсах.

$$\forall z_i, z_i \in Z_{ij} \text{ РАЗМЕСТИТЬ}(z_i)$$

Процедура прекращается при нахождении множества Z_{ij}^* , для которого удается разместить все входящие в него заявки. Заявка i размещается на ресурсе r_j . Если такое множество Z_{ij}^* не найдено, то Заявке i необходимо отказать.

Такой алгоритм достаточен для решения многих прикладных задач довольно большой размерности, хотя в некоторых случаях (при большой загруженности ресурсов, при большом разбросе в размерах и времени выполнения заявок) он приближается по сложности к полному перебору.

Теоретические исследования проблемы резервирования были апробированы в процессе создания программной подсистемы "Машина резервирования". Данная подсистема создана как отдельный модуль, имеющий

специальный интерфейс для настройки на конкретную проблемную область. Иными словами, подсистему проектировали таким образом, чтобы после настройки параметров использоваться в программных системах, обслуживающих задачи оптимизации резервирования ресурсов. На рис. 2 приведена информационная модель данного программного модуля.



Рис. 2

Модуль настройки обеспечивает инициализацию подсистемы на конкретную проблемную область. С его помощью подсистеме передается формат БД, конкретные свойства ресурсов, критерии оптимизации. После настройки подсистема способна обращаться к БД и обрабатывать запросы на резервирование, которые поступают в подсистему по установленному протоколу. Результаты обработки запроса выводятся на монитор в виде рекомендации оператору, который имеет возможность скорректировать и подтвердить резервирование. Подсистема предлагает оператору набор вариантов резервирования с проставленными коэффициентами предпочтения. Эти коэффициенты вычисляются на основе встроенного в подсистему алгоритма и настроек критериев оптимизации.

Подсистема "Машина резервирования" была использована в программном продукте "HOTELLO-2". Данный продукт представляет собой систему ведения гостиничного хозяйства, включающую: бронирование номеров, учет и анализ финансовой деятельности, ведение базы гостей, комнат, счетов.

Благодаря возможности иерархической организации структуры гостиницы, "HOTELLO-2" не накладывает ограничений на количество комнат. Все функции системы можно условно разделить на следующие группы.

Создание новых броней. Все функции по созданию новой брони сосредоточены в подсистеме *Reservation Wizard*, которая помогает пользователю легко достичь желаемого результата и на каждом шаге снабжает его исчер-

пывающей информацией. При создании брони система поддерживает следующие функции:

- выбор необходимого временного промежутка с подсказками в виде календаря и длительности брони;
- определение списка всех номеров, свободных в данный промежуток времени и удовлетворяющих запросу.

Пользователь может увидеть список всех номеров, свободных в данный промежуток времени (вне зависимости от категории номера), а также произвести бронирование номера непосредственно из календаря броней. Для этого необходимо просто с помощью мыши указать конкретные комнаты и временной промежуток (рис.4).

Работа со счетами. Система предлагает богатый набор различных функций для облегчения пользователю работы со счетами. В терминах системы счет может быть выписан гостю, комнате, группе или третьему лицу. Подсистема *Open-Invoice Wizard* помогает пользователю быстро найти существующий или создать новый счет, открыть для рассмотрения любое количество счетов и легко переходить со счета на счет; добавить, удалить, изменить позиции счета; перенести группу позиций из одного счета в другой, используя drag&drop.

На рис. 3 представлена информационная модель, показывающая, каким

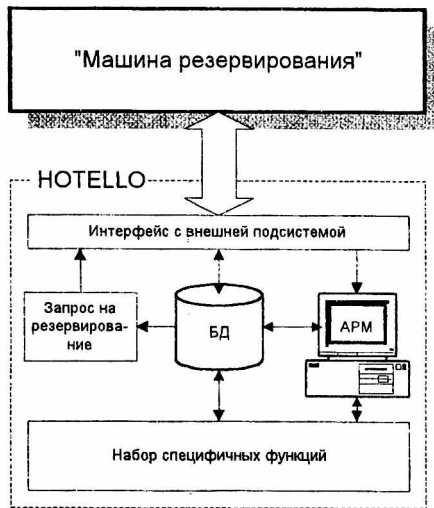


Рис. 3

образом подсистема резервирования внедрена в реальный программный продукт.

Поддержка сезонных цен на номера. Цена на номера в гостинице не постоянна и может меняться в зависимости от сезона, длительности брони, пакетных цен. Система позволяет пользователю создать список сезонных цен на каждую категорию номеров. По этому списку система автоматически определит цену номера и подскажет ее в момент резервирования, выдав соответствующий комментарий.

Оптимизация резервирования. Одной из проблем при работе гостиницы является оптимальное распределение броней по номерам с целью увеличения числа обслуживаемых гостей. Система имеет интеллектуальную подсистему, которая анализирует распределение броней и предлагает список номеров в порядке предпочтения. Данная подсистема учитывает следующие критерии:

- приоритет в использовании номера, предоставленный пользователем;
- примыкание новой брони к соседним;
- количество дней, которое остается между соседними бронями;
- количество дней, которые данный номер уже эксплуатировался.

Ограничение доступа к системе. В связи с тем, что систему могут обслуживать как один пользователь, так и целая группа клерков, система позволяет защищать определенные функции от несанкционированного доступа. Пользователь может эксплуатировать систему в двух режимах доступа:

- отсутствие ограничений в доступе. В этом режиме при входе в систему не нужно регистрироваться и вводить пароль. Все функции системы открыты для использования;
- режим ограничения в доступе. В этом режиме в систему может войти только зарегистрированный пользователь. При регистрации нового пользователя *Supervisor* может проставить ему определенные права в использовании функций системы.

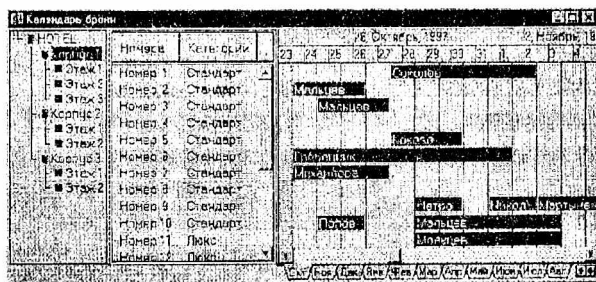


Рис. 4

Итак, в данной работе была рассмотрена проблема оптимизации резервирования в системах распределения ресурсов. Разработан метод решения задачи оптимального распределения ресурсов с минимизацией потерь и другими критериями. Предложенный метод позволяет избежать исчерпывающего перебора вариантов. Рассмотрены вопросы программной реализации указанного метода. Приведенный метод может быть применен для разработки различных программных систем распределения ресурсов. В статье также рассматривалось использование указанного метода для разработки системы автоматизированного управления ресурсами гостиницы "Hotello-2".

Список литературы: 1. Кофман А., Крюон Р. Массовое обслуживание. Теория и приложения. Пер. с франц. М., 1965. 2. Саати Т.Л. Элементы теории массового обслуживания и ее приложения. Пер. с англ. М., 1971.

Поступила в редколлегию 28.10.97